



BIROn - Birkbeck Institutional Research Online

Enabling Open Access to Birkbeck's Research Degree output

The emergence of interpersonal and social trust in online interactions

<https://eprints.bbk.ac.uk/id/eprint/52517/>

Version: Full Version

Citation: Prifti, Ylli (2023) The emergence of interpersonal and social trust in online interactions. [Thesis] (Unpublished)

© 2020 The Author(s)

All material available through BIROn is protected by intellectual property law, including copyright law.

Any use made of the contents should comply with the relevant law.

[Deposit Guide](#)
Contact: [email](#)

The Emergence of Interpersonal and Social Trust in Online Interactions

Ylli Prifti



A dissertation submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
Birkbeck, University of London.

Department of Computer Science
Birkbeck, University of London

November 7, 2023

I, Ylli Prifti, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

My PhD work is in the area of extracting and modelling user-created data on the web. In particular, I focussed on locating and extracting user data that 'signals' the evolution of human, 1-on-1 interactions between participants of large social networks who are forever stranger to each other.

The booming of "Online Social Networks" created an opportunity for social scientists to study social phenomena at a scale unseen before. The vast amount of information combined with computer science techniques led to significant developments in a relatively new field: Computational Social Science. Furthermore, in recent years the Gig Economy and mass adoption of "business sharing" sites such as Airbnb, Uber, or JustEat drove a new wave of computational social science research into reviews, feedback, and recommendations. All these ingredients of the larger Social Trust have been vastly discussed in the literature, in both the social aspect and computational models of trust.

However, some fundamental gaps remain, and there is often confusion about when trust is being expressed and how reviews (or recommendations) relate to social trust. Additionally, the computational trust models found in the literature tend to either be entirely theoretical or focused on a specific data set, thus lacking universal applicability. The latter problem, I believe, was due to a lack of data available to researchers in the early stages of the web. Today, the broader Online Social Networks have matured and consolidated mechanisms for allowing access to data. Access to information is rarely trivial for more specialised and smaller online communities. Yet smaller, focussed platforms are precisely where social trust and interactions could be observed (or not observed) and perhaps acquire a *meaning* that approaches the social trust social scientists see in in-person interactions.

To address this gap, we initially propose and discuss the following research question:

”Is there a meeting point between online interactions and social trust so that the core components of trust are retained?” We addressed this general open question by working on a computational architecture for data retrieval in social media platforms that can be suitably generalised and re-applied to different platforms. Lastly, as we enjoy the luxury of vast amounts of data that closely represent interpersonal and social trust, we addressed the question of ”what models of trust emerge from data” and ”how do existing models of trust perform with the data available”.

I have defined a category of online social networks that retains the core components of social trust, which we call ”Online Social Networks of Needs.” Hence, I have a classification and categorisation mechanism for grouping online social networks of needs by the level of trust necessary for cooperation (aka. the *cooperation threshold*) and interactions to be triggered among participating cognitive agents.

My focus has always been on data acquisition, and I have designed and implemented a system for data retrieval that is easily deployed to social media/social web platforms. A case study of such a system performing in a challenging scenario is further detailed to show the more extensive applicability of such a system for data retrieval and contribution to a scenario of complete distrust, anonymity, and ephemerality of data (such as 4chan.org). Further, studying the granularity of 4Chan data, we discovered that:

1. ephemerality is not sustained, and web archiving sites have a complete view of the ephemeral data [1],
2. we can track sentiment and topic modelling of moderation in 4chan [2], and
3. it is possible to have a live view of the topics and sentiment being discussed in the live board and see how these changes over time

We¹ studied the dynamics of high trust interactions [3] and found gender biases [4, 5] in care interactions.

Another topic related to trust but concerning institutions and media is the ’spillover’ effect between 4Chan and the traditional media. As a premise, 4chan anonymous threads have anticipated important global trends, notably the ”Anonymous” movement. Apart from the US, how do national topics interact with the essentially global discussion that is taking place there? Again, thanks to our extensive data collection/analysis, we sought

¹The use of the pronoun ”we” in this context means ”the reader and I”

to determine the level of participation from a selected non-US country, Norway, and the degree to which Norwegian 4chan /pol/ users and domestic news influence each other [6]. We continued the journey by collecting data from eight social networks of needs into the top two high trust demanding categories. Whilst these datasets are made available to researchers [7], we further study emerging networks and their properties and project the online social networks of needs into multiplex graphs by transforming the root links. Finally, we look into the applicability and predictive power of the non-reductionist model of trust proposed by Castelfranchi. We look at total social trust holistically and consider signals to evaluate fluctuations of the social capital influenced by economic and political dynamics and domination of the public discord by conspiracy theories.

Summary of contributions

1. the first comprehensive real-time scrape of 4Chan (in literature, only post hoc solutions were available);
2. the application of Castelfranchi's theoretical model of trust to actual data from online social networks;
3. one of the first studies on the relationship between the institutional (nationwide) press and extremisms on 4Chan;
4. the study of the application of predictive models to heterogeneous multi-source data (not user-created but not very trustable either), and
5. contributing live data scraping expertise into several other publications [8] [9].

Acknowledgements

Special thanks to my wife and kids for the support, encouragement and, most importantly, understanding for the lengthy times off during the past five years to make this work possible.

Immense gratitude for my parents and their sacrifices that made this journey possible for me and opened doors I wasn't aware existed.

Many Thanks to my sisters for the long talks and their firm belief that there is no need to do a PhD. It's great to see they are looking to do their own.

It has been a fantastic experience and collaboration with my supervisor. Alessandro's encouragement to jump on this journey and his input and direction have been crucial and inspiring to getting me here.

Special thanks to Prof Gottlob and the OXPath team for their help and for publishing an open-source version of OXPath on GitHub.

Special thanks to Dr Andrea Cali for his support and lengthy discussions on deep web querying. Special thanks to Dr Stelios Sotiriadis for his help with MongoDB Clusters and for allowing me to run guest sessions on Distributed Systems. Special thanks to Dr George Magoulas for his input and the discussions during the Knowledge Lab sessions. Special thanks to Dr Tingting Han for the opportunity to assist and expand knowledge on Big Data Analyt-

ics.

Special thanks to my colleagues: Iacopo Pozzana for the input and collaboration on 4chan, Justina Deveikyte for the discussions on web data retrieval, Seongil Han for the information and discussions on sentiment analysis, Paschalis Lagia for the talks on Machine Learning, and Salvatore Rapisarda for the discussions on systems performance and baselines.

Contents

1	Introduction	21
1.1	A question of trust	21
1.2	A question of data	22
1.3	Models of trust emerging from data	23
1.4	Structure and research layout	24
1.4.1	Introduction and Context	24
1.4.2	Data and Experimentation	25
1.5	Validation and Conclusions	25
2	Literature Review	27
2.1	Trust definition	30
2.2	Research on trust	35
2.2.1	Trust Information Collection	37
2.2.2	Trust evaluation	38
2.2.3	Trust dissemination	39
2.3	Conclusion	39
3	Theoretical Framework	41
3.1	Online traits of social and interpersonal trust	42
3.2	Characterisation and Quantification	45
3.3	Social Networks Analysis and Experimentation	52
3.3.1	Multiplex networks on OSNNs	55
3.4	Conclusions	56

4	Methodology	59
4.1	Research Methods	60
4.1.1	Observations and qualitative research	60
4.1.2	Non-empirical research in software engineering	62
4.1.3	Data collection and Case studies	63
4.1.4	Data analysis, quantitative and empirical research	64
4.2	Research Reproducibility	64
4.3	Privacy and ethical considerations	65
4.4	Conclusions	66
5	A distributed system for interpersonal and social trust data	67
5.1	Introduction	67
5.2	Related Work	67
5.3	A Modern System for Web Scraping	70
5.3.1	Architecture characteristics	71
5.3.2	Architectural design	72
5.3.3	An open implementation of data retrieval distributed system	74
5.4	Conclusions	96
6	Data collection campaigns and case study	99
6.1	Introduction	99
6.2	A Case Study - 4chan data collection	100
6.2.1	Data retrieval method	101
6.2.2	Data analysis	107
6.2.3	Case study limits and conclusions	118
6.3	Data collection campaigns and datasets	120
6.3.1	High trust datasets	120
6.3.2	Medium trust datasets	122
6.4	Conclusions	122
7	Trust evaluation and trust models emerging from data	125
7.1	Network analysis for the online social network of needs	126

7.2	Multiplex care network	131
7.3	An attribute-based trust model	139
7.4	Degrees of Trust predictability	151
7.5	Conclusions	156
8	Conclusions, limitations, and future work	159
8.1	Limits and Future work	162
	Appendices	165
A	Code	165
A.1	Data collection code snippets	180
B	Additional images and illustrations	183
C	Code repositories	187
	Bibliography	189

List of Figures

2.1	A refinement of literature work by Castelfranchi et al., Govier et al . . .	28
2.2	S.March - Trust Continuum: The limit of forgivability	32
2.3	W. Sherchan et al.: Comparison of existing Trust literature	36
2.4	W. Sherchan et al.: Building a social trust system - classification	36
2.5	Ziegler: Trust Metric Features	38
3.1	Basic interaction flow	45
3.2	Degrees of Trust, Data-Points with high trust	50
3.3	Factorisation Machines and the Feature Vector from Degrees of Trust .	51
3.4	Multilayer network representation of Online Social Networks of Needs	53
3.5	Multiplex graph for OSNNs	56
5.1	Tenant View	73
5.2	Component View	75
5.3	Technology Stack	81
5.4	Discovery query output structure	92
5.5	Discovery query output structure	93
6.1	Screenshot from 4chan /pol board on 5th of January 2022	101
6.2	Kubernetes deployment	106
6.3	4chan running view	107
6.4	Distribution of newly discovered threads by the time of the day and day of the week	109
6.5	New thread activity comparison by the time of the day and day of the week	110
6.6	Cumulative Distribution - number of threads by the length of life . . .	111

6.7	Logical Regression - Thread live time vs number of posts	112
6.8	Three-way comparison results projection	114
6.9	Deleted Posts Topics	117
6.10	Deleted posts sentiment	117
6.11	The observed distribution of 4chan Live Board and archived boards transitions	118
6.12	Kolmogorov-Smirnov sentiment analysis distribution comparison . . .	119
6.13	Rover.com discovery collection data structure visualisation	121
7.1	Childcare.co.uk network	127
7.2	Childcare provider received reviews distribution	128
7.3	Random graph view of 3000 parents, reviews, and provider nodes on childcare.co.uk	129
7.4	Eigenvector Centrality	130
7.5	Centrality measures for a sample of 5K nodes from the childcare network in London	131
7.6	A detailed view of the care network	133
7.7	Centrality measures of multiplex care network of 5K nodes in top 10 UK locations with the highest households	134
7.8	Multiplex care network simplified view of 5000 nodes from the top 10 locations with the highest households in the UK	136
7.9	Different perspectives of the Multiplex care network of 2000 nodes from the top 3 locations with the highest households in London	138
7.10	Probability distribution functions for CF Trust, Rating and Number of reviews received	144
7.11	CDF comparison of Random Trust, CF-Trust and reviews received . .	145
7.12	Scatter Plot CF-Trust vs Number of reviews received	146
7.13	CDF Convergence over ten iterations of belief calibration	148
7.14	Comparison of CF Trust with random and optimised parameters	150
7.15	MSE distribution for FM Predictions	153
7.16	Factorisation Matrix	154

A.1 Kubernetes CronJob deployment of a Scraping Project 166

B.1 4chan data structure 184

B.2 Number of discovery instances by time 184

B.3 Live board and t_{-1}, t_{-2}, t_{-3} boards topic models 185

B.4 Live board and t_{-1}, t_{-2}, t_{-3} boards sentiment 185

B.5 Rover.com search result example 186

List of Tables

3.1	Online social networks of needs - Classification	46
3.2	Online social networks of needs - Common characteristics	48
6.1	Data campaign output for the six months May to October 2021	108
6.2	Properties for the three-way comparison	113
6.3	Three-way comparison results (d/m differences/matches)	113
6.4	High Trust dataset characteristics	122
7.1	Top 10 providers by centrality measures. R: number of reviews received and L: number of locations where they provide their services.	132
7.2	Ten matching providers on two different platforms	135
7.3	Extradimensional links	137
7.4	Attribute contributions	140
7.5	Top 10 providers with the highest CF-Trust and the number of reviews received. Prov: Childcare Provider, CF: Castelfranchi Trust Score, Rand: Random Trust Score, R: Rating, C: Review Count	143
7.6	Rolling Parameter Optimisation with different sample size	149
7.7	Random walk parameter optimisation	149
7.8	Optimised Belief Parameters	150
7.9	Social Capital score per 100k households	151
7.10	Data for building the factorisation matrix	153
7.11	MSE of FM Predictions for the top 10 postcodes with the highest combined households	154

7.12	MSE of FM Predictions for the top 10 postcodes with the highest matching providers	155
7.13	MSE of FM Predictions for the results with the 5 highest and 5 lowest MSE	155
A.1	Kubernetes Cluster	165

Chapter 1

Introduction

1.1 A question of trust

Trust is a cognitive concept that spans a multitude of disciplines. One of the fundamental questions widely discussed in the literature is the definition of trust and how the various definitions of trust in the different disciplines correlate. The research for a universal explanation of "Trust" that spans all disciplines is an open quest. The quest becomes more complex when considering "Trust" in the computational social sciences and intelligent agents in an artificial intelligence context.

At the very least, the scale and the blurring of the term "interaction" in online interactions between social cognitive entities provide another unanswered question: "How does an online interaction between two cognitive entities correlate to interpersonal and social trust?". This question becomes more relevant when applying trust models to data representing interactions on online social networks, online peer-to-peer services, electronic commerce, and many other online exchanges.

To make the question more addressable by this research, we ask the following equivalent question: *"What fundamental ingredients of Trust, in online interactions, must be retained to guarantee an expression of interpersonal and social trust?"*

In addressing the above question, I discuss the ingredients of trust to retain the strongest definition of interpersonal and social trust. Further, we relax some terms and quantitatively transition from "high levels of trust" towards "lower levels of trust". By contrast, we discuss and study the behaviours and patterns emerging from a case of "total distrust".

The definition and categorisation of web communities and social networks that retain interpersonal and social trust characteristics is the first significant contribution of this research that helps clarify when the application of trust models is appropriate to data retrieved from the web.

1.2 A question of data

When trust models are discussed in the literature in the context of computational social science - these are often either theoretical, based on simulations or backed by data from one single source¹. In the latter case, we often observe the source not retaining the fundamental ingredients of expression of social trust.

Having defined a category of websites that encourage, capture and display data representing social interaction where social trust is expressed and measured, I am presented with multiple research opportunities. Validate existing models, derive models from the data, and more importantly - solve the problem of data retrieval and make the method and the data available to the research community.

However, the problem of retrieving the data is a non-trivial computer science problem. For this to be a meaningful contribution, the problem statement must consider the complexity of a dynamic and heterogeneous category of websites, which continuously amasses a large quantity of non-structured publicly accessible

¹Discussed in detail in Chapter 2

data.

In this report, I propose the following: *"A query-based open-source, scalable and distributed system used to retrieve large sets of data from the web"*. I discuss the architecture design of such a system, its implementation and its availability to the open-source community. I also show several running instances and actual use cases of such a system.

The availability of an open-source system to the research community that can run queries towards web communities to retrieve data is novel literature. It is the second significant contribution of this research project. Furthermore, large amounts of a variety of datasets representing interpersonal social trust have been made available as a result.

1.3 Models of trust emerging from data

Trust research is often characterised as trust information collection, trust evaluation and trust dissemination. This research contributes to all three categories. The contribution to the evaluation of trust compares existing models to many datasets, combined and structured into a single large connected graph. I study the properties of the graph and allocate different trust scores derived from existing models to graph nodes. I derive an alternative trust method that fits the data and has a better predictability score. The availability of such a graph presents the opportunity to (i) validate universal validity of existing models and (ii) search for trust models and patterns emerging from data.

In the literature on trust and web interactions, we find two significant trends which rather define the whole field. The first is about modelling user behaviour and applying models to typical Data Science tasks such as customer profiling, link prediction, shop-basket prediction etc. The model by Castelfranchi et al., which we will discuss in Chapter 3, is a champion of this approach. The second trend is about creating trust in web users, primarily by creating architectures that guarantee trust. Creating trust often is about money transactions, but also importantly about the veracity of personal interactions, such as verifying user-profiles and fact-checking news. An exciting example of the creating

trust research line is the recent EU-Horizon Eunomia project (eunomia.social) which focuses on creating inherently trustable social web architectures. To do so, it leverages technology from peer-to-peer and blockchain to ingenerate users' confidence in the contents shared on their platform. It is intriguing at this introductory level to point out the stark, contrasting visions of trust which fully coexist on the web today:

- **childcare:** stakes are highest, but families trust the site to bring them qualified childminders, which maybe they will be vetted on their own once the contact is established
- **4chan:** stakes are generally low but the user base, often considered a fringe anarco-libertarian social segment, actually do trust the platform and its Japan-based localisation to guarantee their privacy and anonymity even though they might later be exposed to profiling and investigation by the long-reaching arm of US investigative agency
- **Eunomia:** and in general research in web architectures which strives to insulate users from said platforms, as we discover from Eunomia's own summary: "EUNOMIA is developing the first social media environment designed to prioritise trust over likes. [...] Its open-source tools help the user quickly and confidently assess the trustworthiness of information shared through EUNOMIA without relying on any third-party expert or social media platform to do it for them."

1.4 Structure and research layout

The structure of this report is tightly connected with the structure of the research and can be grouped into three major sections: (i) Introduction and Context, (ii) Data and Experimentation, (iii) Validation and Conclusion.

1.4.1 Introduction and Context

This chapter covered a brief introduction. In Chapter 2, this report will visually set the context for discussing trust as a multidisciplinary topic focusing on computational social science and discuss a critical review of the existing literature on trust models, definitions, and the quantification of trust. The litera-

ture review will place this research in the existing literature and form the basis for the theoretical framework discussed in Chapter 3. The continuous improvement and operations research methods for architectural design and software implementation are discussed in Chapter 4, followed by a discussion of the empirical methods for data collection and methods for quantitative analysis of trust scoring. In Chapter 4, I will also discuss validation methods and use-case scenario layouts.

1.4.2 Data and Experimentation

Chapter 5 focuses on data, data retrieval and the challenges of mining data from the web. The computer science challenge is addressed with the proposal of a distributed architecture for a queryable web system for data mining, specialised in trust data collection. I discuss the implementation and execution of such a system and present two use cases for data collection. Chapter 6 discusses the statistical properties of several data collection campaigns executed during the past three years. I transform the semi-structured disconnected and heterogeneous data into a highly connected composite graph used for experimentation with trust and trust models.

1.5 Validation and Conclusions

Chapter 7 is concerned with validating the results. We start with validating the data and comparing values between different data collection methods. Trust scores from different models are compared against the observed values and trust expressed in real terms in the social communities. Prediction models are run based on results from previous chapters and predictability scores compared. I finish Chapter 7, where conclusions, takeaway results, and future work are laid out to the reader. A chapter with a critique and limits of this work is also included. The following two chapters are appendixes that include and describe run-books for some software produced and analysis reproducibility steps, details about access to the source code and lastly, an analysis of data from a source of total distrust.

Chapter 2

Literature Review

Trust is one of those fundamental cognitive concepts that is embedded in and around our everyday life, affects and models our behaviours, yet we rarely have to consciously stop and think about *what is trust*. However, as I discovered through this research and aim to show in this literature review, the question is hardly trivial. Not only is the discussion about trust, in general, a multidisciplinary and vast subject but also it spans multiple dimensions and variations that make it difficult to visualise. Within the scope of this research, which focuses on models of interpersonal and social trust emerging from data collected from online communities and social networks, we are proposing the Venn Diagram in Figure 2.1 to help put this literature review in context.

The non-triviality of the Trust Modelling is underscored by the amount of literature dedicated to it, the interlink between the different disciplines, and the challenge to frame the discussion to a number of concepts and disciplines relevant to the research. This work and its literature review will be mainly focused on trust in the context of Computational Social Science. We clearly separate the concepts of "social trust" and "interpersonal trust" seen as highly overlapping but clearly distinct concepts. Whilst the concepts are often used interchangeably in the literature, Govier ([10] p.31) stated the clear difference between the two:

"Social trust and interpersonal trust are different in significant ways. Interpersonal trust is based on experience, sometimes deep and intimate

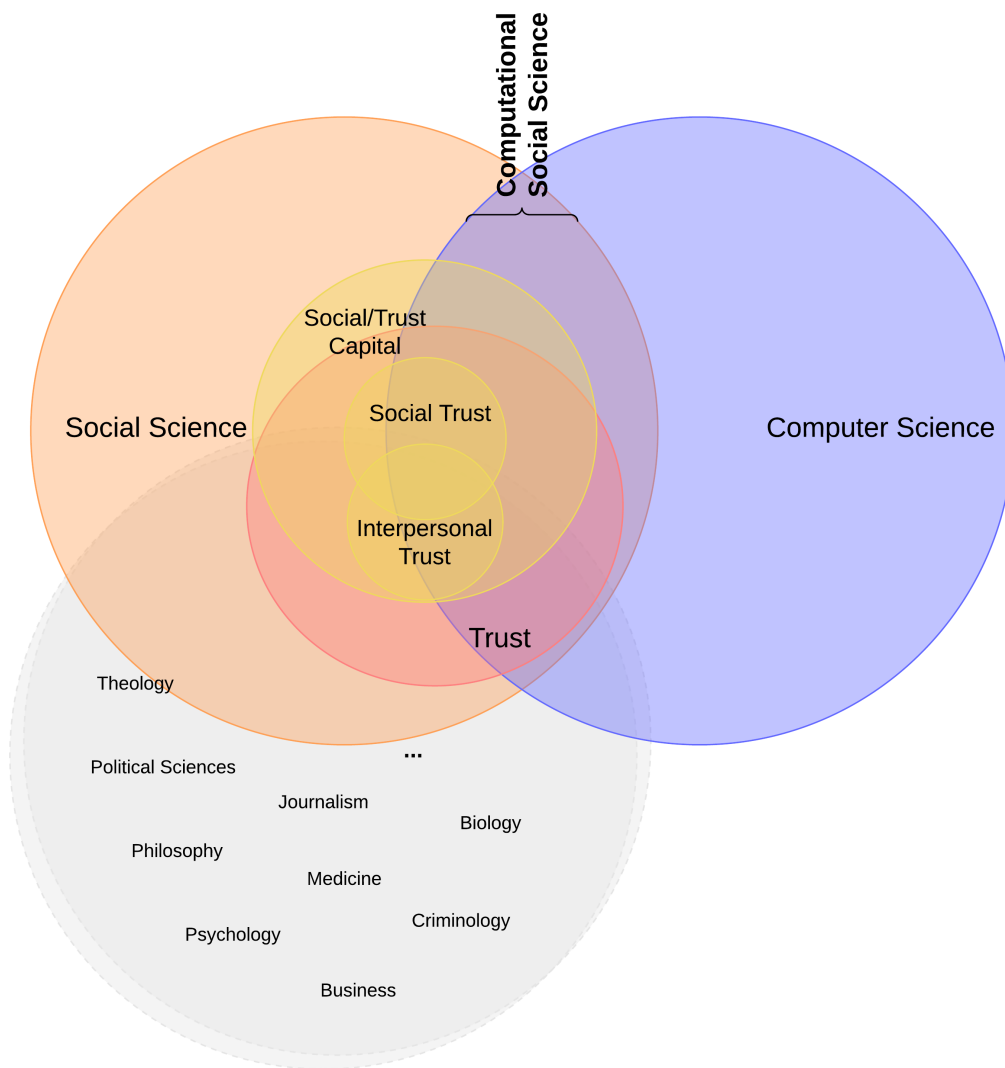


Figure 2.1: A refinement of literature work by Castelfranchi et al., Govier et al

experience, with another individual. In some cases of social trust, we may have limited experience with the other persons involved; in others, we have none at all. (...) despite these differences (...), it is neither an accident nor a logical mistake that the word 'trust' should be used across personal and broadly social contexts."

There are clear traits of distinction, interchangeability, and interlinking (i.e. how one affects the other) between "social trust" and "interpersonal trust" that are not widely discussed in the literature but emerged naturally during the analysis of the data collected as part of this research. In fact, in Chapter 3, we discuss and use some of these characteristics

to define a new group of online social communities where the traits of core trust [11] (Page 102) are clearly and explicitly expressed. According to Castelfranchi *Core Trust* is:

a set of mental states (MS – CTX,Y) -called Core Trust- with these components:

- *a set of X's goals and, in particular, one specific of them (gX) in order to trust Y;*
- *a set of X's competence beliefs (B–ComX,Y) on Y about τ ;*
- *a set of X's disposition beliefs (B–DisX,Y) on Y about τ and*
- *a set of X's practical opportunities beliefs (B–PrOpX,Y) on Y about τ at that given moment (time) and site (space)*

In Figure 2.1, we indicate that social trust and interpersonal trust are contributors to another concept: social capital (often referred to as trust capital). Whilst "social capital" is not the main focus of this research, we believe it to be important to highlight that social capital is most commonly defined [12] as

"the aggregate of the actual or potential resources which are linked to possession of a durable network of more or less institutionalised relationships of mutual acquaintance or recognition"¹

The concept of social capital is important both in the sense that *"When a society has social capital, just about everything is easier because people can turn to others for information and assistance"*([10], p.152) but also *"(...) [social capital] must be built from [interpersonal trust]. (...), trust capital is a macro, emerging phenomenon; it must also be understood in terms of its micro-foundations."*([11], p.29).

The two statements can be explicitly re-phrased as follows: (i) an abundance of social capital will influence and promote more trust among individuals and hence higher social trust, and (ii) interpersonal trust and social trust are fundamental components of social capital, hence societies with high interpersonal and social trust will inevitably have higher

¹Whilst this definition of social capital is the work of (Bourdieu 1985, p248; 1980 [13]), we are referring to the work of Portes, 1998 [12] analysing the origin and various definitions of social capital

social capital. This concept is clearly stated by Putnam et al. ([14] p.177) that suggests, because of this nature of social capital and trust, societies will converge to an equilibrium of either high social capital and high trust or the opposite spectrum

”Stocks of social capital, such as trust, norms, and networks, tend to be self-reinforcing and cumulative. Virtuous circles result in social equilibria with high levels of cooperation, trust, reciprocity, civic engagement, and collective well-being. (...) Conversely, the absence of these traits in the uncivic community is also self-reinforcing. Defection, distrust, shirking, exploitation, isolation, disorder, and stagnation intensify one another in a suffocating miasma of vicious circles.”

In Chapter 7, after calculating the trust values for each agent in a community, we use these definitions of social capital to compute the aggregated social capital values by location and quantify the relations between social capital and wealth generation data.

In conclusion, whilst this introduction to the literature review is focused on contextualising and visualising the disciplinary belonging of ”social trust” and ”interpersonal trust”, in the following sections, we will discuss and review the literature on the broader concepts of ”trust definitions”, ”trust models” and typical ”methods of research on trust”.

2.1 Trust definition

What is trust? When trying to answer this simple question, the literature inevitably points towards numerous definitions that have been extensively quoted whenever the definition of trust is discussed. Rousseau, 1998 [15] defined trust as:

”Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behaviour of another.”

We take this definition to mean that this is really about the psychological state between an individual - the Trustor, that clearly is a cognitive entity (who has a psychological state and consciously accepts vulnerability), and the Trustee because of positive expectations. Whilst it is very likely that Rousseau intended the Trustee as a cognitive entity, this

definition doesn't necessarily imply or require it. In fact, the definition remains valid in a hypothetical scenario where a cognitive agent trusts a ticket machine to deliver the ticket after having accepted the vulnerability of advancing the money based on the expectation and the behaviour that the machine usually does and will deliver the ticket. This is a common pattern in literature where the trustee is not strictly required to be a cognitive entity, and trust is discussed in its broader context.

Gambetta defines trust in terms of probability of cooperation [16]:

"... given a degree of trust, predicted on whatever evidence other than the interests of that person - the question is: how high does that probability have to be for us to engage in an action the success of which depends on whether the other person or persons will act cooperatively?"

This definition of trust introduces two new aspects: (i) a potentially quantifiable variable (i.e. the probability of performing a certain action) and a threshold for cooperation, and (ii) a situation at which trust about an action is evaluated.

Most of the research efforts on quantifying and computing trust are part of developments in information security. This concept is usually described as Computational Trust. Quoting from Wikipedia ²

"Computational Trust applies the human notion of trust to the digital world",

referring to Marsh's work on computational trust. Some of the models of trust discussed in the later sections of this chapter are developed in the context of computational trust. In Chapter 3, we take a different approach, looking closer at the traits of trust emerging between cognitive agents interacting via digital means.

Marsh, whilst considered a pioneer of computational trust, also introduced a concept of trust continuum in his PhD thesis in 1994 [17] and later refined it further in [18].

²https://en.wikipedia.org/wiki/Computational_trust

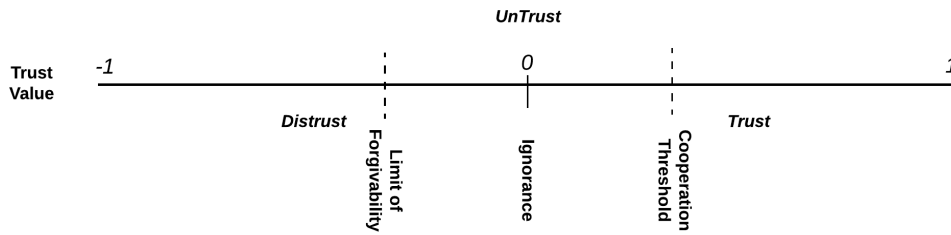


Figure 2.2: S. March - Trust Continuum: The limit of forgivability

Marsh discussed trust as a continuum between -1 , a situation of complete distrust and $+1$, complete trust. Marsh clarifies the difference between *Untrust* or *Lack of Trust* and *Distrust*. The former is a situation of ignorance, where there is not enough information to either trust or distrust. On the other side, this is different to *Distrust*, where the Trustor firmly believes that the Trustee will perform an action that goes against his interest (intentionally or unintentionally).

In Marsh's trust continuum spectrum, we also find the "Cooperation Threshold" put in perspective. We can have situations of trust, but not cooperation. For example, whilst looking for a babysitter on online sitter communities, we review many babysitters and trust them to be good with children, yet we look further until we find the few we trust beyond the cooperation threshold, and these are the ones we will contact to offer the job. Similarly, Marsh discussed "Limit of Forgivability", beyond which point it is unlikely the Trustor would trust again and might have decided about the intentionality of the Trustee's action going against the desired outcome.

Marsh's work is important not only because he discussed trust in its composing components (delegation, cooperation, forgiveness, regret etc.) but also because he proposed a first formal model for trust. This consisted of concepts of *situations* (α, β, \dots), *agent* (a, b, c, \dots), *knowledge* (e.g. x knows y $K_x(y)$), *importance* (e.g. of α to x , $I_x(\alpha)$), *utility* (e.g. of α to x , $U_x(\alpha)$), *basic trust* (i.e. the propensity an agent x has to trust someone in general, T_x), *general trust* (i.e. similar to basic trust but in relation to another agent y , $T_x(y)$), and lastly *situation trust* (i.e. the trust an agent x has on agent y to perform action

α , $T_x(y, \alpha)$ with the situational trust expressed as:

$$T_x(y, \alpha) = U_x(\alpha) \times I_x(\alpha) \times \widehat{T_x(y)} \quad (2.1)$$

With the proposed formalisation in 2.1 Marsh extends the concept of trust as discussed in Game Theory (for example, as the quantifiable amount in Prisoners' Dilemma) and Artificial Intelligence (where trust is discussed in terms of situations and agents with cognition in terms of utility)

Whilst Marsh was a pioneer in formalising and discussing Trust in an algorithmic fashion, Castelfranchi et al. [11] set up to build a non-reductionist model of trust. Castelfranchi formalises the concepts of *Trust*, *Lack of Trust* and *Distrust* but goes further from Marsh's model to separate *Distrust* from *Mistrust* based on trustee's intentionality. *Mistrust* is defined as:

Definition 2.1.1. MISTRUST: Agent i mistrusts j to ensure $\neg\phi$ by performing action α if and only if:

1. i wants to achieve ϕ ;
2. i expects that
 - j has the opportunity to ensure $\neg\phi$ by performing action α AND
 - j intends to perform action α AND
 - the internal preconditions for the execution of action α by agent j hold AND
 - the external preconditions for the execution of action α by agent j hold.

There is a clear distinction in this definition between *Distrust* and *Mistrust*. Whilst the former is the belief of the Trustor that the Trustee won't perform action α hence the Trustor won't achieve the goal ϕ , in the latter, the Trustee is actively pursuing an action α that will achieve the goal $\neg\phi$. Mistrust is the active pursuit of actions for the Trustor *not* to achieve his goal. Castelfranchi breaks down trust into a number of components

on another dimension to those discussed by Marsh. These are the components of "basic trust" broken down as (i) *competence*, (ii) *disposition*, (iii) *dependence*, (iv) *fulfilment*, (v) *willingness*, (vi) *persistence*, (vii) *self-confidences* and (viii) *motivation*.

In [19] we see a clear distinction between *Trust* and *Social Trust*, with the latter requiring the trustee (or agent y) to be a cognitive entity. This is discussed in terms of delegation (delegation being the decision to trust), distinguishing between weak delegation (i.e. no presupposition in terms of agreement, deals, or promises) and strong delegation (i.e. presupposition of goal adaption by the trustee y). Castelfranchi states:

"... *social trust in the strong delegation, which is its typical and strict sense in the social sciences*". ([19], p.8)

Another contribution of Castelfranchi relevant to this research is the discussion of the quantification of trust and the definition of the degrees of trust, as in:

Definition 2.1.2. Degrees of Trust: The degree of trust the cognitive agent X has on the cognitive agent Y about the situation τ is

$$DoT_{XY\tau} = DoC_X[Opp_Y(\alpha, g)] * DoC_X[Ability_Y(\alpha)] * DoC_X(WillDo_Y(\alpha, g)) \quad (2.2)$$

where:

- $DoC_X[Opp_Y(\alpha, g)]$ is the degree of credibility of X 's beliefs about Y 's opportunity of performing α to realize g ;
- $DoC_X[Ability_Y(\alpha)]$ is the degree of credibility of X 's beliefs about the Y 's ability/competence to perform α ;
- $DoC_X(WillDo_Y(\alpha, g))$ is the degree of credibility of X 's beliefs about Y 's actual performance and can be broken down as $DoC_X(WillDo_Y(\alpha, g)) = DoC_X[Intend_Y(\alpha, g)] * DoC_X(Persist_Y(\alpha, g))$, given that Y is a cognitive agent.

In other words, the degree of trust can be expressed as a composition of beliefs and credibility of agent X on agent Y 's opportunity, ability and competence, intentions

and persistence to perform the action required to achieve X 's goal. In our discussion, when setting the theoretical groundwork for this research, we will use Castelfranchi's degrees of trust definition 2.1.2 as a sound foundation for the models emerging from data.

In conclusion, most trust definitions and trust models in literature originate from the social sciences. They tend to describe trust in terms of Trustor, risk and vulnerability, Trustee and positive expectations, goals, and actions. When trust is discussed in terms of computer science, game theory and artificial intelligence, we see a change in language (from Trustor and Trustee to agent X and Y), a shift in goal (from decomposition into cognitive components to quantification and trust continuum) and lastly a change in representation (from human language to formal language and equations). The different representations are by no means only formal. We find in Castelfranchi's model a very close representation of the data we collected during this research.

2.2 Research on trust

Research on trust goes beyond the trust definition. In a meta-analysis publication on Trust models in Social Networks, Scherchan et al. [20] compiled the table in Figure 2.3 on different aspects of trust models.

Whilst some more recent trust models and metrics ([21], [22]) are not represented in Table 2.3, the categorisation and metadata bring to light some other essential characteristics on the literature about trust models.

For example, it is clear that most research focuses on Trust Information Collection and Trust Evaluation, but less attention is paid to Trust Dissemination. These three categories are also discussed in [20] as a categorisation for all literature on trust models, as shown in Figure 2.4

This report contributes in all three categories, and we will briefly review the literature categorised as: (i) Trust Information Collection, (ii) Trust Evaluation and (iii) Trust Dissemination

Method	Origin Discipline	Trust Properties	Trust Computation Model	Trust Information Collection	Trust Evaluation	Trust Dissemination	Malicious Attack Resistance	Application Domains
EigenTrust-Kamvar <i>et al.</i> (2003)	C	Pr, Sj	LN	E	G	TR	Y	P2P
PeerTrust-Xiong <i>et al.</i> (2004)	C	Dy, Cs, Sj	LN	E	G	TR	Y	P2P
Yu <i>et al.</i> (2004)	C	Pr, Sj	LN	E	G	TR	Y	P2P
TidalTrust-Golbeck <i>et al.</i> (2005)	C	Pr, Sj	Prob.	E	G	TR	N	SN
Josang <i>et al.</i> (2006)	C	Dy, Pr, Sj	Prob.	Bl	G	TR	N	P2P
Zhang <i>et al.</i> (2006)	S	Pr, Sj	LN	E	G	TR	N	SN
SUNNY-Kuter <i>et al.</i> (2007)	C	Sj	BN	B	G	-	N	SN
Maheswaran <i>et al.</i> (2007)	C	Pr, Cs, Sj	LN	E	G	TR	N	SN
PowerTrust-Zhou <i>et al.</i> (2007)	C	Pr, Sj	BN	E	N	-	Y	P2P, SN
Caverlee <i>et al.</i> (2008)	S	Dy	LN	B, E	G	TR	Y	P2P, SN
Liu <i>et al.</i> (2008)	C	Dy	Binary Classification	B	I	-	N	SN, e-commerce
Paradesi <i>et al.</i> (2008/2009)	C	Cp	BN	E	-	-	N	WS
Yan <i>et al.</i> (2009)	C	Sj	LN	B	-	-	N	MA
Zuo <i>et al.</i> (2009)	C	Pr, Cp	LN	TC	G	-	N	SN
Adali <i>et al.</i> (2010)	S	-	Log.	B	G	-	N	SN
Nepal <i>et al.</i> (2010)	S	Es	Prob.	B	-	-	N	WS, SN
Trifunovic <i>et al.</i> (2010)	S	Pr, Sj	LN	B, E	H	-	Y	SN
Nepal <i>et al.</i> (2011)	S	Sj, Dy, Cs	LN	B	I	-	N	SN

Origin Discipline	Trust Properties	Trust Computation Model	Trust Information Collection
P: Psychology, S: Sociology, C: Computer Science	Cs: Context-specific, Dy: Dynamic, Pr: Propagative, Cp: Composable, Sj: Subjective, Es: Event Sensitive	Log.: Logarithmic, Prob.: Probabilistic, BN: Bayesian Networks, LN: Linear Model (Sum or product)	B: Behaviour, E: Experience, Bl: Belief, TC: Trust Certificate

Trust Evaluation	Trust Dissemination	Attack Resistance	Application Domains
G: Graph-based, I: Interaction-based, H: Hybrid, -: Not specified	TR: Trust-based Recommendation, VZ: Visualisation, -: Not specified	Y: Malicious attacks considered, N: Malicious attacks not considered	P2P: Peer-to-peer networks, SN: Social Networks, WS: Web Services, MA: Mobile Applications

Figure 2.3: W. Sherchan et al.: Comparison of existing Trust literature

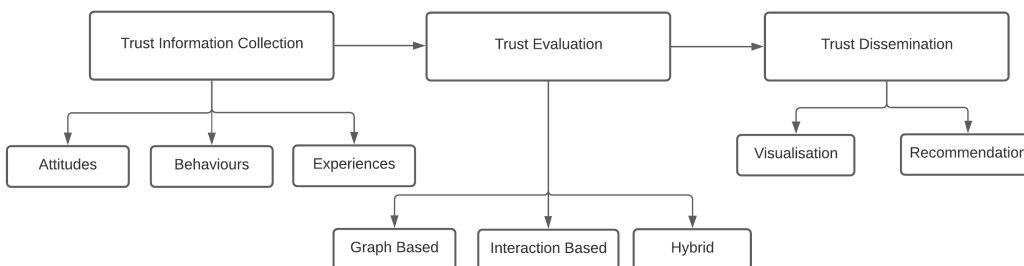


Figure 2.4: W. Sherchan et al.: Building a social trust system - classification

2.2.1 Trust Information Collection

Figure 2.4 suggests that trust information collection usually focuses on attitudes, behaviours, and experiences. We find this categorisation to be generally a good representation; in fact, we will discuss our data collection campaigns' focus against this categorisation. Interestingly, this type of categorisation does not offer any insight into the aspects of data retrieval, unlike what the category name suggests. In general, the literature focuses little on the aspect of trust data³ retrieval, even though we believe the availability of data influences and determines the direction of the research on trust. For example, this is the case on trust metrics such as the '*Advogato Trust Metric*' [21], where the metric is derived from the data available from the Advogato Community Network. Further from this, Ziegler's Appleseed [22] extends on Advogato, and its performance is tested against it because "*Advogato has already proven its efficiency in practical usage scenarios such as the Advogato online community*" (pg.152).

Whilst data retrieval literature is vast, we find little to no evidence of more focused literature on how this applies, affects and is affected by trust information collection. This phenomenon has been recognised early in literature, and the importance of access to data is recognised. Golbeck [24] states the following: *Evaluation is very difficult when working with social trust, particularly in modelling, propagation and other estimation methods. (...) Nearly all social networks and other applications with trust ratings from their users keep this data hidden. From a research perspective, this is challenging because there are no open sources of data available to work with. For researchers who are fortunate enough to have access to their own social networks (...), they are still testing on only **one** network. (...) In fact, this lack of data means that frequently, research on trust models and inference algorithms is published with little empirical analysis.*

³The use of the term "trust data" in the literature is associated with "*information that an agent or algorithm can use to evaluate trust*". This meaning, for example, is seen consistently in a collection of papers on trust models and metrics by Golbeck [23]

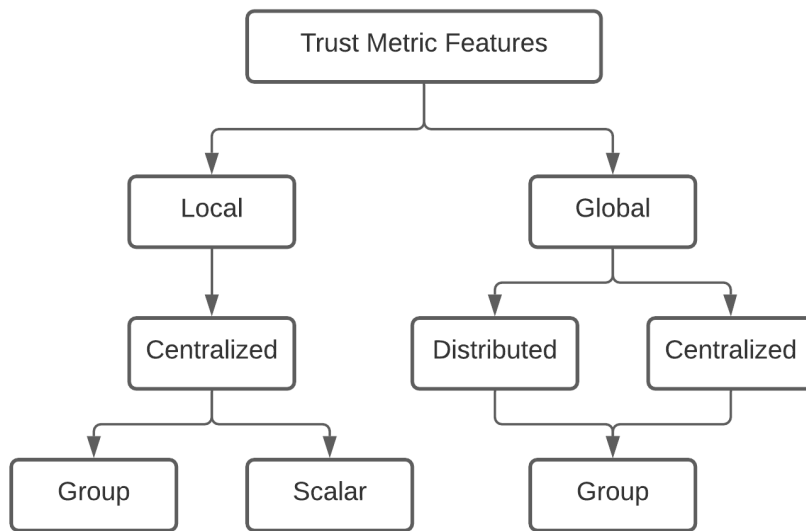


Figure 2.5: Ziegler: Trust Metric Features

This has been our⁴ experience with trying to get data from online communities. When contacting and trying to collaborate, most did not reply to our requests even when we shared we had collected data using other means.

2.2.2 Trust evaluation

Trust evaluation is, in general, concerned with trust metrics and trust models. Sherman's categorisation considers the methods used to calculate a trust score. Ziegler provides another categorisation for trust metrics based on their features instead, shown in Figure 2.5. This categorisation takes into account if the calculation is based on the totality of the network (global) or by partially computing network neighbourhoods if the calculation is centralised or distributed, and if the metric is a scalar calculation for a node/agent or a group calculation for a number of linked nodes/agents.

Levien [21], Ziegler [22], Golbeck [25] discuss their and other trust metrics by placing them in the Figure 2.5 categorisation. There is a vast number of trust models and metrics proposed in the literature. Following from the table in Figure 2.4, a number of more recent metrics are worth mentioning for their fundamental role in the field (FOAF [26], PageRank [27], EigenTrust [28], Advogato [21], Appleseed [22], FilmTrust [29], SNTrust [30]).

⁴I reached out to my supervisor and used the university structures to reach out to online communities for anonymised access to their user data. Unfortunately, this was unsuccessful, and most were reluctant to share these data

2.2.3 Trust dissemination

Trust dissemination is closely connected with the practical use of trust evaluations. Table 2.4 suggests very few trust dissemination categories (trust-based recommendations and visualisations). This, together with the applications domains (also limited to only a handful of categories), pictures a very narrow view of the practical use of trust metrics and models. We know this is untrue, and other classes and practical uses can be found in literature and the enterprise world. Trust models have been vastly developed and affected by security and identity management. Trust-related concepts (such as trust corrosion, Sybil attack [31] and attack resistance [32]) are key concepts and offer fresh insights and direction on how to address fake news and conspiracy theories. These studies have become even more critical with the latest developments of governance bodies' efforts to roll out COVID-19 vaccination, whilst the public struggles to understand whom to trust [33]. We will explore some areas and pragmatic approaches for trust dissemination in later chapters of this PhD thesis.

2.3 Conclusion

Literature on trust was discussed as a vast subject that spans multiple disciplines. Multiple trust definitions can be found in the literature, and some gaps were identified. Perhaps surprisingly, an association of interpersonal and social trust with online interactions and trust data from online social networks is seldom found in the literature. Trust models are either theoretical or validated on single networks, where trust is generally discussed in its broader meaning. We weren't able to find a holistic approach in the literature which aims to retrieve data from multiple social networks that retain and encourage interpersonal and social trust and evaluate trust models emerging from data. This will become the main focus of this PhD Thesis. We focus on interpersonal and social trust in online social communities and networks and a relevant definition.

Literature on trust is categorised as (i) Trust Information Collection, (ii) Trust Evaluation and (iii) Trust Dissemination and this research will contribute to all three areas.

The literature agrees on the difficulty of validating trust models because of the difficulty of retrieving and accessing trust data. A substantial part of this PhD thesis will address this problem. In turn, the availability of vast amounts and diverse data on trust creates an

opportunity to discuss trust models emerging from data and follow a pragmatic approach to trust dissemination as applied to the sources at hand.

Chapter 3

Theoretical Framework

I started this journey with my master's degree where I argued that the business model of the sharing economies (often called "Business Sharing" models), has not realised its full potential and is struggling to penetrate industries requiring high demand of trust, such as child care or elderly care. I analysed one particular community of carers in the UK, childcare.co.uk. Whilst some important conclusions were derived, many questions remain open. I continue on the journey of this PhD research with the intent of answering some of them. As discussed in the literature review, there is clearly a question of trust models: "*how to choose an appropriate trust model?*". The question is only partial, and the reader may ask "*an appropriate model of trust to achieve what and apply where?*". Before addressing these two important questions, I am extending the discussion that started with the literature review into two areas that will set the theoretical basis for addressing the above questions. These are:

- What traits, attributes and properties of social and interpersonal trust can we find in online communities, social networks or elsewhere on the web?
- Can we capture this information in a scalable and sustainable way for it to become an important seed in the study of trust?

These two questions clearly set us on the path to addressing the bigger question: "How to choose a trust model that accounts for the data available? Is there consistency among different models? Are there any better models emerging from data?". With this aim in mind, I am setting on the journey of explaining the theoretical framework to build upon

and address these research questions.

3.1 Online traits of social and interpersonal trust

In Chapter 2 we discussed the difference between "social" and "interpersonal" trust and quoted Govier [10] to highlight some differences. It is clear in Figure 2.1 that both concepts have a large overlap, affect each other, and where one is found, it is likely the other trait to emerge. This justifies the use of the term interchangeably in the literature. There are, however, certain aspects that we will discuss within the specific domain of either "interpersonal" or "social" during this research. We pay particular attention to *"interpersonal trust is based on personal experience"*.

This definition might seem too contradictory with the word "online" since interpersonal might be interpreted to mean "in person" or "physical" which the word "online" obviously excludes. I believe that the interpersonal aspect can emerge online, and there are enough narratives and evidence of intimate relationships initiating and expanding completely online. Hence, I find no contradiction in finding interpersonal trust emerging online. However, it is easier for the interpersonal experience of trust-building (or destroying) events to happen more naturally during physical exchanges.

In the evaluation of trust, and the more specific aspect of it, "interpersonal", I am introducing as part of the quantification process a diversification between "online" interpersonal exchanges and "physical" or "in-person" exchanges where the agents involved in the exchange have some interchange that requires both to be physically present at the same location at the same time and interact with each other. I will expand the categorisation further but want to use this diversification to introduce a trust exchange process that can be found in literature and is the basis for categorising a group of online communities that clearly express these aspects of trust and will be the subject of study in this research.

Definition 3.1.1. Online Social Networks of Needs - OSNNs are the group of online

communities where agents and interactions can be reduced to the following process

1. A cognitive Agent i expresses its goal ϕ to the community of cognitive agents AGT
2. A subset of agents $z \subseteq AGT$ expresses their will, have the ability and the opportunity to do action α to ensure ϕ
3. Agent i evaluates and expresses its core trust¹ by selecting agent $j \in z$ to do α to ensure its goal ϕ
4. Agent j does α which requires both agents to have an **in-person** exchange.
5. Agent i and j feed back about their interaction and the achievement of goal ϕ and updates its Core Trust on agent j . This information is propagated to AGT and core trust is updated accordingly.

There is some overlap of this definition with the definitions of "sharing economy"², "collaborative consumption", "circular economy", even though there is generally no consensus in the literature about these definitions [34]. We believe the definition of OSNNs to be novel in literature because it expands the exchange process to the expression of trust, is explicit about the properties of the agents involved and requires the feedback process after the exchange.

This definition and characterisations clearly exclude conventional social networks such as Twitter or Facebook. Though trust traits can emerge³, they clearly do not contain the necessary characteristics of building and expressing interpersonal and social trust, as defined in 3.1.1. In most cases, there are no in-person interactions in the strong sense of the definition, and where these interactions are present [36], there is no feedback process or base core

¹Core Trust as defined in [19]

²Defined as "an economic system in which assets or services are shared between private individuals, either free or for a fee, typically by means of the internet." in the Definitions from Oxford Languages <https://languages.oup.com/google-dictionary-en/>

³For example, Facebook groups tend to bring people together by subject and interest. The trustworthiness of posts and trust propagation can be found in the literature for Facebook groups [35].

trust.

Another category of web applications that are often [37–39] discussed within the realm of trust and are also excluded by definition 3.1.1 in the strict sense of interpersonal and social trust, are the e-commerce sites, such as Amazon and E-Bay. Reviews and feedback certainly happen on these websites and there is clearly the basis for core trust emerging, but there are very few to no interpersonal relations. Certainly, none happens in person and there is very little sense of community or other social traits.

In contrast, there is a group of social networks that allow and promote the initiation of face-to-face social interaction and then collect feedback about such interactions.

These group often takes the form of sharing economies models (like Uber or Airbnb) applied to different and heterogeneous needs like a taxi or a place to stay but also the need for a babysitter or elderly care, need to fix a broken pipe at home or help with home cleaning. We will be referring to these groups of social networks as Online Social Networks of Needs or OSNNs.

A simplistic representation is given by the flow diagram 3.1. OSNNs commonly express the following characteristics:

1. Participants (agents) are individuals who have a specific real-life need
2. Interactions between individuals translate to real-life interactions ⁴
3. Feedback about interactions is collected and made publicly available to all individuals
4. Individuals make their selection based on their assessment of previous feedback but also other features influence the decision to collaborate or not (distance, price, and other influence decision-making)

The characteristics mentioned above can make OSNNs crucial in studying, extending and validating trust models and metrics. Whilst these are a number of well-established sites being in business for longer than 10 years, numerous new communities are emerging,

⁴Note that the real-life interaction to satisfy a need can be online or face-to-face.

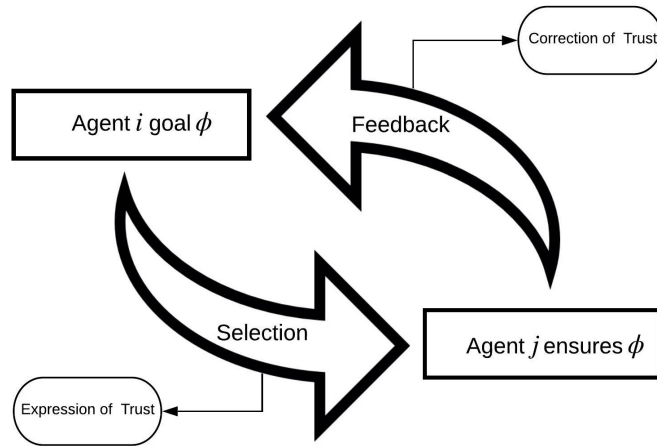


Figure 3.1: Basic interaction flow

having less visibility and certainly not comparable in size with mainstream online social networks. Yet they are dominant in specific interests and needs. They also tend to use less sophisticated web technologies, rarely allow access to their data via APIs, or similar technologies or, based on our experience, have little interest in sharing data with academic institutions. However, because of the nature of these businesses, extensive parts of their data are publicly available on their web pages.

3.2 Characterisation and Quantification

Having defined OSNNs and introduced some characteristics of the web communities of interest to this study, it is clear that many websites belong to this group. I am going to introduce a characterisation within the group based on the traits and forms of the expression and emergence of trust. As we go through the list, lines begin to blur and the belonging to the OSNNs group begins to fade. In fact, out of nine different categories, only the first six categories belong to OSNNs as defined in 3.1.1. All the categories proposed, presuppose web communities where experience feedback and reviews are collected and publicly shown. If we look at Table 3.1, we can see that from top to bottom, there is a sense of natural quantification of the value of trust. Arguably, trusting someone with the care for our loved ones is more trust-demanding than property care, for example. Whilst this narrative might look natural and acceptable, this is clearly different from quantifying trust and being able to empirically show the quantity of trust expressed and exchanged in each instance. Definition 3.1.1 and the common characteristics of the web communities belonging to

Table 3.1: Online social networks of needs - Classification

N°	Category	Description
1	Health and Care	Web communities where health and care for children, elderly, animals, and other loved ones are requested and offered
2	Property Care	Web communities where property maintenance work is offered or requested
3	Property Sharing	Web communities where sharing someone's property (for example a car or a house) is requested and offered
4	Tutoring	Web communities where tutoring, private lessons and skill learning are offered and requested
5	Online Gigs	Web communities where different gigs that can be performed and/or offered online are offered and requested
6	Skill Sharing	Web communities where advice and advanced skills are requested and offered
Other groups not part of OSNNs		
7	Third-Party Trust Providers	Web communities where users can provide feedback about a service they have received
8	E-Commerce	Web communities where different actors sell and buy goods and services
9	Social Network Groups of Interest	Forums and groups on social networks where a particular interest is discussed and where leaders emerge based on reputation.

the classification in Table 3.1 offer the opportunity to address this same challenge.

Following from definition 2.1.2 in Chapter 2 we can quantify trust if we can quantify its composing components: Opportunity, ability and competence, intent and persistence ⁵.

What common characteristics do OSNNs web communities have, and how do they contribute to the trust composing components?

Table 3.2 shows a short extract of examples of OSNNs in the top three categories, common attributes found in each of them, and how these attributes contribute to the three trust composing components.

Table 3.2 can be read as follows: Agent X evaluates Opportunity when deciding if to trust agent Y about situation τ ⁶ based on the location of Agent Y , its declared availability,

⁵I have simplified the wording from the definition in [19]; however, their use is intended to mean "the degree of credibility of agent X 's belief ..., represented by the parameters α , β and γ that are subjective to agent X "

⁶situation τ is represented as the duality of action α to achieve goal g in 2.2

its activity on the website and last time agent Y has updated its profile. A number n of attributes O contributes to component Opportunity, and each contribution is subjective to the Agent x by coefficient α .

$$Opp_{x,y,\tau} = \sum_{i=1}^n \alpha_{i,x,y} O_{iy} \quad (3.1)$$

The indication $\{x,y\}$ to indicate Agent X and Agent Y will be dropped and is considered implicit from here on unless otherwise specified. We can represent Opp as $Opp_{\tau} = \sum_{i=1}^n \alpha_i O_i$. Following this representation for the other composing components of trust (ability and competence AC , intent and persistence IP), we can rewrite (2.2) as follows:

$$DoT_{\tau} = \sum_{i=1}^n \alpha_i O_i \cdot \sum_{i=1}^m \beta_i AC_i \cdot \sum_{i=1}^k \gamma_i IP_i \quad (3.2)$$

Where:

- DoT_{τ} , or explicitly $DoT_{x,y,\tau}$, is the degree of trust of Agent X in agent Y for situation τ
- $\sum_{i=1}^n \alpha_i O_i$ is agent X 's belief in agent Y 's Opportunity to do τ expressed as the sum of contribution of the composing n attributes O (i.e. location, availability, last login, ...) each multiplied by agent X 's subjectivity coefficient $\alpha_{i,x,y}$ or implicitly α_i
- $\sum_{i=1}^m \beta_i AC_i$ is agent X 's belief in agent Y 's ability and competence to do τ expressed as the sum of contribution of the composing m attributes AC (i.e. qualifications, documents provided, number of reviews, ...) each multiplied by agent X 's subjectivity coefficient β_i
- $\sum_{i=1}^k \gamma_i IP_i$ is agent X 's belief in agent Y 's intent and persistence to do τ expressed as the sum of contribution of the composing k attributes IP (i.e. profile completeness, time to respond, tenure, ...) each multiplied by agent X 's subjectivity coefficient γ_i

The attributes are heterogeneous and take different forms (numbers, dates, categories, locations, etc.), and they need to undergo normalisation. For example, *Location* is

⁷ α indicating the coefficient of subjectivity for each attribute contributing to the opportunity element. Not to be confused with action α and goal g that are incorporated in τ

typically represented as a UK postcode for a precise location of the job provider (i.e. the Trustor) or a set of partial UK postcodes to represent an area. In Equation 3.2, the Location between Agent X and Agent Y about situation τ can be represented by the distance between the two Agents (the location of Agent X is also situational). In recent developments [40], it has been suggested that the inverse-square law can be seen as a universal law of human mobility. Using k/d^2 where k is constant (typically 100) and d is the distance might be a more meaningful representation since it accounts for a faster decay of trust with increased location (which [40] suggests a universal law of human behaviour).

Table 3.2: Online social networks of needs - Common characteristics

Example	Common Attributes	Contributes to
childcare.co.uk, doctify.com, carehome.co.uk, rover.com	Location, Availability, Last Login, Last Update, ...	Opportunity
	Qualifications, Documents Provided, Number of Reviews, Rating, ...	Ability and Competence
	Profile completeness, Time to respond, Tenure, Number of jobs...	Intent and Persistence
Category 1: Health and Care		
checktrade.com, trustatrader.com, ratedpeople.com	Location, Skills, Last Login, Last Update, ...	Opportunity
	Qualifications, Description, Number of Reviews, Rating, ...	Ability and Competence
	Profile completeness, Time to respond, Tenure, Number of jobs...	Intent and Persistence
Category 2: Property Care		
cottages.com, airbnb.com, vrbo.com, getaround.com	Location, Skills, Last Login, Last Update, ...	Opportunity
	Qualifications, Description, Number of Reviews, Rating, ...	Ability and Competence
	Profile completeness, Time to respond, Tenure, Number of jobs...	Intent and Persistence
Category 3: Property sharing		

The intersection between the three sets of attributes, respectively (Opportunity), (Ability and Competence), (Intent and Persistence) is not empty. In fact, the same attribute can contribute to multiple components. For example, *NumberofJobs* contributes both to "Ability And Competence" and "Intent and Persistence" at different subjectivity coefficients. An equivalent representation would be the union of the three sets of attributes. This doesn't invalidate equation (3.2) since the contribution coefficients will be 0 where there are attributes not contributing to composing components of degrees of trust (hence

contributing 0 to the sum and not affecting the product). We can define T as the set of attributes contributing to any of the trust components (i.e. Opportunity, "Ability and Competence" or "Intent and Persistence") where $T = OUACUIP$, and rewrite 3.2 in the extended form as 3.3

$$DoT_{x,y,\tau} = \sum_{i=1}^n \alpha_{i,x,y} T_{iy} \cdot \sum_{i=1}^n \beta_{i,x,y} T_{iy} \cdot \sum_{i=1}^n \gamma_{i,x,y} T_{iy} \quad (3.3)$$

The degrees of trust can take values from 0 (no information) to 1 (decision to trust). Analysing further the formula, how it behaves and the trust value association, we see the following cases arising:

- **The degree of trust is 0:** It only takes one of the factors of the degrees of trust to be 0 for the DoT to be 0. This means that all attributes need to be 0 since each of the components is the sum of its attributes which in turn are all non-negative values ≥ 0 . On the other hand, this situation is expected. If an agent doesn't have any information on another agent's ability to do the job, they are unlikely to proceed and trust each other.
- **The degree of trust is 1:** For the degree of trust to be 1, it will take all the normalised components to be 1. In turn, this means that for each component, the sum of the normalised composing attributes, adjusted with the belief components, should sum up to 1. This is also expected since in reality, absolute trust is a rare and unlikely event. Even more so, in the scenarios which we are considering here. The process of trust is rather a process of finding the agent who can be trusted *best*.
- **Low degree of trust values:** Once normalised, the three components of the degree of trust values are between 0 and 1. The multiplication yields DoT values that are rather close to 0 than to 1. In fact, the calculation in Chapter 7 shows a maximum DoT value 0.5 and a higher distribution for values closer to 0. The calculated DoT values can further be normalised between 0 and 1 for better representation.

We can use feedback on delegation to build a subset of data points where there had been a decision to trust, hence $DoT_{xy\tau} \approx 1$ ⁸. Since $0 \leq DoT_{xy\tau} \leq 1$, where $DoT_{xy\tau} \approx 1$, we can

⁸Whilst, as discussed in Chapter 3, the cooperation threshold is usually less than 1. You can delegate

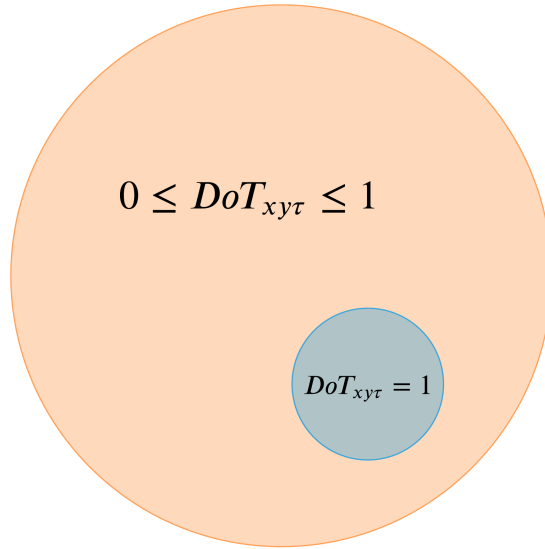


Figure 3.2: Degrees of Trust, Data-Points with high trust

assume that each component is equal to 1. That is $\sum_{i=1}^n \alpha_{i,x,y} T_i = 1$, $\sum_{i=1}^n \beta_{i,x,y} T_i = 1$, and $\sum_{i=1}^n \gamma_{i,x,y} T_i = 1$ of the product.

Whilst it might seem a strong assumption for each component to be equal to 1, it doesn't invalidate Equation (3.3) since the trust data points are an expression of maximisation of $DoT_{xy\tau}$ which in turn means the maximisation of each composing component.

Example 3.2.1. Assume we have the data of several trust exchanges that happened on childcare.co.uk. A number of service providers $y \in Y$ have been given a childcare job, and a review was provided after the fact. Let the list of job providers be $X = \{Alice(A), Bob(B), Charlie(C), \dots\}$ and the list of service providers be $Y = \{Eve(EE), Mallory(MA), Trent(TT), \dots\}$. Let A be the set of attributes: $T = \{Location, LastUpdate, NumberOfReviews, Tenure, \dots\}$. Furthermore, two additional attributes represent the time of the exchange and the rating as an expression of whether agent x 's trust in agent y was matched. Let the data observed be:

$S = \{(EE, A, \{W1^9, 2020-08, 22, 18, 80\}, 2021-01, 5),$
 $(TT, A, \{E1, 2020-09, 20, 16, 60\}, 2020-12, 4),$
 $(TT, A, \{AB12, 2020-09, 20, 16, 60\}, 2020-12, 4),$

without having complete trust in the trustee; we can assume this value is close or equal to 1.

⁹The first part of a UK postcode indicating the local region of the exchange

$fv^{(1)}$	1	0	0	...	1	0	0	...	0.3	0.3	0.3	...	12	0.1	0.3	...	8
$fv^{(2)}$	1	0	0	...	0	1	0	...	0.5	0.5	0	...	7	0.1	0.4	...	10
$fv^{(3)}$	0	1	0	...	1	0	0	...	0.5	0	0.5	...	2	0.8	0.1	...	6
$fv^{(4)}$	0	1	0	...	1	0	0	...	0.3	0.3	0.3	...	18	0.1	0.3	...	10
\vdots	0	0	1	...	1	0	0	...	0.2	0.2	0.2	...	22	0.1	0.3	...	6
$fv^{(n)}$	0	0	1	...	0	0	1	...	0.2	0.2	0.2	...	1	0.1	0.3	...	10
	A	B	C		EE	MA	TT		EE	MA	TT		Time	Loc.	Qual.		Target
	Agent X (Trustor)				Agent Y (Trustee)				Other Agents Rated				Trust Attribute Coefficients				

Feature Vector $fv^{(n)}$

Figure 3.3: Factorisation Machines and the Feature Vector from Degrees of Trust

(TT, A, {BR6, 2020-09, 20, 16, 60}, 2020-12, 4)}

The problem layout, with the Trustor evaluating many Trustees' attributes, to decide on whether to engage or not in a trustworthy activity, with sparse data-points¹⁰ in terms of expressions of trust, preferences and previous collaborations 3.2 lends itself to the use of factorisation machines [41]. The problem statement can be worded as "given a set of attributes for agent Y, a set of previous expressions of trust for agent X, what is the likelihood of X trusting Y for the situation τ (i.e. achieving goal θ by performing action α)"

Following Example 3.2.1, the input feature vector for a factorisation machine can be written as in Figure 3.3. The feature vector, for each row, represents the Trustor's direct relation to the Trustee (i.e. where there has been a direct interaction, w_i in the factorisation model) or indirectly because the Trustee has had interactions with agents with common trust traits. The feature vector includes the three components of degrees of trust that will reinforce or weaken the links based on the subjectivity of the Trustor. These are modelled by $\langle v_i, v_j \rangle$ in the factorisation machine model 3.4

$$y(\hat{x}) := w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle v_i, v_j \rangle x_i x_j \quad (3.4)$$

The input target vector is $DoT_{xy\tau}$ calculated by 3.3 after the trust event has happened

¹⁰As we will see in the following chapters, the expressions of trust data points are very sparse in comparison to the number of agents and attributes.

and where feedback has been given, hence taking into account the feedback of the Trustor on the Trustee after the event. The degree of trust, which takes values between $0 \leq DoT_{xy\tau} \leq 1$, is normalised between 0 and 10. Running the factorisation machine model will return an estimated $DoT_{xy\tau}$ in the form of $y(\hat{x})$ on all remaining space on 3.2. The degree of trust calculation is a local, centralised and scalar trust calculation in reference to 2.5 and hybrid trust evaluation in reference to 2.3. The second part of the calculation is placing it in a graph, looking at the effects on the graph, and comparing it with different trust metrics. In the next section, we look at the graph theory and propose a set of experiments.

3.3 Social Networks Analysis and Experimentation

Social Networks Analysis [42]¹¹ methods are typically used when dealing with social structures and data from online social networks. Trust data and social interactions are a good fit for network representation, and this is often the case, as discussed in 2.3 and 2.4. Most trust metrics rely heavily on network centrality measures¹².

An "Online Social Network of Needs" is a social network we can represent in a directed (potentially bipartite) graph form, with agents as nodes and interactions as links. The links are directed from the Trustor to the Trustee to represent that $Agent_x$ (the Trustor) trusted $Agent_y$ (the Trustee) and provided feedback for the performance of action α . Sometimes, the Trustee replies to the feedback given, and the bidirectional links represent this. The attributes described in Table 3.2 can be represented as node and link attributes. We have multiple sources for each category in Table 3.1, and providers are likely offering their services in various OSNNs. Whilst there would be challenges in entity reconciliation among different data sources, we can build separate graphs for each source. This representation, as shown in Figure 3.4, has the characteristics of a multiplex graph. We can use the multiplex graph theory to analyse emerging structures from the different trust scores. Some measures from graph theory and graph properties also provide meaningful trust information. For example:

¹¹Social Network Analysis is the process of investigating social structures through the use of network and graph theory. https://en.wikipedia.org/wiki/Social_network_analysis

¹²<https://en.wikipedia.org/wiki/Centrality>

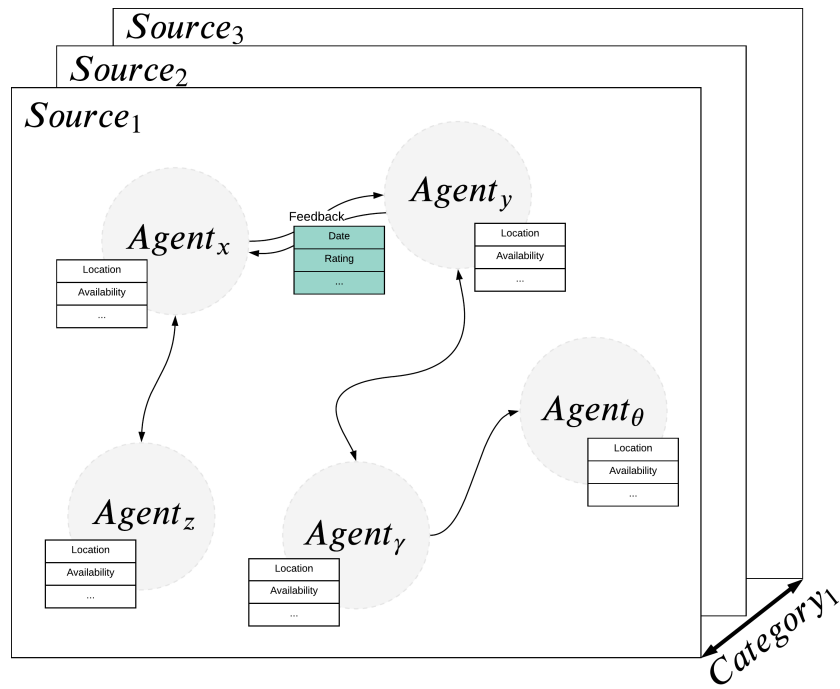


Figure 3.4: Multilayer network representation of Online Social Networks of Needs

- **graph density** (given by $D = L(G)/N(N - 1)$, where $L(G)$ is the number of links in the graph and N is the number of nodes) can further confirm the sparsity assumed in the previous section
- **degree centrality**, defined as the number of links from each node (given by $d(i) = \sum(j)m_{ij}$ where $m_{ij} = 1$ if there is a link from i to j , 0 otherwise) can be used as a trust metrics¹³ since it indicates the number of times an agent has been trusted
- **closeness centrality**, defined as the sum of the shortest distance between the node and all other nodes (given by $c(i) = 1/\sum_{i,j,i \neq j} d(i,j)$, where $d(i,j)$ is the length of the shortest path between node i and node j) might give indications of the impact of location on trust.
- **betweenness centrality**, defined as the number of shortest paths that pass through the node (give by $c(i) = \sum_{i \neq j \neq t} \frac{\sigma_{jt}(i)}{\sigma_{jt}}$, where σ_{jt} is the number of shortest paths between nodes j and t and $\sigma_{jt}(i)$ is the number of shortest paths between nodes

¹³Centrality metrics are often used as trust metrics as discussed by Di Meo et al. [43]

j and t that pass through the node i), gives indications on trust communities and their preferred trust choices.

- **eigenvector centrality** [44], often called "Prestige centrality", the eigenvector centrality¹⁴ is calculated on the adjacency matrix representing the graph. With Pagerank being a special case of eigenvector centrality for directed graphs, Eigenvector centrality is often used as a trust metric that is a function of the trustworthiness of its neighbours.

In addition to centrality measures, the information available allows for other social network analysis measures to be calculated and provided helpful insight into the network and how these influence trust formation. Some of these are

- Homophily, the measure of actors creating connections with their similar, can help evaluate how this impacts trust formation
- Multiplexity, the measure of creating ties in more than one area, can help confirm the expectation that trust can translate into continued collaboration
- Mutuality and Reciprocity, the measure to what degree actors reciprocate relations, a measure that can help evaluate the asymmetric characteristics of trust
- Network closure, a measure of transitivity, can validate the expectation of transitivity and decay of trust.
- Propinquity, the measure of the tendency of actors to have more connection with geographically close others, something we expect to be a strong underlying property (by definition) in OSNNs
- Clustering Coefficient, the measure of the likelihood that two associates of a node are associates

Other graph centrality metrics can be used, and scalar metrics (such as the degree of trust $DoT_{x,y,\tau}$) are projected in the proposed graph and compared for performance.

¹⁴https://en.wikipedia.org/wiki/Eigenvector_centrality

3.3.1 Multiplex networks on OSNNs

The social graph for OSNNs, as proposed in the previous section, has the individuals participating in the trust interactions as nodes and their expressions of trust and interactions as links. Other aspects of the graph, such as the locations where services are offered and the locations of the trustee or the types of services provided versus services required, are essential aspects that affect the number of links and their strength.

For a sound and comprehensive analysis, a fixed set of elements are represented by nodes and several types of relationships are represented by layers (essentially standard networks) of the multiplex.

Multiplex networks and multidimensional networks are a special type of multilayer network, which in turn are networks with multiple kinds of relations¹⁵. Unlike with multidimensional networks, in multiplex networks, only one link of a given type exists between two nodes. For example, when building a network of care providers, since there can only be one link between the provider and the receiver of care, we will represent reviews as nodes to account for the fact that, in reality, the receiver of care can repeatedly use the same care provider and leave multiple reviews. Since the reviews are unique, links will also be unique.

We expect this representation to provide a better narrative on the emerging patterns of trust where location, service types and other links play crucial roles. The multiplex graph in Figure 3.5 shows the different representations with nodes connected across plains, representing the fact that the same nodes will form further links. Additionally, each plane has also multiple graphs, indicating multiple sources of data as shown in Figure 3.4

Graph representations of social networks are generally considered part of complex network theory, and multilayered networks (sometimes called multidimensional network theory or multiplex graph theory) are part of it. The aspects of building the multilayered networks discussed here and shown in Figure 3.5 are somehow familiar in literature [45–47].

Some measures seen in the previous section are expanded to the multilayered networks and provide expanded details over the graph at hand (with the considerations over multi-dimensions). For example, in the multilayered graph, now represented as $G = (V, E, D)$,

¹⁵https://en.wikipedia.org/wiki/Multidimensional_network

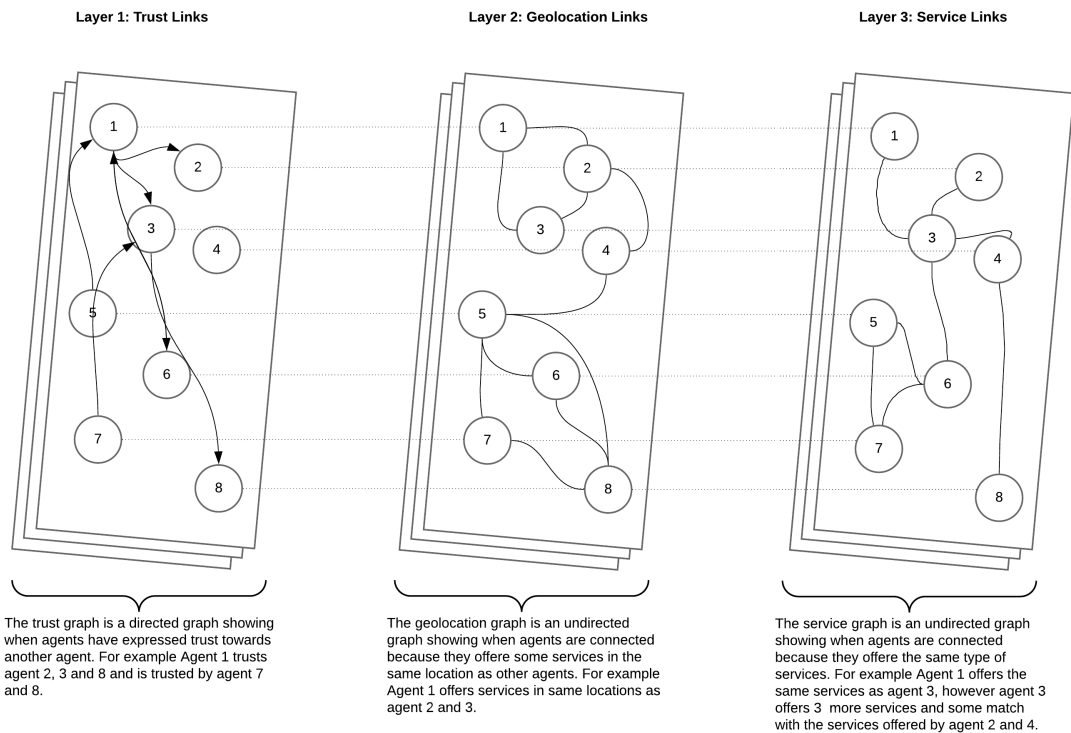


Figure 3.5: Multiplex graph for OSNNs

all the centrality measures provide additional information across all layers. However, they need to be calculated, taking into account the multiplexity.

3.4 Conclusions

We defined a category of social networks called Online Social Networks of Needs (OSNNs) that express and retain some fundamental traits of interpersonal and social trust. In OSNNs, interactions are between individuals, explicit and often happening physically in person (interactions between individuals are happening at the same time and place). Facets of trust are expressed explicitly via feedback, and trust is fundamental to being successful. Based on this definition, we have proposed six categories of OSNNs depending on the strength of trust required and represented in their interactions. These categories are: (i) Health and Care, (ii) Property Care (iii) Property Sharing (iv) Tutoring (v) Online Gigs and (vi) Skill Sharing.

Each category is characterised by a common set of attributes that contribute by a coef-

ficient α to either "Opportunity", "Ability and Competence", or "Intent and Persistence". We expanded on Castelfranchi's proposed degrees of trust 2.1.2 and proposed the Equation 3.3 for the quantification of trust based on this common set of attributes found in OSNNs.

We discussed the opportunity to use factorisation machines for predicting the likelihood of agent X trusting an agent Y for a situation τ based on their previous trust experiences with other agents and similar stations.

We proposed using complex networks, graph theory and multiplex networks to analyse the patterns of trust building on OSNNs and apply some well-known methods for social network analysis.

Chapter 4

Methodology

In previous chapters, we have laid down the context of this research, the research questions and challenges, and the theoretical framework for addressing them. Apart from the profound knowledge of the research topic, we are faced with a breadth of challenges in various disciplines to be able to make a meaningful contribution. Among others, we are faced with the challenge of identifying a number of sources that fall under the definition of OSNNs 3.1.1, classifying them according to the proposed classification scheme in Table 3.1 and identifying attributes (explicit, from metadata or derived) that contributed to degrees of trust 3.3.

Having identified numerous such sources, we are faced with the challenge of efficient data retrieval from heterogeneous and diverse sources. An important part of this research is focused on data cleaning and transformation before setting on qualitatively and quantitatively analysing the data.

In this chapter, we set to describe the research methods and tools used to achieve these goals, ethical and privacy considerations, and the reproducibility of the research.

4.1 Research Methods

In the following sections, we describe a combination of a variety of research methods needed to complete the research described in this report.

4.1.1 Observations and qualitative research

We set on this journey following the observation that a particular category of trust, interpersonal and social trust, can be found represented in a particular category of online social networks that we have defined and called online social networks of needs. The name comes from the fact that these have the characteristics of an online social network, but most exchanges are around particular needs.

(a) Identifying sources

In searching for sources, we limited our search to sources that bring together individuals and service providers in the United Kingdom. Whilst this helps limit the scope and focus of this research, we don't see this as a limitation, and the same methods can be applied to sources from other countries. Different categorisations have been proposed [48–51], generally applied to the sharing economy with the perspective of enabling platforms, value creation, services and non-service exchanges. The categorisation proposed in Table 3.1 is based on needs that are generally exchangeable peer-to-peer, with a common underlying trust need for the exchange to happen. Using Table 3.1, a number of search keywords are proposed in each category ¹. We used the following heuristic to search for data sources:

- i. For each category, use the keywords to search on conventional web search engines
- ii. For each result verify that they have the characteristics according to definition 3.1.1 and services are offered in the United Kingdom
- iii. Add the conforming sources to their belonging category
- iv. Stop when each category has at least 10 sources or when the web search doesn't have any more qualifying sources.

(b) Classifying sources

The source classification is somehow natural due to the keywords used in the web search

¹The full table of keywords is detailed in Chapter 6

categorisation. However, a further verification step was performed to check all sources manually towards the categories they were placed in. We used the following simple algorithm to validate data sources were placed in the correct category:

- i. For each category and each data source navigate to the corresponding web URL,
- ii. Perform a search for services and in the list of results verifies that the services offered belong to the corresponding category. Correct the placement when deemed incorrect,
- iii. Stop when all categories and sources have been verified

(c) **Identifying attributes**

In the previous chapter, we discussed the degrees of trust 2.1.2 as proposed by Castelfranchi et al. We proposed Equation 3.3 as a mathematical representation of trust composed of attributes and their contribution to the three components of trust: i. opportunity, ii. ability and competence, iii. intent and persistence. Each of the data sources has different types of attributes that we can group into two categories:

- i. **explicit:** these are attributes that are expressed in explicit forms, for example *location* or *review score* that are usually expressed in the form of a postcode or numeric form.
- ii. **derived:** these are attributes that are derived from other information. For example, the last time someone logged in or the sentiment score of the text in the reviews received is meaningful information to the degrees of trust.

Another consideration, whilst identifying attributes, is their availability among numerous data sources. In these terms, attributes can be classified into three groups:

- i. **global:** these are attributes that are found among all (or almost all) of the data sources. *Location* and *review score* are among global attributes.
- ii. **local:** these are attributes that are found among all (or almost all) of the data sources in a specific category, but not in the data sources in the other categories. For example, *availability* is an attribute commonly found in the *Health and Care* category but less so in the *Property Care* category.

- iii. **unique:** these are attributes that are relevant but unique to specific data sources in the category. For example, *Minimal age*, referring to the minimum age of the renter, is an attribute we find in peer-to-peer care hiring services (such as *getaround.com*) but not in other sources in the same category.

4.1.2 Non-empirical research in software engineering

Having compiled a large list of data sources (i.e. social networks that conform to the definition of OSNNs) in the various categories, and having identified numerous attributes, we are presented with another challenge: *how to efficiently, reliably and repeatedly collect information from the data sources?*

To address this challenge, we relied on non-empirical research in software engineering. The method followed, generally known as vision-strategy-execution-metrics [52] (shortly VSEM), is described in the following steps:

- i. **Vision and Mission:** A vision and mission statement to guide the overall research towards the goals we are trying to achieve. Whilst we will expand the full details of the vision and mission in the next chapter, a relevant short statement can be: *Create a system that allows academics to collect large amounts of data from social networks without the need for coding skills ...*
- ii. **Strategy:** Following the guidance from the vision and mission statement, we created a list of design questions and a strategy for addressing them. For example, to address the *"without need for coding skills"* aspect of the mission, we asked the following design question: *"how can we support a web query language in our system?"* to which the subsequent strategy decision is associated: *A modular system able to support multiple, configurable, existing or new web query engines.*
- iii. **Execution:** Having set several goals, design questions and strategies to address them, we set on the route of execution and software building. We use some known software engineering methods, building on top of state-of-the-art and well-known software frameworks and tools.
- iv. **Metrics:** We created several metrics to validate that the built software can achieve

the goals set. To validate the metrics, we use some case studies, one specifically designed to validate the metrics set.

4.1.3 Data collection and Case studies

Having built a system able to reliably collect information from social networks, specially designed for OSNNs, we set on building and running multiple data collection campaigns. Each data collection campaign requires the following steps:

- i. **Entity Discovery** Each data source has different ways of identifying Trustors, Trustees, and acts of trust. We wrote discovery web queries to identify a holistic view of all entities in each of the data sources.
- ii. **Attributes data extraction** Each entity representing Trustor, Trustees, or their relations has information scattered among one or more web pages. We wrote the web queries to extract the information for the attributes identified by the methods described in the previous section.
- iii. **Running data collection campaigns** For each data source, we run at least one data collection campaign. A data collection campaign is complete when all entities have been discovered and attribute information has been collected for each entity. Occasionally, we run continuous data collection campaigns amounting to hundreds of completed campaigns for a single data source.
- iv. **Validation and data correctness** Whilst most of the Validation and correctness of the data was done when building the web queries, on completion of the data collection campaigns, some further light-weighted validations were performed. The process consisted of selecting random records and manually validating that the information collected matched the information on the web pages it was collected from.

During this research, we used case studies on two different occasions. The case study method used is described in the following steps:

- Case study definition and objectives
- Options and candidate selection

- Data collection and analysis
- Results presentation and goals achievement

In conclusion, the case studies were used to describe in great depth the possibilities and the limits of the data collection software and the details of creating and running a data collection campaign. Since these two aspects are core to our research, the case studies show some of the more nuanced decision-making processes and challenges we faced.

4.1.4 Data analysis, quantitative and empirical research

In the previous sections, we explained methods used to define and collect relevant data and information. This section presents some methods used to make sense of these data in the realm of trust definition, as discussed in the previous chapter.

We have collected a large amount of data and started by analysing some of the **descriptive statistics** about the data collected during this research project. Having familiarised ourselves with the structure (or semi-structure) of the data from each of the numerous sources, we set on a process of **extract-transform-load** where data was cleansed of invalid values, collated on more regular structures, normalised and ready to be processed in the models of trusts discussed on the theoretical chapter.

As such, we used empirical methods to quantify the attribute belonging to parameters and components in the degrees of trust Equation 3.3. For example, to quantify the belonging parameters for "*distance*²" to "*Opportunity*", "*Ability and Competence*", "*Intent and Persistence*" we started with an equal split of 1/3 for each and adjusted the parameters every-time an expression of trust occurred by observing the data.

4.2 Research Reproducibility

During this research project, we took extra steps to ensure reproducibility. Since this research project spans multiple disciplines, multiple tools and mechanisms were used to achieve this goal.

- **Open Source** We designed and built a system for large-scale data collection. This system is open source, easily verifiable, and executable for any new goals or to

²The distance between the Trustee and Trustor for the specific trust situation

reproduce the results discussed in this report.

- **Data Collection queries** To reproduce the data resulting from the data collection campaigns, two artefacts are necessary: the **open-source system** and the **web queries**. The web queries are publicly published together with this report and can be found in the appendix chapters. Re-running the data collection campaigns will inevitably produce non-identical data sets since the web sources are very dynamic and information changes hourly at the very least. Additionally, the web structure also changes in time, affecting the web queries that need to be adjusted to new structures. To ensure that the campaigns are verifiable in a lasting manner, web archiving services³ can be used to verify the extraction results are the at points in time.
- **Open Data** All data used, resulting from the data collection campaigns, have either been published in research data repositories⁴ or made publicly available and described in the appendix.
- **Open Data analysis** All executable notebooks⁵ have been provided in the appendix, and links to executable, publicly available source codes have also been published.
- **Open graphs** The data transformation scripts for building networks and graphs from the semi-structured data, together with access to the graphs themselves⁶, are made publicly available and described in the appendix pages.

4.3 Privacy and ethical considerations

These pages often refer to the trust agents as "Trustee" and "Trustor" and the interactions between them as an expression of trust. We are aware that each of them is an individual interacting on the web, sharing information about themselves and interactions between them. As such, we took extra steps to be respectful and mindful of the sensitivity of the

³For example <https://web.archive.org/web/20200919231027/https://www.childcare.co.uk/profile/57873>

⁴For example <https://researchdata.bbk.ac.uk/id/eprint/154/>

⁵These are generally python Jupyter notebooks

⁶These are generally in a Neo4j graph database

information we deal with. Additionally, some of the information is personal identifiable information, governed by the General Data Protection Regulation (GDPR), retained in domestic law in the United Kingdom. To ensure we are compliant, the following methods have been part of our research project:

- **Public data** Any data we have collected and analysed is publicly available data that is accessible by anyone on the web, and that doesn't require registration or logging into private areas.
- **Anonymisation and Pseudonymisation**⁷ We have taken steps to anonymise the information made public as part of this research by removing any personal identifiable information where possible or pseudonymising it where it was crucial to the research objectives.
- **Encryption** We have enabled encryption for all data stored at rest. On most occasions, we used public cloud infrastructures that natively offer encryption at rest and generally a more mature level of data protection.

4.4 Conclusions

This chapter describes how we applied some of the consolidated research methods to achieve our research objectives. Various research methods are necessary, from observations to non-empirical software engineering, case studies and data analysis. We described how we paid particular attention to the reproducibility and privacy of the data.

⁷Both terms have specific definitions under GDPR. For reference <https://www.privacy-regulation.eu/en/recital-26-GDPR.htm>

Chapter 5

A distributed system for interpersonal and social trust data

5.1 Introduction

In this chapter, we propose a system architecture for collecting data from the web, focusing on online social networks of needs (OSNNs) where interactions are mostly face-to-face. Together with the architecture definition, we present an implementation of such architecture. We will discuss a case study to see the performance and limits of the proposed architecture and the system built.

5.2 Related Work

Whilst the architecture presented and implementation proposed in this paper are generally applied computer science on data scraping, data collection and generally data science, it has been built with online social networks of needs (OSNNs) in mind.

The WWW reached 1B registered websites in 2014 and is now approaching 2B ¹

Most estimations of the internet size are usually based on the number of indexed pages on the leading search engines. Counters are generally in the form of users, number of pages, number of websites, number of tweets, etc.

In reality, it is a non-trivial problem to determine the memory size of the internet. The situation becomes more challenging if we consider the deep web, which is usually estimated to be much larger than the visible web.

¹ca. 1.92B as of December 2021 according to <https://www.internetlivestats.com/>

Nevertheless, the indeterministic characteristic of the memory size of the internet, the number is bound to be large and ever-growing. The amount of data presents unprecedented opportunities for data mining and information extraction from the web. This has proven to be true given the number of scientific papers and research based on data from the web.

However, the web is unstructured. Previous tentatives to apply a machine-readable structure [53] to the web have failed to become large-scale standards. As such, in the modern days, data on the web are either made available by their owners in the form of temporal datasets or extracted using crawlers and scraper that leverage existing APIs² or public web pages.

Large mainstream online social networks and often well-established social media sites offer access to their data via APIs. Methods for leveraging API access for research purposes can usually be found in literature [54–57].

Even though data mining via APIs is the easiest way to access structured data directly, it comes with challenges and issues. For example, Pfeffer et al. showed how to tamper with twitter’s sample API [58]. Online social networks are massive, and APIs only allow for sampled or local³ data access. The locality is determined by the point of view of a particular profile, group, tweet, or hashtag. Hence, even when APIs allow access to structured data, we often find in literature alternative approaches. For example, to build a holistic view of the data on Facebook [59], Provetti et al. crawled 12.5M profiles on Facebook with a Breadth-First-Search crawler [60]. Of particular interest and related to this work are the study of Personalised Multi-Agent Systems (such as Amazon, eBay or Booking.com) and methods proposed for web scraping on such sys-

²Short for “*application programming interface*”

³The reach of the starting node usually determines locality, for example, friends on Facebook, or a sampled search of tweets with a certain hashtag

tems [61].

Web Scraping is widely used in the business world and for scientific purposes. De S Sirisuriya categorised the different techniques in the following groups [62]

1. Traditional copy and paste
2. Text grabbing and regular expression
3. Hypertext Transfer Protocol (HTTP) Programming
4. Hyper Text Markup Language (HTML) Parsing
5. Document Object Model (DOM)Parsing
6. Web Scraping Software
7. Vertical aggregation platforms
8. Semantic annotation recognising
9. Computer vision webpage analysers

In a more recent state-of-the-art analysis on web-scraping, Sarr et al. apply a different categorisation based on approach [63] with the following different approaches discussed.

1. Mimicry Approach
2. Weight Measurement Approach
3. Differential Approach
4. Machine Learning Approach

The considerations above need also be seen from another dimension. The web tends to be divided into three categories based on its reachability. When we discuss the web, we commonly refer to the "Surface web" that tends to be reachable from traditional mainstream web engines. However, an even bigger and more information qualitative [64]

part of the web is the "Deep Web". The Deep Web is usually hidden behind passwords, not linked to or many links deep that are difficult to reach with the traditional approaches of web crawling. Dedicated methods are found in the literature that addresses Deep Web data extraction. Of particular interest for our research is the work of Gottlob et al. [65] on OXPath - an XPath extension for web crawling and scraping that is particularly successful in extracting data from the deep web.

Another web section is the so-called "Dark Web", usually hidden behind private networks and accessible via VPNs. The dark web is infamously known for the number of illegal activities happening in its realm.

In conclusion, as discussed in other related works [63,66], we find a gap in both literature and existing commercial or open solution that offer the combinations of the characteristics discussed in this section. We will discuss the desired standard features and propose a solution in the following sections.

5.3 A Modern System for Web Scraping

Whilst approaching the problem of data retrieval from OSNNs (3.1.1), we realised that we needed a system that could harness data from APIs, Semantic web and the non-trivial case of scraping not machine-readable data. Occasionally, there are good reasons for scraping data even when an API is available [67]. Often, OSNNs are small to medium businesses without the visibility and the resources of mainstream social media⁴.

We focused our research on a sustainable architecture for web data collection campaigns that could be used to harvest data from OSNNs. As such, we started by setting up the following mission statement:

A system that is accessible, free and open source with a low learning curve, able to utilise modern tools and cloud technologies for researchers and institutions that need to harness the web for data at scale.

⁴This is usually the case for OSNNs that are locally focused or where the Trust demand is high. However, some more global OSNNs, such as Airbnb, have API access and public datasets that can be used for research

5.3.1 Architecture characteristics

We identified some architectural characteristics that should be satisfied for a long-term, sustainable implementation that proves successful and fit the purpose.

- **Queryable:** We believe that an architecture that can reuse existing technologies for querying web pages and extend it to harness data across pages and sites would be appealing to a wider audience. It can have better chances of universal applicability, sustainability and a lower learning curve. When looking at the different techniques suggested by [62] (5.2), apart from the semantic web, all techniques are either not repeatable (1, 6), require ad-hoc coding that mostly isn't transferable (2,3,4,5,9) or requires you to learn and use proprietary and costly platforms (6,7). Whilst we want a system that can use the semantic web, this, on the other hand, is not universal and often is not supported by OSNNs.
- **Scalable:** Online social networks and often the OSNNs are massive, with Facebook estimated to have 2.5B active users [68] (i.e. 2.5B or more active profiles). Hence, given enough resources, architecturally, the solution should be able to scale horizontally to harness social networks the size of Facebook. Efficiency and cost should be embedded in the architecture. Similarly, vertical scaling to both allow for the use of commodity hardware and potentially use the full extent of available resources, being mindful of resource starvation, should also be a characteristic.
- **Distributed:** Web Crawling architectures [69] are often described as a dual process of discovery (breadth) and data extraction (depth) where architectural choices are made around priorities (for example, breadth-first-search) and ordering of discovered URLs. As the system scales up, it is understood that multiple agents will run simultaneously. Distributed characteristics for synchronising, achieving a common goal, avoiding effort duplication and conflict and finally, a distributed storage to support storing of semi structured data are part of the fundamental architectural characteristics of the system.
- **Open Source and Extendable:** As stated in the goals, the intention is to fill a gap that has often been observed when approaching the issue of collecting data

from the web. Often the tools are either not free and commercial, or they rarely incorporate more modern technologies, scale and are extendable.

- **User Centric:** As discussed in [58] and [62], both APIs and different mechanisms for data scraping might have downsides and be inaccurate in terms of the data as seen by end users versus data retrieved. We believe in a WYSIWYG⁵ mechanism where the data collection is as seen by the end-user and not what is presented to a crawler agent, or what is provided via APIs.
- **Security, Privacy, and Policy Compliant:** The agents participating in OSNNs are, by definition, individuals requesting or offering services. We believe in a system that considers each subject's data security and privacy a core characteristic. Data encryption, aggregations, and anonymisation mechanisms must be at the core of the designed system. Furthermore, access to (public facing) web pages needs to comply with the usage policy of the individual websites⁶. We understand that a more pragmatic approach must be taken when dealing with individual policies and the full extent of privacy and security. Part of the compliance cannot be incorporated in the system design and must be considered by the specific implementation and system use.

5.3.2 Architectural design

In the following sections, we are going to describe the architectural design from different perspectives, aiming to exhaustively explain its goal and the means of achieving it.

5.3.2.1 Tenant View

We envision the resulting system being used by different groups of people that collaborate on the same projects. We call these tenants and diagram 5.1 shows how the usage of the system flows from the tenant prospective where:

1. **Tenants** have one or more users collaborating

⁵Whilst WYSIWYG (i.e. the acronym "what you see is what you get") is often applied to web development IDEs, in this context is a good description of the meaning "user-centric" being close to "as seen by the end user"

⁶These cannot be more restrictive than the legislation on the country where such web pages are being consumed

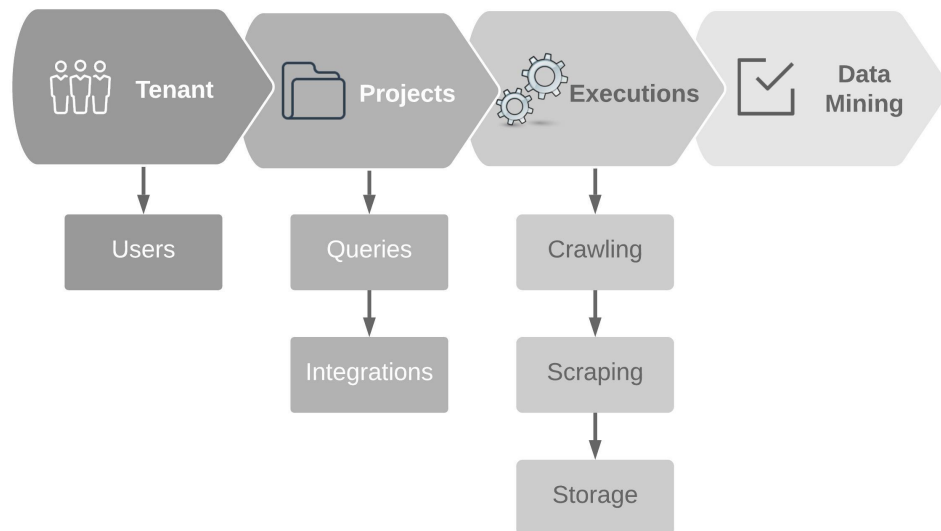


Figure 5.1: Tenant View

2. Tenants create **projects** and for each project define one or more **crawling queries** and one (and only one) **scraping query**. Queries can be generated at run-time and retrieved as part of an **API integration**
3. The system will schedule projects for **execution** and will execute asynchronously the **crawling queries**. Results will be ordered and input to the **scraping query execution**. Structured and semi-structured results will be **stored** in persistent storage.
4. Data is made available to the tenant and its users for **data analysis**

5.3.2.2 System Components view

From the system components' perspective, our system is not much different from other web data collection systems. These are often composed of a crawler (often called "web spider" or "data discovery component"), and a scraper (often called "data extractor component") [67, 70]

However, the characteristics described in Section 5.3.1 are inherently embedded in the component design 5.2. Some of these are:

1. Crawling and Scraping is query-based. A crawler or an extractor instance is launched against crawling or extraction queries.

2. Each running instance of the scheduler has distributed characteristics and will ensure that queries are run once, that there is agreement on the order of URLs to scrap and there is a locking and unlocking mechanism on URLs when the extraction is in execution, completed or timed-out.
3. The document storage must support storing semi-structured data and advanced data queries. This will be the basis for data mining and analysis on large documents that will be distributed across multiple physical servers.
4. There is provisioning for API Integration. This will be the basis for extended query techniques when dynamic queries are generated on demand or queries are generated by Machine Learning techniques that analyse the structure of the target pages.
5. The system is containerised and instances can be launched and added when demand grows. Each instance is self-synchronising.
6. Container orchestration is considered part of the system and crucial to deliver horizontal scalability and public cloud provisioning

In conclusion, the architecture presented in this section is optimised for scraping OSNNs based on some observed functional and structural commonalities that these types of networks have. OSNNs typically have:

1. web search that is based on location and often a service category
2. search results page that presents providers in a list and some key metadata such as price, distance, rating, sex, gender etc.
3. profile pages with full descriptive details
4. feedback, rating, and recommendation pages.

As such, we will call the proposed architecture "OSNNs Scraping Architecture" or briefly "OSNNs-ScrA".

5.3.3 An open implementation of data retrieval distributed system

Whilst pursuing the implementation of OSNNs-ScrA, we faced some non-trivial challenges to achieve what was set out in the architecture blueprint.

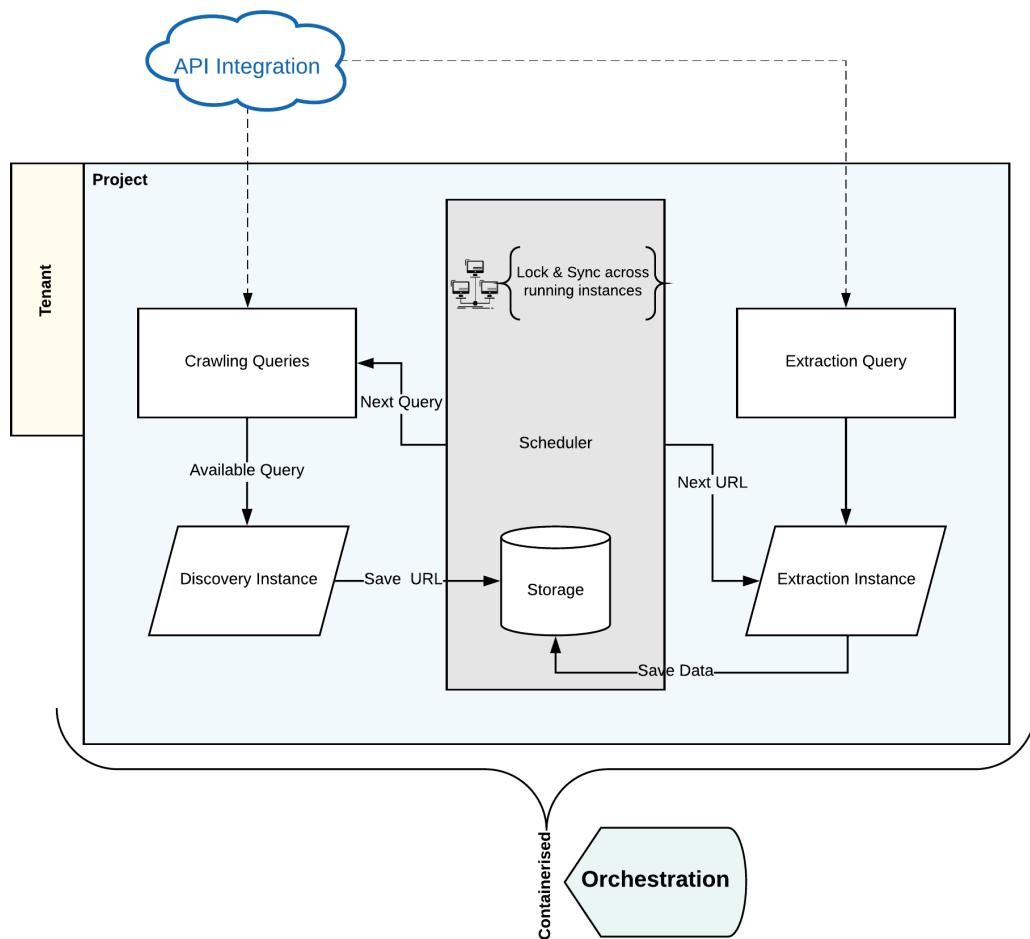


Figure 5.2: Component View

5.3.3.1 The queryable web

The concept of "queryable" is often tightly coupled with structure. Due to the success of the "Structured Query Language" [71], efforts to make the web queryable often took the form of SQL extensions [72].

However, that doesn't overcome the problem of "structure". The web is unstructured, with patterns emerging between sites of similar categories. The Semantic Web [73] and the query languages build on The Semantic web standard, such as RDF [74], are viable solutions to queryability and structure on the web. On the other hand, the Semantic Web has not been adopted as a standard at speed it was initially expected [75]. Much of the web is not structured according to the Semantic Web

standard.

Other approaches are derived from the markup nature of the web and the output as seen by the end users. Web users visually consume browser interpretation of HTML content incorporating other media, styling and JavaScript. Regular expressions, Document Object Model and HTML Parsing, are methods [62] that use the markup nature for searching, parsing and extracting content from web pages. In a comparison work [76] between Document Object Model (DOM), Regular Expressions (RegEx) and XPath [77], RegEx and XPath have similar performances in memory usage and speed of extraction and, unlike DOM, can be used as queries, be external, decoupled and instrumental to the web scraping implementation.

Between the two, XPath is a query language working well with markup constructs whilst Regular Expressions can easily grow in complexity [78] as you have to manage the flexibility and different styles of writing HTML (with multiple spaces, double quotes, single quotes, no quotes, in one line, in multi-lines, with inner data, without inner data). The superiority of XPath to Regular Expressions (and DOM, to the extent that DOM is an unsuitable choice) translates into XPath as the preferred choice on our core web extraction engine.

- **XPath - XPath Queryability**

The efficiency of building an XPath-based query engine for extracting data from the web has already been shown with XPath [79]. XPath is open source, and we will use the command line interface⁷ as one of our query engines for our implantation.

Research on XPath and its implementation has been carried out at Oxford University at the beginning of the last century. The XPath implementation is also used commercially and quoting from the open-source GitHub repository⁸

Meltwater uses XPath to extract millions of documents from 100'000s of sources daily.

⁷<https://sourceforge.net/projects/xpath/files/xpath-cli/1.0.1/>

⁸<https://github.com/xpath/xpath>

OXPath has fundamental characteristics that make it a preferred choice to other similar web scraping tools. These have been discussed largely in [65]. (a) The OXPath Language construct is a superset of XPath. XPath is an established query language for markup constructs such as XML or HTML. (b) OXPath supports "Kleene Star" navigation and follows up actions with additional constructs for termination conditions. (c) Extraction markers are embedded in the definition and transparent to the overall construct. (d) Support for actions and user interaction simulation, and (e) Full support of the XPath node navigation functions.

Whilst the OXPath implementation⁹ has today shown its limitations (for example, it being tightly coupled to a specific browser version), it also provides further opportunities for improving on scalability, sustainability and performance. OXPath is used in automatic full-site extraction [80], redundancy driven data extraction [81] and browserless web data extraction [82].

A second alternative engine that was implemented as part of this research project was used for some of the web data extraction campaigns. On some occasions, we found alternative features that might be needed which are not part of the scope of OXPath.

- **Data Retrieval Web Engine - JSON Queryability**

During our research and building OXPath queries, we found ourselves in the following situations where OXPath wasn't the best choice.

- (a) **Simpler request** - The navigability of the resulting query didn't require actions and user interaction or browser rendering, but rather simpler requests and link follows would fulfil the need
- (b) **Pre-Actions and Batch Requests** - The need to perform pre-actions (such as logging in) and then retain user cookies for a batch of links provided in input.
- (c) **Support for Modern Browser** - The latest OXPath CLI was built in 2017. It has embedded gecko drivers¹⁰, selenium and Firefox versions that are, at the time of writing, over five years old. More modern web technologies sometimes have

⁹<https://github.com/oxpath/oxpath>

¹⁰<https://github.com/mozilla/geckodriver>

unexpected or unsupported behaviour. (see "case-study" A Case Study - 4chan data collection)

- (d) **Closer to output format** - XPath was built with XML in mind, and its primary output is XML format. We have seen in the last decade a shift in web technologies from mainly XML-based (SOAP, Web Services, XHTML) to JSON-based (REST APIs, JSON+LD, GraphQL). In our architectural blueprint, by design, JSON is our primary output. There might be a need for a query definition that is closer to the output. Additionally, JSON has been successfully used as API query definition, for example, for API queryability in GraphQL [83]

With this in mind, we build an open-source python package that supports JSON queries, uses the latest gecko driver, browsers and Selenium, supports pre-actions and batch requests and, lastly, can either use Selenium or simpler requests where support for user actions and browser rendering is not needed. The package, called **Data Retrieval Web Engine**, can be installed as a python package '*pip install dr-web-engine*'¹¹ or its source code can be cloned from GitHub: <https://github.com/ylliprifti/dr-web-engine>

The following example query (file name google-example.json):

Listing 5.1: google-example.json

```

1 {
2   "_doc": "https://www.google.com/search?q=Donald+Duck",
3   "links": [{
4     "_base_path": "//div[@id='search'] [1]//div[@class='g'] ",
5     "_follow": "//a[@id='pnnext'] [1]/@href",
6     "link": "//div[@class='rc']/div[@class='r']/a/@href",
7     "title": "//h3/text()"
8   }]
9 }
```

For comparison, the above query, is equivalent to the following XPath Query:

¹¹Latest version on the pip repository <https://pypi.org/project/dr-web-engine/>

Listing 5.2: XPath equivalent query

```

1 doc("https://www.google.com/search?q=Donald+Duck")
2   /(//a[@id='pnnext'] [1] / {click /}) *
3   //div[@id='search'] [1] //div[@class='g'] :<links> [
4     .//div[@class='rc'] /div[@class='r'] /a/@href :<link=string(.)>
5     [? //h3/text() :<title=string(.)> ]
6   ]

```

It can be executed using the following command line:

```

1
2   # installation
3   python3 -m pip install dr-web-engine
4
5   # query execution
6   python3 -m web_engine.runner -q google-example.json

```

And it will produce the following output:

Listing 5.3: google-example output

```

1
2 {"links": [{"link": ["https://en.wikipedia.org/wiki/Donald_Duck"],
3           "title": ["Donald Duck - Wikipedia"]},
4           {"link": ["https://cosleyzoo.org/white-pek-in-duck/"],
5           "title": ["White Pekin Duck - Cosley Zoo"]},
6           {"link": ["https://www.cheatsheet
7           .com/entertainment/donald-duck-turned-85-years-old.html/"],
8           "title": ["Donald
9           Duck Turned 85-Years-Old and Disney Fans Are Quacking ..."]},
10          {"link": ["https://en.wikipedia.org/wiki/Daisy_Duck"],
11          "title": ["Daisy Duck - Wikipedia"]},
12          {"link": ["https://www.headstuff.org/culture
13          /history/disney-studios-war-story-donald-duck-became-sgt/"],

```



```
11     "title": ["Disney Studios  
12     At War - the story of how Donald Duck became a Sgt ..."]}  
13     ...
```

5.3.3.2 The storage engine

The storage engine is another core component of the OSNNs-ScrA implementation, where different options are available, each with its benefits and pitfalls. The following are some of the characteristics derived from the architecture blueprint that helped in the storage engine selection.

- Scalable, Distributed and available in the Public Cloud
- An advanced and mature NoSQL query engine
- Open Source
- JSON¹² Document-based for supporting semi-structured data

It is understood the world of NoSQL databases is very dynamic and ever-changing. However, the choice of an Open-Source Document DB reduced the selection to two main candidates: MongoDB and CouchDB. Existing literature [84, 85] comparing the two points to MongoDB having better performance.

We decided to support MongoDB Clusters for data storage that was external to the containerised solution but a dependency on it. We understand MongoDB (or any other storage choice) will become tightly coupled with the rest of the implementation, especially as specific CRUD queries are tailored for the selected store. Special care was taken in this implementation so that SOLID principles were followed, and a different interface implementation is always possible.

¹²JavaScript Object Notation (JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serialisable value). <https://en.wikipedia.org/wiki/JSON>

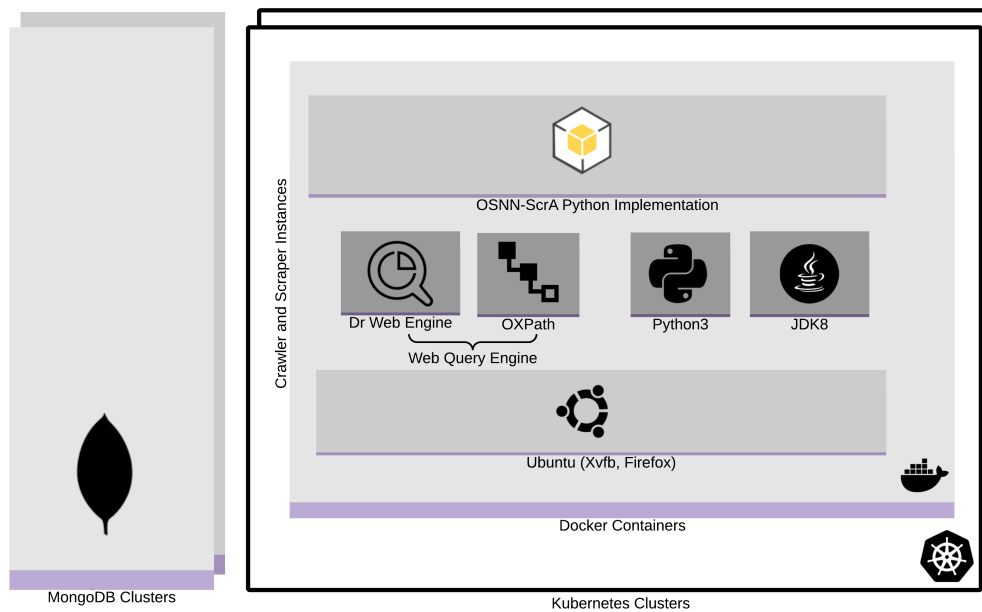


Figure 5.3: Technology Stack

5.3.3.3 Technology stack

We describe the technology stack in **Figure 5.3** following a top-down approach (i.e. from the cluster management into the running instances of our scraping implementation) going through an on-premises deployment on Birkbeck, University of London - Department of Computer Science infrastructure.

- **Kubernetes Clusters**

We envision a system that spins up multiple instances of scraping or crawling based on different projects and queries. Each instance will have specific needs for resource, speed, and repetition¹³. To facilitate container orchestration and the spinning up or down running instances, we distribution was implemented to support the deployment of one or many Kubernetes [86] clusters supporting the scraping of one or more projects, potentially using cost-efficient commodity hardware [87]. In the case study, which is explained in detail in the appendix, we deployed three very different Kubernetes clusters for the

¹³Some highly-dynamic and ephemeral sites have short-lived content. To acquire a "workable" view, there needs to be crawling and scraping at high speed

distributed execution of a single data retrieval project.

Docker [86] is a mature open-source containerisation tool that works well with Kubernetes and is widely used in the public cloud. We used a docker containerised environment running Ubuntu 18.04. OXPath uses the Selenium Web Driver¹⁴ which in turn uses Firefox¹⁵ and both are incorporated into OXPath build. The X-Video Frame Buffer (Xvfb) is used for silent runs that do not produce visible visual content. A snippet of the Dockerfile used to create the containers has been included in 5.3.3.3

Listing 5.4: Dockerfile code sniped

```

1  ## START INTERMEDIATE
2  FROM ubuntu:18.04 as intermediate
3  LABEL stage=intermediate
4
5  # Take an SSH key as a build argument.
6  ARG SSH_PRIVATE_KEY
7
8  # Add bitbucket to our list of known hosts for ssh.
9  RUN mkdir -p /root/.ssh/ && \
10     echo "$SSH_PRIVATE_KEY" > /root/.ssh/id_rsa && \
11     chmod -R 600 /root/.ssh/ && \
12     ssh-keyscan -t rsa bitbucket.org >> ~/.ssh/known_hosts
13
14 ## Get runnable
15 RUN chown -R dcs-service:dcs-service /home/dcs-service
16 RUN git clone --single-branch --branch develop \
17     git@bitbucket.org:yprifti/data-gather.git \
18     /home/dcs-service/Dev/data-gather --depth=1
19
20 ## START MAIN
21 FROM ubuntu:18.04
22
```

¹⁴<https://www.selenium.dev/>

¹⁵The browser can be overridden, and an alternative browser path can be provided

```
23 ##use /bin/bash as shell to allow use of 'source'
24 SHELL ["/bin/bash", "--login", "-c"]
25
26 ## ... sections of Dockerfile have been intentionally omitted ...
27
28 RUN chmod +x /home/dcs-service/Dev/data-gather/runner
29 ENV PATH="/home/dcs-service/Dev/data-gather/:${PATH}"
30
31 #ARG RUNNER
32 WORKDIR /home/dcs-service/Dev/data-gather
33 ENTRYPOINT ["runner"]
34
35 CMD ["cc"] #Override by k8s deployment configuration
```

5.3.3.4 Code and Integration

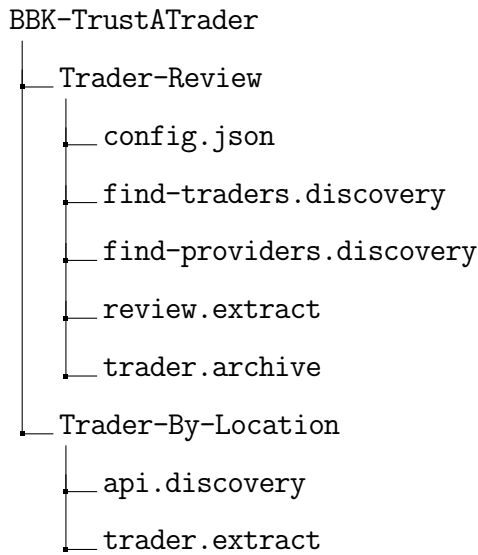
We have introduced several modern components together with several novel architectural aims. Our efforts are ultimately achieved by the core implementation (i.e. the python implementation of the system) that integrates all components and builds the end-to-end user flow from a query into a semi-structured database in MongoDB.

- **Project Configuration module - The lights and bulbs**

In OSNNs-ScrA, a project configuration is defined by its folder and file structures. Multiple methods can be used to hook the folder structure into the running instances. In our implementation, the project folder is mounted as a docker volume into the running instances.

We will use the following hypothetical example to illustrate how a project is defined by its folder structure and files. A project called "BBK-TrustATrader" will collect two sets of data (collections) called "Trader-Review" and "Trader-By-Location", each respectively having their queries for crawling and extracting data will be represented by the following

folder structure:



The project configuration for the above project will be loaded in an entity and be injected across the core of the implementation to indicate the following special meanings:

- *Project:*

A project is a hierarchical structure that can have one or more data collection campaigns in it. For each project, a new MongoDB Database will be created.

- *Collections:*

A collection represents a data collection campaign and each collection have its own set of queries. A collection is the runnable entity out of each data collected. Each collection can override some of the system configurations in a file called *config.json*. For example, a new MongoDB collection will be created for each collection defined in the project and the MongoDB server can either be defined in the system *config.json* or overridden in each project collection.

- *Queries:*

There are two types of required queries that each collection needs to define. These are 1. **Discovery Queries** - OXPath queries that will be used to crawl and discover new URLs and added to the extraction queue and 2. **The Extraction Query** - one OXPath

query that defines how data will be extracted from each URL discovered. Optionally, an **3. Archiving query** can be provided for projects that takes multiple snapshots of the same page. The archiving query is an XPath query that looks on a page for an indication that it has been archived and can be removed by the extraction queue. Furthermore, each of the queries can be provided dynamically via an API hook. Whenever the query files start with an *"api."* prefix, it indicates that rather than the XPath query, the file contains the configuration on how to call an external API for retrieving an XPath query. The following is an example of an API configuration file and the XPath query returned. Each API call generates a different query, as in this case, the API execution iterates through a list of postcodes.

Listing 5.5: api.discovery

```
1 {
2   "endpoint": "https://
   extractor-api.azurewebsites.net/trader-discovery?round=round1",
3   "callback": {
4     "endpoint
   ": "https://extractor-api.azurewebsites.net/trader-discovery",
5     "method": "POST"
6   }
7 }
```

and the resulting JSON as returned by the API call is shown in the following listing. The XPath content have been intentionally omitted.

Listing 5.6: api.discovery API call return

```
1 {
2   "payload": {
3     "postcode": "OX13",
4     "trade": "Air Conditioning Specialists - Automotive",
5     "round": "round1",
6     "update_time": "20200501222948"
7   },
```

```

8   "xpath":"... omitted ..."
9 }

```

An example of a discovery query is shown in the following XPath query:

Listing 5.7: XPath discovery query

```

1   doc("https://www.trustatrader
2   .com/search?trade_name=Electrician&location_str=BR5")
3   //ul[@class="profile-cards__list
4   "]//a[@class="profile-card__heading-link"]:<links>[
5       .:<link=qualify-url(@href)>
6   ]

```

- **Query Manager - Working with XPath queries**

As seen in the previous section, the XPath queries incorporated in the project configuration are the template queries that will be used for data collection. However, these are not used directly during execution. This is because further transformation might be needed. These are highlighted in the architecture blueprint by the "next query" action. The next query action, depending on the type of query, might inject the next URL into the query or need to call another API endpoint to get the next discovery query to run. Once a query has been prepared and is ready for execution, this is saved in a temporary folder and its physical path is returned for execution. This process is managed by the XPath query manager module. A code snippet that shows how the query manager generates the next extraction query for execution and returns its physical temporary path is included in the appendix. In an attempt to keep the code snippets relevant to the section context, we have omitted the implementation of private methods.

One essential item injected in the query manager methods is the *URL*. The URL keeps its meaning as the resource identifier, hence any new URL discovered is added to the queue, or its attributes are updated if existing. The queue is processed by the distributed locker module to get the next URL for data extraction (i.e. scraping).

- **Distributed Locker - Coordinating the effort**

The OSNNs-ScrA is expected to run multiple discoveries and extraction¹⁶ instances across a number of nodes in a Kubernetes cluster, each running multiple nodes. For example, in the case study, we show a Kubernetes cluster of 6 nodes and each node with 16 GB of Ram we scaled each container to run 30 extraction instances and each node to run 10 pods (one container per pod). That means 1800 running extraction instances, each processing the URL queue, running the OXPath query against the next URL, storing data and moving to the next URL.

Each instance of the core system also runs an instance of the Distributed Locker - a distributed module that coordinates the queuing and de-queuing efforts so that the queue is prioritised, and each instance gets the next item, without duplicating the effort. The queue is persisted in the MongoDB store, which in turn is a distributed NoSQL database. The Distributed Locker interface is as follows:

Listing 5.8: DistributedLocker Interface

```
1 from interface import Interface
2
3 class IDistributedLocker(Interface):
4
5     def next(self, batch_size: int = 1) -> str:
6         pass
7
8     def next_archive(self) -> str:
9         pass
10
11     def current_campaign(self) -> int:
12         pass
13
14     def move_next_campaign(self):
15         pass
```

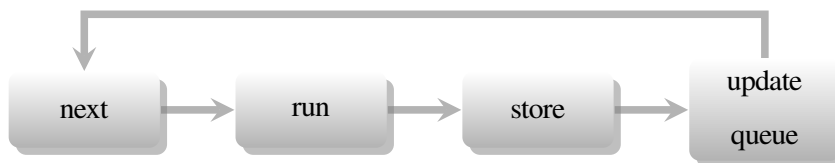
¹⁶Other types of instances may also exist and will follow the same patterns

Each running instance will call the distributed locker for the next item to process. Additionally, once the query has been run, the locker needs to update its state after the result has been stored. Hence, the locker instance is passed to the storage module.

There are two types of queues: i. the extraction queue and ii. the archiving queue. The archiving queue is optional and only used when an archiving query is provided for *continues campaigns*. These are long-running campaigns without a termination condition. As time passes by, some URLs might become obsolete, archived or deleted. The archiving query is used to identify these URLs, and the *next_archive* method is used to get the next item from the archiving queue.

A snippet of the interface implementation is provided in the appendix.

All the flow comes together in the following steps:



Listing 5.9: Running the core

```

1  locker = DistributedLocker(project_store, collection_name)
2
3  url = locker.next(batch_size)
4
5  extraction_result =
6  runner.extraction_runner(url, collection_name, project_config, False)
7
   project_store
   .extractor_store(url, extraction_result, collection_name, locker)

```

- **OXPath Runner and DRWeb Runner - Running the queries**

The **”Runner”** is ultimately the module that will reach the web at the next URL in the queue, retrieve the content, extract and structure the data into and make it available to other modules for further elaboration and storage. However, these can be done in multiple different ways and each will fundamentally change the behaviour of the system. The runner interface, included in the code section below, is composed of three methods. Each method is tasked with the execution of the corresponding query.

Listing 5.10: Runner Interface

```
1 from py.core
   .entity.ProjectConfiguration import ProjectConfiguration as Entity
2 from interface import Interface
3
4
5 class Runner(Interface):
6
7     def discovery_runner(self, project_configuration
   : Entity.ProjectConfiguration, collection_name: str):
8         pass
9
10    def extraction_runner(self, url: str, collection_name: str,
   project_configuration
   : Entity.ProjectConfiguration, clear: bool = True):
11        pass
12
13
14    def archive_runner(self, url, collection_name
   : str, project_configuration: Entity.ProjectConfiguration):
15        pass
```

We have made available two Runner implementations: (a) we have made use of OX-Path [65] and its CLI¹⁷ and (b) we have made use of the python module: Data Retrieval Web Engine¹⁸

¹⁷Command Line Interface

¹⁸<https://pypi.org/project/dr-web-engine/>

If the OXPath query in the **code snippet 5.7** was saved in the file at location `"/temp/temp-trader.discovery"` and the OXPath CLI runtime (footnote 7) was available at location `"/temp/oxpath-cli.jar"` the following would be a valid OXPath execution command:

```
1 java -jar /temp/oxpath
   -cli.jar -q /temp/temp-trader.discovery -f JSON -mval -jsonarr -xvfb
```

and the following JSON is the output of the execution:

Listing 5.11: JSON output of the OXPath query execution

```
1 {
2   "links": [
3     {
4       "link": [
5         "https://www.
6         trustatrader.com/traders/aa-electrical-services-electricians-bromley"
7       ]
8     },
9     {
10      "link": [
11        "https://www.trustatrader.
12        com/traders/taylor-made-electrical-services-ltd-electricians-bromley"
13      ]
14    },
15    ...
16    ,
17    {
18      "link": [
19        "https
20        ://www.trustatrader.com/traders/alpha-ohmega-electricians-greenwich"
21      ]
22    }
23  ]
24 }
```

We have already shown in 5.3.3.1 a similar execution using the Data Retrieval Web Engine module.

In the core implementation of OSNNs-ScrA, the XPath execution command line is generated and executed at runtime by the Runner Interface Implementation. For example, the code snippet below shows the implementation of the extraction runner that generates the XPath command line and executes and captures the output.

The result is a JSON (footnote 12) payload

```
return {"output": output, "error": error}
```

 that contains the JSON output of the XPath execution of the query or, in case of the execution terminated with an error, an empty output and the error message. The resulting JSON output is processed further in two ways: i. search for special attributes and ii. storage and queue update.

Similarly, we have kept the same form of CLI integration for the Data Retrieval web Engine, even though, given this is a python module, a more elegant way was possible.

- **Conventions and Metadata**

The output result of the XPath query execution is a JSON (footnote 12) payload that represents the semi-structured data extracted from the web page identified by the processed *URL*. The JSON structure of the data is defined by the XPath query. There are some conventions in regard to how the queries should be written. This is mainly to simplify the complexity of the task by using convention over configuration¹⁹. The conventions ultimately facilitate the post-processing and the evaluation of special attributes that are part of the output.

- **Conventions**

In previous sections, we have already seen some of the conventions. For example, the query file naming conventions help identify the query types by their file name extensions (**.discovery*, **.extract* or **.archive*) or by their prefix to identify API hooks (*api.**). In this section, we will discuss some of the query conventions. These are conventions that require the XPath queries to incorporate some special attributes in the JSON output.

¹⁹https://en.wikipedia.org/wiki/Convention_over_configuration

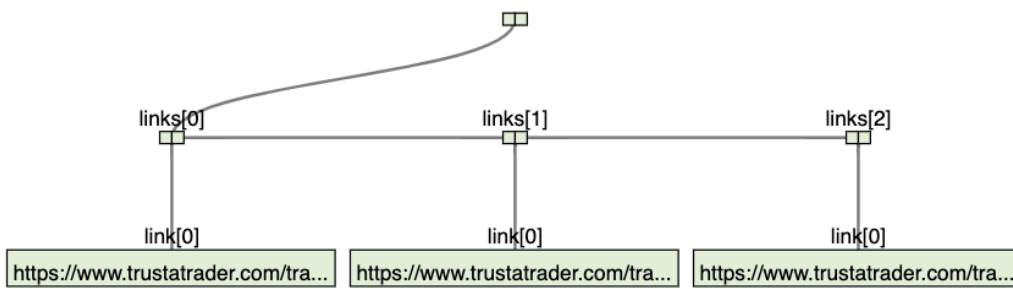


Figure 5.4: Discovery query output structure

Discovery queries conventions: The discovery queries are required to be written so that it generates JSON with the structure in listing 5.11. This is shown in the OXPath query in the **code snippet 5.7**. The output structure, represented by diagram 5.4, is an array of *links*, each containing one single *link*. The array will be post-processed and each link (i.e. *URL*) will be added to the scraping queue.

Archive query conventions: The archive query, an example shown in the listing 5.12, is used to identify URLs that are not valid anymore, either because they have been deleted or archived. The result is expected to have the structure in diagram 5.5.

Listing 5.12: OXPath archive query

```

1 doc("https://boards.4chan.org/pol/thread/218832673")
2   /.:<attributes>
3     [? //div[@class='closed']:<archived=string(.)> ]
4     [? //div[@class='boxbar']:<deleted=string(.)> ]

```

If either of the fields has a value, the URL will be removed from the queue and not selected anymore for extraction.

- **Metadata and post-processing**

Query conventions are mostly translated into a predefined structure in the output semi-structured data that are stored in our MongoDB cluster. Before the metadata is stored, it goes into a normalisation process. In the example in Figure 5.5, the attribute *”archived”*

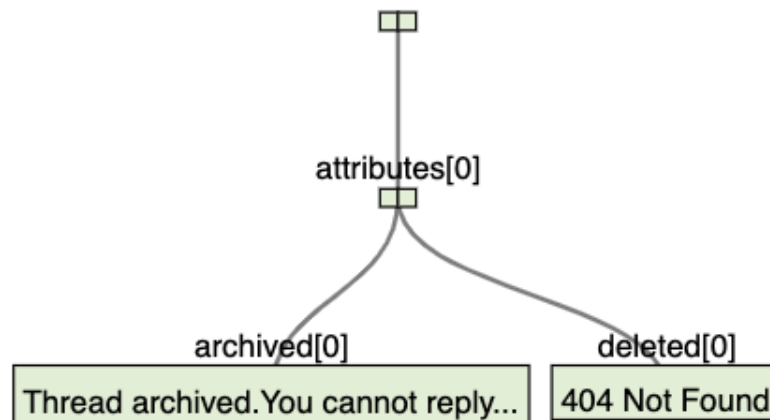


Figure 5.5: Discovery query output structure

can or cannot be present and when present can have different values. However, by convention, if present, whatever its value, it means the page is archived. During the post-processing, the metadata is always transformed in the following JSON structure:

```
{"attributes": {"archived": false, "deleted": false } }
```

The attributes *”archived”* and *”deleted”* are always present and boolean, but their values depend on the output of the archive query execution.

- **MongoDB Cluster, NoSQL, and Storage**

The storage system is external and central to the rest of the OSNNs-ScrA system, as shown in diagram 5.3. Whilst this has some downsides in terms of storage latency and decoupled system management, it immensely simplifies the synchronisation, locking and queuing algorithm among the distributed running scraping instances.

The additional system latency, due to the centralised and external storage, is highly negligible at a low scale. In fact, a scraping system is rarely a low-latency system. When low latency is required, the network latency, download of the remote web content, data extraction and structuring are by far slower than storage, when controlling for common parameters such as network and memory.

The decoupled system management (i.e. when scaling the system up or down

- you have to separately scale your scraping instances and additionally scale your storage cluster as the two are independent in our architecture) does contribute negatively to the system maintenance and administration. However, as we will see in the next section, the benefits overcome the additional burden.

- *Storage at high scale.* As the system scales up horizontally and writes take longer to propagate in the distributed storage system, we expect that the queuing, locking, and synchronisation would be negatively impacted. We dealt with this issue in the storage configuration. We used MongoDB in our implementation, configured to guarantee consistency but not availability. This configuration addressed issues with false positive URL locks in the queue (i.e. URLs that are currently being processed but not yet complete) and false negatives (i.e. URLs returned as free to process though already returned to another instance and the lock is being propagated) allowing for URL processing concurrency (i.e. when reading inconsistent data the queue might return the same URL to multiple instances resulting in wasted processing power). To deal with the sacrificed availability (i.e. as the storage system reaches consistency, it might not be available; hence, read/write operations might fail), we introduced a retrial mechanism in the core implementation such that each read/write was retried in a non-time linear fashion for a certain number of times before giving up. The implementation is shown in the code snippet A.5 in the appendix.

The Fibonacci and the "Power non-linear delay" methods have been included, together with a linear method and *NO_RETRY* option that will just exit after the first call.

- *NoSQL and coupling.* The API for storage operations offered by MongoDB is highly specific and specialised for MongoDB. The storage operations are highly coupled with the selected storage system. The use of interfaces, abstraction, and dependency injection helps to some degree decouple the core system from the storage. However, there are a number of assumptions that need to be incorporated in the interface definitions that are

part of the architecture blueprint. We have assumed that we will be dealing with a NoSQL distributed Document DB storage system and that the concepts of Database, Collection, Aggregation and CRUD operations are similar to the way they are defined and used in MongoDB. This is true for most of the DocumentDB databases we evaluated. The interface that incorporates our assumptions is shown in the code listing A.6 in the appendix. The concrete implementation will translate into the corresponding MongoDB client operations but also incorporate the *Delay Retry* calls as shown in the code snippet 5.13

Algorithm 1 Next URL algorithm

```

Initialize MongoQueueQuery
URLQueue ← run(MongoQueueQuery)
URLQueue gets URLs that are not deleted nor archived ordered ascending by last
extraction (i.e. last time selected) and limited to 1 item
if length(URLQueue) = 0 then
    Return and Exit
else
    URLQueue[0] ← mark()
    return URLQueue[0]
end if
  
```

Listing 5.13: Storage operations implementation snippet

```

1
2     def run_aggr(self, pipeline: list):
3         # self.col.aggregate(pipeline)
4
         return self.delay_retry.retry([self.col.aggregate], [[pipeline]])
  
```

A number of NoSQL queries are integrated into the concrete implementations of the storage operations and other modules that interact with the central storage system. For example, our implementation of the distributed locker is represented by the pseudocode 1. As seen by the algorithm, the goal is reached by the result of the query itself. Once a URL has been selected, it gets marked with the current date-time, and it will go to the bottom of the queue. The MongoDB aggregation query is shown in the code snippet 5.14

Listing 5.14: MongoDB Queue Operations

```
1 db.getCollection("childcare.co.uk").aggregate([
2     {"$match": {"$and": [{"$or": [{"attributes.deleted
3         "":{"$exists": false}}, {"attributes.deleted": {"$eq": false}}]},
4         {"$or": [{"attributes.archived
5         "":{"$exists": false}}, {"attributes.archived": {"$eq": false}} ]}}]}
6     ,{"$project": {"_id": 1, "url": 1, "last_extraction": 1}}
7     ,{"$sort": {"last_extraction": 1}}
8     ,{"$limit": 1}
9 ])
```

In the next chapter, we will discuss a case study to show how the system's deployment addresses a challenging data retrieval task.

5.4 Conclusions

We set on a task to solve the problem of data retrieval, having in mind simplicity and web queryability, understanding that any such system needs to take into account the size and scale of the task.

We designed and implemented a system that is composed of three key components:

1. Web query engines that make the system support web queries in input to achieve the data retrieval task. Queries can be written using the XPath query language (an extension of XPath) or using JSON construct as input to the dr-web-engine query engine.
2. A set of modern software, such as Kubernetes, Docker, and Helm, programmatically combined to achieve the goal of distribution and large-scale system. You can use commodity hardware to scale the system up. Whether adding nodes to existing clusters or additional new clusters, the new computational power can join the data retrieval query execution and pick up the tasks in sync with the existing nodes.
3. Natively distributed database system to store the semi-structured output of the

execution that can naturally scale as the data size grows. Our implementation supports MongoDB clusters.

In conclusion, as discussed extensively in the next chapter, we can run large-scale data collection campaigns based on web queries.

Chapter 6

Data

collection campaigns and case study

6.1 Introduction

In our journey to expand on the work of trust models in online social networks, focused on interpersonal and social trust, in Chapter 5, we expanded on addressing the data problem. Having defined the social networks of needs 3.1.1 and categorised them by the level of trust needed for an exchange to take place (i.e. cooperation threshold), we expanded upon proposing a distributed architecture for data retrieval, universally applicable to all social networks of needs in these categories. We concluded Chapter 5 detailing the achievement of such implementation.

In this chapter, we expand in detail on a case study of data retrieval running on such a system. The case study proposed is of particular interest for two fundamental reasons:

- *presents a challenging task in dealing with fast-paced ephemeral data and provides a benchmark for the performance of the system and,*
- *the selected source presents a view of the opposite spectrum of interpersonal social trust: an environment of complete anonymity, ephemerality, and mistrust. A breeding ground for conspiracy theories, racism, and misogyny to thrive.*

6.2 A Case Study - 4chan data collection

We have selected 4chan to use for our data collection campaign. In particular, the political board /pol from 4chan.org ¹ as a particularly challenging source for data retrieval.

4chan is an anonymous and ephemeral, topic-based opinion-sharing and discussion board. 4chan is associated with the birth of the hacker group Anonymous ² and has often been discussed in the literature associated with controversial topics such as conspiracy theories, racism, and antisemitism. The ephemeral and anonymous characteristics of 4chan make for two strong reasons for its selection as the source of our case study:

i.) **Data Retrieval:** Threads and posts on 4chan are short-lived, and usually, there is a very short window for scraping the threads and posts. Most of the studies on 4chan in the literature use the 4chan API³ for data retrieval during these short-lived windows [88–90]. We believe, whilst demonstrating the effectiveness and efficacy of the system discussed in this chapter, we can also:

- Provide an alternative dataset by using a different source (i.e. direct scraping from the web pages)
- Validate the information presented by the 4chan API and provide a more granular view than the API offers
- Provide a three-way comparison between the data as seen on the web pages, the 4chan API and the internet archiving service 4pleb ⁴. If 4pleb proved to be a reliable source of 4chan data, it would defy the ephemerality of 4chan threads and posts.

ii.) **Complete Distrust** Whilst information shared on 4chan is orthogonal to the trust discussed in this report and the definition of OSNNs 3.1.1, they surely represent strong characteristics of an environment of complete distrust.

¹<https://boards.4chan.com/pol>

²[https://en.wikipedia.org/wiki/Anonymous_\(hacker_group\)](https://en.wikipedia.org/wiki/Anonymous_(hacker_group))

³<https://api.4chan.org/boards.json>

⁴<https://archive.4plebs.org/pol>

To summarise the scale of the task, we list some relevant characteristics that need to be taken into account when running the data collection campaign.

- **Board structure**

Each discussion board, including /pol, is organised into a live board and an archive board. The live board includes threads (i.e. discussions) and posts (i.e. replies) that are open for discussion and new replies. The speed of the upcoming new replies also determines the longevity of each thread. Once no new replies are posted, a thread is moved to the archive board and the space is freed for a new discussion in the fixed-length live board.

- **Short-lived threads and pots**

In general, based on our observations and the literature [91], threads on 4chan /pol board can be short-lived and disappear from the live board within seconds. We have also noticed that short-lived threads, often, are deleted altogether and don't go to the archive board. Most of the datasets in the literature, that scrap the 4chan API every 5 minutes, have no visibility of these threads or any posts that might appear on them.

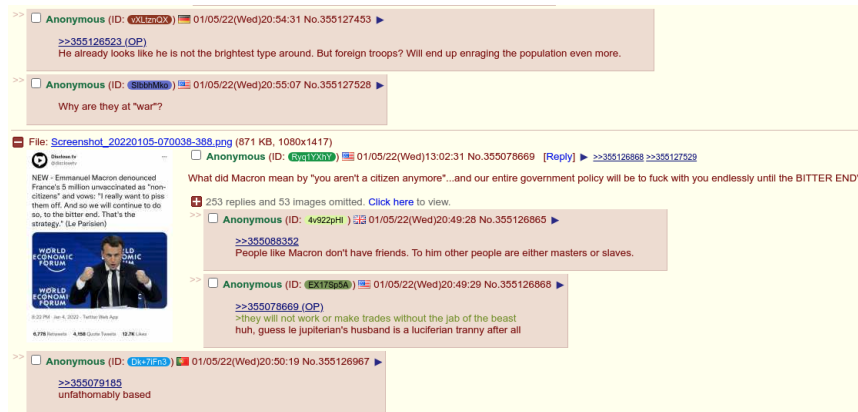


Figure 6.1: Screenshot from 4chan /pol board on 5th of January 2022

6.2.1 Data retrieval method

In this section, we are going to describe the method for retrieving data from 4chan. Among others, we are going to describe the web queries, the computational power and deployment method, scale and running instances, storage configuration and last but not least, how data can be accessed for further analy-

sis.

- **XPath Web Queries:**

To retrieve threads and posts from 4chan, we are going to write two queries: i.) The Discovery query: the query that allows for monitoring of the live board page and discovers new threads as they are posted on the board ii.) The Extraction query: the query that, given a thread and its URL, will retrieve all information from the thread, including the thread data and metadata and all replies (i.e. posts) on the thread.

The discovery query scans the web page in the live board page, `http://boards.4chan.org/pol/catalog` looking for all URLs under the HTML element *div* with *style class* "thread". The query is shown in the following code snipped 6.1:

Listing 6.1: XPath discovery query - 4chan.discovery

```
1 doc("http://boards.4chan.org/pol/catalog")
2   //div[@class="thread"]:<links>[
3     ./a:<link=qualify-url(@href)>
4   ]
```

To test the query and systematically build the more complex extraction query, the publicly available container published as a result of this research project can be used. To download and run a local copy of the container, we run the following docker and XPath commands:

```
1 docker pull ylliprifti/yp-phd-pub:pub-build
2 docker run -it --rm -v /temp/projects:/home
   /dcs-service/Dev/data-gather/projects ylliprifti/yp-phd-pub:pub-build
3
4 java -jar bin/oxpath-cli.jar -q projects
   /4chan/live-4chan/4chan.discovery -f JSON -mval -jsonarr -xvfb
```

The commands (which require a local installation of Docker⁵) in turn are going to perform

⁵<https://docker.com>

the following actions:

1. Download the docker image from the docker repository⁶
2. Run the docker image locally, mount the project folder into the container and launch a */bin/bash* console
3. Run OXPath and test the query

Snipped from the extraction query⁷, that is substantially longer than the discovery query are shown in Appendix A. All code for this case study can be accessed in the GitHub public repository, accessible at the following URL: <https://github.com/yprift01/4chan-data-project>

• **The storage system:**

We plan to continuously capture data from 4chan and store multiple versions of the same thread as it evolves from initial submission and grows with multiple posts (i.e. replies) until it is archived with its final version before finally it is deleted. We estimated storage growth on the order of 10GB/month. We intend to monitor the live threads no less than every 10 seconds. With about 100 running discovery instances every minute, each performing 200 reads and writes into the storage (i.e. the maximum number of live threads in the /pol board). At each discovery, every thread will either be inserted in the queue if it doesn't exist, or the timestamp of the last discovery on the live thread will be updated.

Very early in our research, we made the informed choice of fixing in 5 minutes the period of our cyclic extraction. Every 5 minutes, at least 200 instances of the extraction queries must be completed. Whilst most of the threads can be extracted in seconds, some threads can have thousands of posts, and the ex-

⁶The docker image is 1.5 GB in size and the first download might take a long time depending on the internet speed.

⁷A direct link to the query on the public repository for this project can be found here: <https://github.com/yprift01/4chan-data-project/blob/main/chan/live-4chan/4chan.extract>

traction query can take up to one hour to extract the data. To achieve the target of extracting data from threads every 5 minutes, we need to introduce redundancy so that extraction instances can pick a thread at least every 5 minutes. We empirically observed that running 500 extraction instances every 5 minutes guarantees the 5-minute time frame. This results in 100 extraction instances running every minute. Each instance runs a "read" for selecting the next thread to extract and a large write for writing the data back to the storage system.

The storage requirements are determined by the frequencies and volumes discussed above: we need the system needed to support about 500 read/writes per second, can grow by 10GB/month and is capable of supporting extended data analysis. Unable to make use of the cloud version of MongoDB due to cost and budget ⁸, we created an *on-premise cluster* made of 5 nodes configured as 1 arbiter node, 1 primary node and 3 secondary nodes as shown in the replica set members configuration file: https://github.com/yprift01/4chan-data-project/blob/main/mongodb/rs_members_config.json.

The cluster connection used for the project configured on *config.json* is shown in the following code snippet 6.2:

Listing 6.2: MongoDB Configuration

```

1  "mongo": {
2    "connection
      ": "mongodb+srv://<user>:<password>@rs.prifti.us/?readPreference
      =secondary&serverSelectionTimeoutMS=5000&connectTimeoutMS
      =10000&authSource=admin&authMechanism=SCRAM-SHA-256",
3    "config_db": "config",
4    "data_core_db": "data-core" }

```

The folder structure and the OXPath web queries determine how the results are stored and the output structure. In our case, the folders structure "*chan/live-4chan*" means a new MongoDB database named "*chan*" was created containing a collection named

⁸<https://www.mongodb.com/pricing>

"live-4chan" where all data are stored.

- **Computation power and Kubernetes clusters:**

To support the workload of the 4chan data retrieval project, we make use of five heterogeneous Kubernetes clusters made of 14 worker nodes in total with the characteristics shown in the following table:

Parameter	Value	Description
Control Plane nodes	5	There are five Kubernetes clusters with one control plane ⁹ node each that also serves as a worker node
Worker Nodes	14	There are 14 worker nodes, each of them running data discovery or extraction instances
Compute	38 GHz	There are about 38 Gigahertz of computing power among all nodes of all clusters
RAM	225 GB	There are about 225Gigabyte of RAM available among all nodes of all clusters
Disk Memory	1.2 TB	There are about 1.2 Terabytes of disk memory available among all nodes. This storage is only used as temporary storage and is separate from the permanent MongoDB storage.

A visualisation of the clusters, nodes and computational power is shown in the following image 6.2:

- **Execution and results:** Once we have tested the discovery and execution query, we are ready to deploy it into our Kubernetes cluster and start the data collection campaign. We used Helm Charts to create templates for data extraction campaign deployment [92]. The helm values for our deployment are published in the following YAML file ¹⁰.

We scheduled the **discovery** query to run 15 instances every 10 minutes on each cluster, totalling 75 instances over 10 minutes and averaging a complete discovery of live threads

¹⁰<https://github.com/yprift01/4chan-data-project/blob/main/helm/values.yaml>

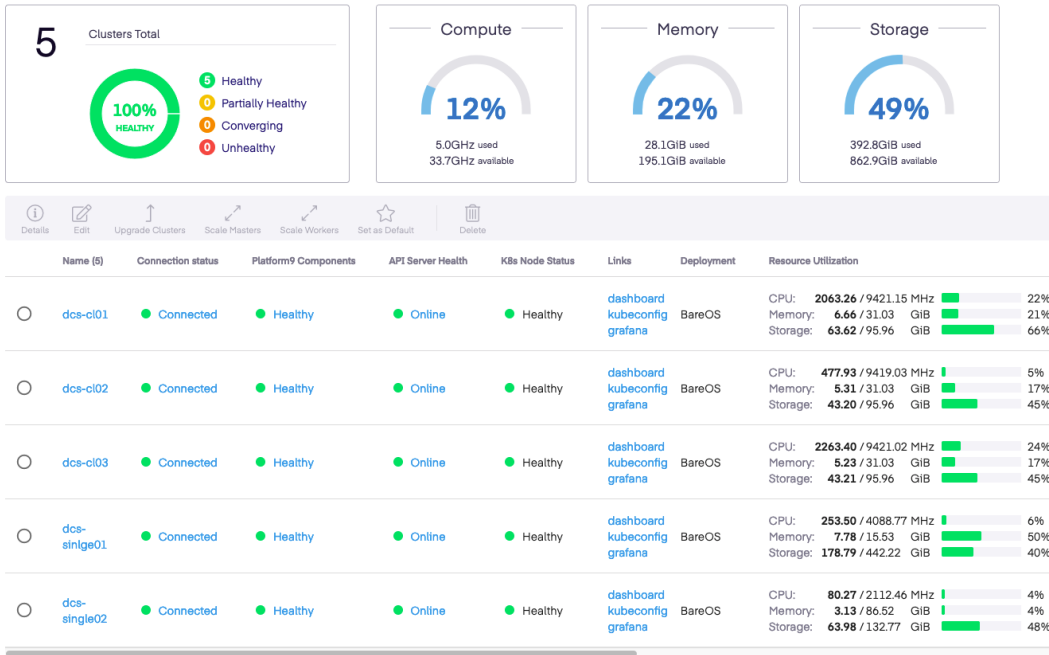


Figure 6.2: Kubernetes deployment

every 8 seconds, discovering about 200 URLs being published in the live board at each time. The high frequency is justified by the dynamicity of the live board, as elaborated in the next section.

We scheduled the **extraction** query to run 100 extraction instances every 10 minutes with a parallelism of 10 instances on each cluster, totalling 500 extractions over 10 minutes and averaging under one extraction per second. This high frequency of running extraction queries allows for multiple snapshots of the life of the thread and its evolution. Whilst no parallelism is needed on the discovery instances; each extraction instance will pick up unique threads; hence parallelism brings an advantage. More than 90% of the extraction queries would complete within 5 minutes. However, a small percentage would take more than one hour to complete the extraction. These are the threads that have many interactions (number of posts) and remain on the live board, sometimes for days.

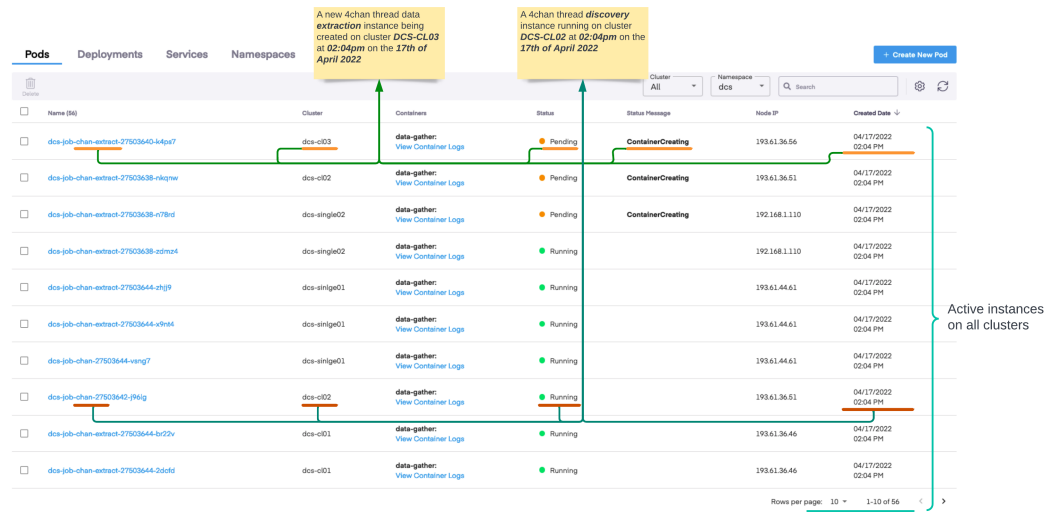


Figure 6.3: 4chan running view

6.2.2 Data analysis

We have been running different versions of the 4chan data campaigns for long periods of time, amounting to about three years of data, with some sporadic gaps in between. In parallel, we have been gathering the same information about the archived threads from the 4chan public API¹¹ and the 4chan archiving service: <https://4plebs.org>¹².

For this case study, we will limit to analysing results for a period of 6 months, from 1st of May 2021 to 30th of October 2021. This time-period was selected to optimise the amount of data available based on the number of discovery instances run over time. Figure B.2 shows the hourly number of instances over a period of about one year and a half from July 2020 to December 2021. The following table shows an overview of the content of the dataset:

¹¹<https://github.com/4chan/4chan-API>

¹²<https://4plebs.tech/foolfuuka/>

Dataset Properties		
Property	Value	Description
Period	1 st of May - 30 th of October	A total of 183 days
Nr. of threads	751,103	Unique 4chan threads discovered over the 6-month period
Nr. of posts	5,552,066	Total number of posts extracted over the 6-month period
Discovery Instances	300,226	Total number of discovery instances completed successfully ¹³ . See the distribution of the running instances over time in Figure B.2.
Data structure	See Figure B.1	Data structure of the dataset
Analysis Reference ¹⁴ : Jupyter Notebook		

Table 6.1: Data campaign output for the six months May to October 2021

The 4chan data campaign and the resulting data have the following notable features:

1. **Multiple snapshots:** The discovery instances run with high frequency and timestamp the threads every time they are found in the live board. The data extraction instances will extract data for threads as long as they are on the live board. This gives us a high precision of the threads to make it to the live board and an evolutionary view of their posts. We can look for anomalies in both the thread timeline (threads being removed from the live board) and the posts' timeline (posts and replies being deleted).
2. **Multiple sources:** Whilst running our 4chan data campaigns by, we also collected data from the 4chan API and 4plebs.org API. The 4chan API only provides thread data for as long as it is in one of the boards (live or archived), meaning you can only collect the information during the short lifetime of the thread. 4plebs.org is an internet archiving website for 4chan threads, though, to our knowledge, it has not been previously used or validated to provide an accurate view of 4chan.
3. **Live view:** The high velocity of the discovery and extraction on the live board gives us a "quasi-live" view of the data for analysis of what is going on in the world right now as viewed from the 4chan web community world view.

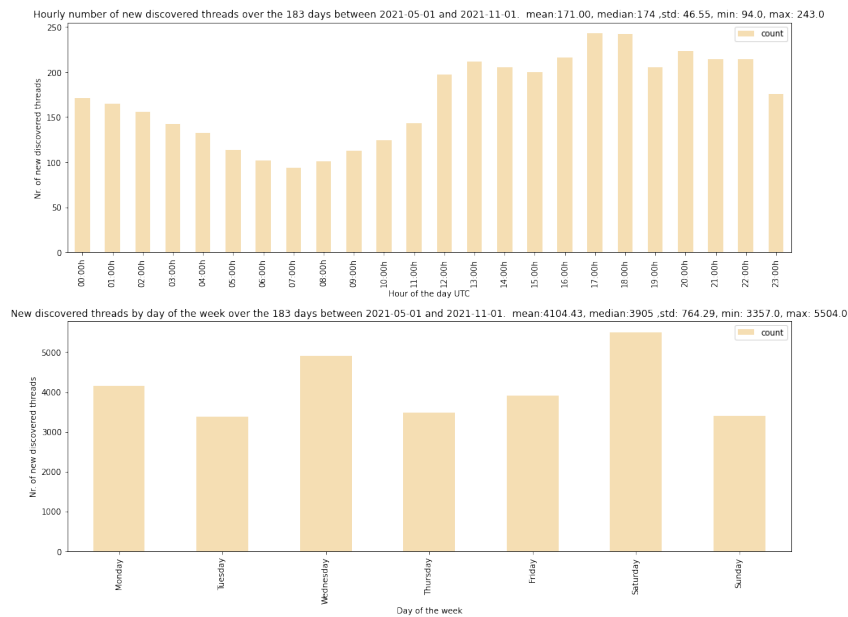


Figure 6.4: Distribution of newly discovered threads by the time of the day and day of the week

Over the six months, we can immediately see a higher activity between 12:00 and 22:00 UTC. These coincide with the working hours of the US East Coast (8:00 to 18:00 EST). The day-of-the-week distribution of newly found threads suggests Saturday at 15:00 UTC as the day with higher activity, followed by Wednesday at 10:00 UTC. On the opposite spectrum, consistently, 6:00 to 7:00 UTC is the period with the lowest activity across all days of the week, as shown in Figure 6.5.

About 70% of the threads have a short life and never get any replies (i.e. posts). The thread's life is directly related to the number of posts a thread receives, as shown by the logical regression in Figure 6.7. There are three natural data segments in the dataset.

- threads with 300 to 400 posts for which the linear regression or a mild negative quadratic regression would provide a model
- threads with up to 20 posts for which negative-coefficient linear regression intuitively works
- a scattered set of points with values ranging from 0 to 200 post as time unfolds. For these, we show, in yellow on Figure 6.7, the linear regression line but non-linear square fitting could also be attempted.

Of the remaining 30% of threads that got any traction, the most common life length of the thread is about one to two hours. Within 12 hours, over 99% of the threads that got any activity would have been archived, as shown in the cumulative distribution in Figure 6.6

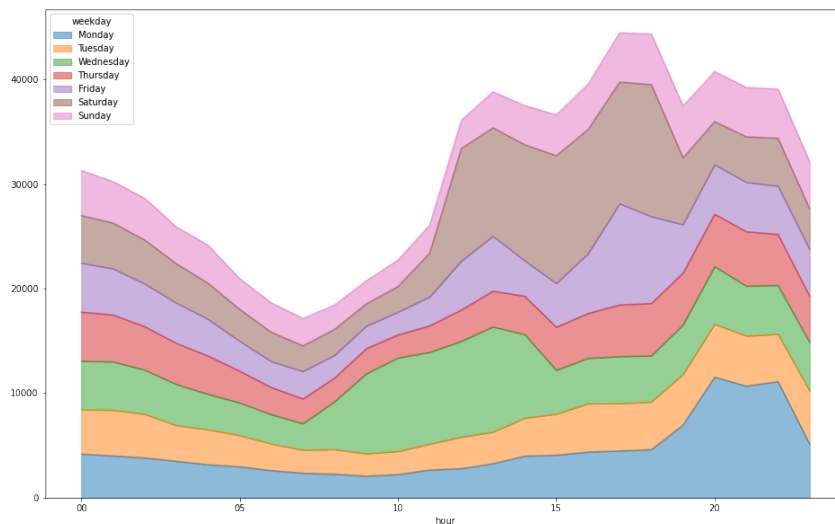


Figure 6.5: New thread activity comparison by the time of the day and day of the week

The statistical data above confirms what is largely known in the literature and discussed in papers such as [88, 93, 94]. On the other hand, the high frequency of the data collected and the multi-source allows us to look at some novel aspects of the /pol board 4chan.org and particularly /pol board.

Of the 751,103 threads discovered during the six months from 1st of May to 30th of October 2021, we have extracted data from the 4chan API and 4plebs API for 407,936 threads. The discrepancy is connected with the ephemerality of the 4chan API service, where threads are pruned after their time in the archive expires. Whilst we cannot draw any conclusions on the completeness of the number of threads exposed via the 4chan API, we can view and compare the completeness of the thread data between what is seen on the website and what is exposed from the 4chan API. Additionally, we can validate the completeness of the archiving service, 4plebs.

Figure B.2 shows the distribution of the number of discovery instances run over time. The number of matching threads we have collected data from all three sources (OSNNs-ScrA

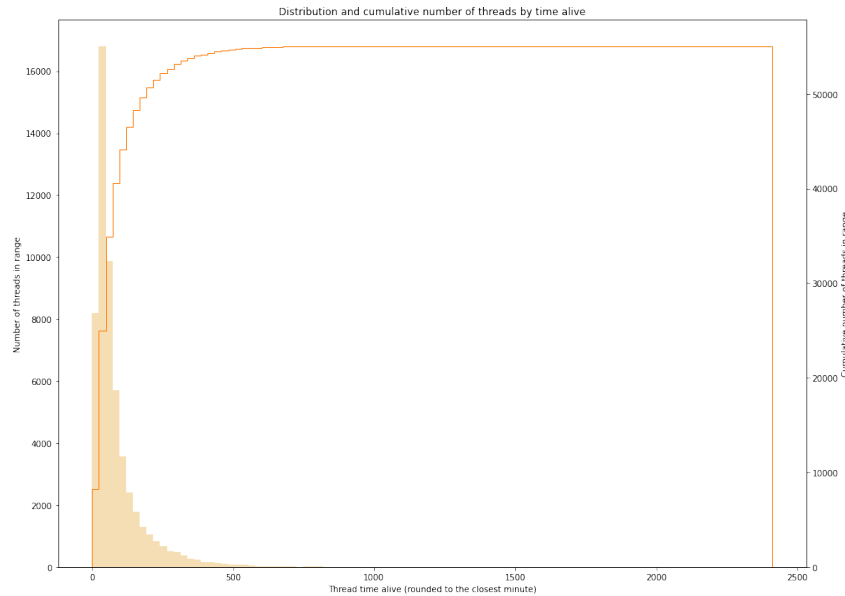


Figure 6.6: Cumulative Distribution - number of threads by the length of life

extraction, 4chan API crawler and 4plebs API crawler) is 103,322, amounting to about 14% of the total discovered threads during the 6 months in question.

Of the 407, 936 threads also crawled from APIs during the period in question, 406, 820 threads have been found in the 4pleb archiving API. This is equivalent to 99.726% match, with less than 0.3% potentially ¹⁵ missing threads on 4pleb.

Of the 1, 116 records that were not found on 4pleb, for 625 of them, we don't have any thread data (i.e. unable to find them via the 4chan API or via the OSNNs-ScrA extractor). We randomly manually tested 10% of these threads and for all of them, the 4pleb API returns ¹⁶ "thread not found". We believe these to be threads that have been deleted and removed at a faster pace for the 4pleb crawler to have collected any data on them. Of the remaining 491 records where we have thread data from either 4chan API or OSNNs-ScrA extractor (or both), when we manually and randomly tested about 10%, we found that most of them were now visible and accessible

¹⁵These can be false positives related to 4pleb API throttling or unreadability of extremely long threads

¹⁶Missing thread example see the following link: https://archive.4plebs.org/_/api/chan/thread/?board=pol&num=345390508

via the 4pleb API, and on the few occasions where we got an error, the error would read as follows ¹⁷: *"Threads this large can't be opened reliably. Please use the chunk API instead."*. In fact, most of them were big threads with hundreds of posts. We believe these to be missing due to a time gap between the time a thread is archived on 4chan and the time it takes to show on the 4pleb API for big threads. However, the number of missing threads on the 4pleb API is neglectable. These are either missing because they had no activity and were purged too fast from 4chan, or they took longer to be reflected on the 4pleb API but are otherwise now retrievable.



Figure 6.7: Logical Regression - Thread live time vs number of posts

We performed a three-way comparison of the thread data from 4chan API, 4plebs API and OSNNs-ScrA (i.e. threads as seen by end users on the browser). The columns and

¹⁷For example see the following thread: https://archive.4plebs.org/_/api/chan/thread/?board=pol&num=333614093

methods compared are as shown in Table 6.2

Three-way thread data comparison		
Property	Type	Comparison Method
Thread Number	Number	Key match across all three datasets
Author name	String	Direct comparison
Author country	String	Direct comparison
Subject & Comment	String	Levenshtein Distance [95]
Thread date	Date Time	Direct comparison
Number of replies	Number	Comparison with a relative tolerance of 1% and absolute tolerance of 1 ¹⁸

Table 6.2: Properties for the three-way comparison

The comparison results only show relevant differences in the number of replies compared to data scraped via OSNNs-ScrA (using the OXPath engine). This is to be expected due to the way the extractor is scheduled. Unlike the extractor from the 4chan API and 4pleb API that is extracting data once the thread has been archived (i.e. its final form), the OSNNs-ScrA is circulating through the live threads taking snapshots until they are achieved or deleted.

The results are shown in Table 6.3.

	4Chan API vs OXPath Extraction	4Chan API vs 4Plebs API	OXPath Extraction vs 4Plebs API
Author Name (d/m)	0 / 102862	0 / 102862	0 / 102862
Country Flag (d/m)	185 / 102677	252 / 102677	67 / 102677
Lev Distance (d/m)	506 / 102356	50 / 102812	470 / 102392
Nr Replies (d/m)	19530 / 83332	776 / 102086	18758 / 84104
Thread Date (d/m)	0 / 102862	0 / 102862	0 / 102862

Table 6.3: Three-way comparison results (d/m differences/matches)

As seen in Figure B.2, there are occasional extractor gaps, translating into the missing final form of the thread. The gap in the number of replies is only because, on these occasions, data are being compared with an incomplete view of the thread.

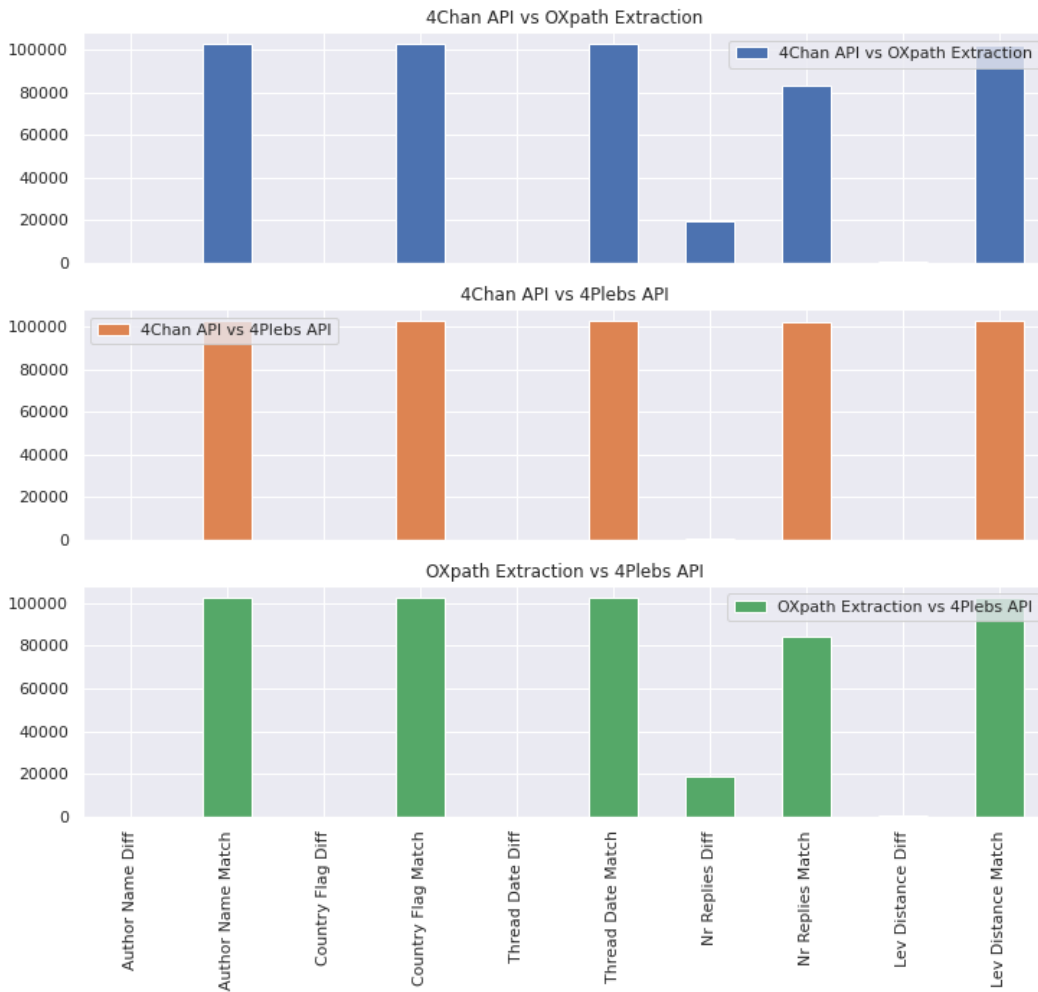


Figure 6.8: Three-way comparison results projection

Another phenomenon likely to impact the difference in the number of replies is the work of janitors that prune posts and threads from the site [93]. Due to the ephemeral characteristics of 4chan.org, it isn't easy to get a sense of moderation of threads or posts on 4chan. This is because 1. most of the data available shows the final state of the thread (i.e. once it is archived) and 2. when hitting a missing thread, it's never clear if it is missing due to moderation or data being purged after expiry.

We have noticed that deleted threads and posts can also return to the live board at their last status after a certain amount of time. These could be the effect of janitors sending threads and posts for review and these not meeting the criteria for being removed.

There is one question we can look to answer with the data available: What threads and posts are being removed from the janitors?

There are no indications on the 4chan APIs if a post or thread has been pruned. The API thread documentation ¹⁹ and the API catalogue document ²⁰ have no information about pruned threads or posts.

However, indirectly we can deduct that threads extracted using OSNNs-ScrA and otherwise cannot be found on 4Pleb, or posts that cannot be found on 4Pleb or data extracted from the 4Chan API, are very likely threads and posts that have been pruned.

We looked at the final state ²¹ of all threads and posts found by direct extraction using OSNNs-ScrA and OXPath engine and compared these data with what was extracted from 4Pleb API to see for significant discrepancies.

We compared 4,534,554 posts collected using OSNNs-ScrA and OXPath engine with the 5,381,417 posts found in the 4Pleb API and found 26,457 posts are missing from the 4Pleb API, amounting to 0.5% of the total posts found over the period of 6 months in question. This is the first indication of the weight of the moderation on 4chan. The archiving service 4Pleb and its API offer an insight into the deleted threads and posts by marking them as *"deleted: 1"* in the resulting dataset. These are different from the missing posts found by our analysis since 4Pleb has no visibility of these posts that existed at some point in 4Chan.

Using topic modelling described in algorithm 2, we analysed the topics and the text sentiment in the deleted posts. In Figure 6.9, we show the topic distribution (i.e. inter-topic distance map of the latent dirichlet allocation model [98]), and in Figure 6.10 the sentiment distribution along the negative and positive axis. The sentiment distribution has negative mean: **-0.458**, positive mean: **0.398**, negative median: **-0.477**, positive median: **0.402**, negative standard deviation: **0.188** and positive standard deviation: **0.167** ²².

Further, we build a system for live monitoring the heartbeat (i.e. topics and sentiment and

¹⁹<https://github.com/4chan/4chan-API/blob/master/pages/Threads.md>

²⁰<https://a.4cdn.org/po/catalog.json>

²¹By final state in this context, we mean the state of the thread and all its posts at the moment of the last extraction. Whilst this is likely to be closer to the archived state and higher match with the other APIs, intermediate states might still contain additional posts that have later been deleted.

²²We are filtering out neutral data points close to 0 ± 0.001

Algorithm 2 Using Natural Language Toolkit [96] and gensim [97] for topic modelling
4chan

Require: $doc_{corp} = deleted_{post_text}$

Ensure: $num_topics = 20$ ▷ Analyse top 20 topics

$sentences \leftarrow sent_tokenize(doc_{corp})$

$tokens_sentences \leftarrow word_tokenize(sentences)$

$POS_tokens \leftarrow pos_tag(tokens_sentences)$ ▷

Part Of Speech tokens for each token in token sentences

$tokens \leftarrow remove_stopwords(lemmatize(POS_tokens))$

while $token \in tokens$ **do** ▷ Use gensim for large corpora
to create bigrams and trigrams and map the document into the bag of words corpus

$bigram_model \leftarrow Phrases(tokens)$

$trigram_model \leftarrow Phrases(bigram_model[tokens])$

$tokens = list(trigram_model[bigram_model[tokens]])$

$dictionary_LDA \leftarrow gensim_corpora_Dictionary(tokens)$ ▷

Prepare dictionary for LDA [98] modelling using gensim

$corpus \leftarrow dictionary_LDA.doc2bow(tokens)$ ▷ Map tokens to bag of words

end while

$lda_model \leftarrow gensim_models_LdaModel(corpus, num_topics, dictionary_LDA)$ ▷

Create gensim LDA model

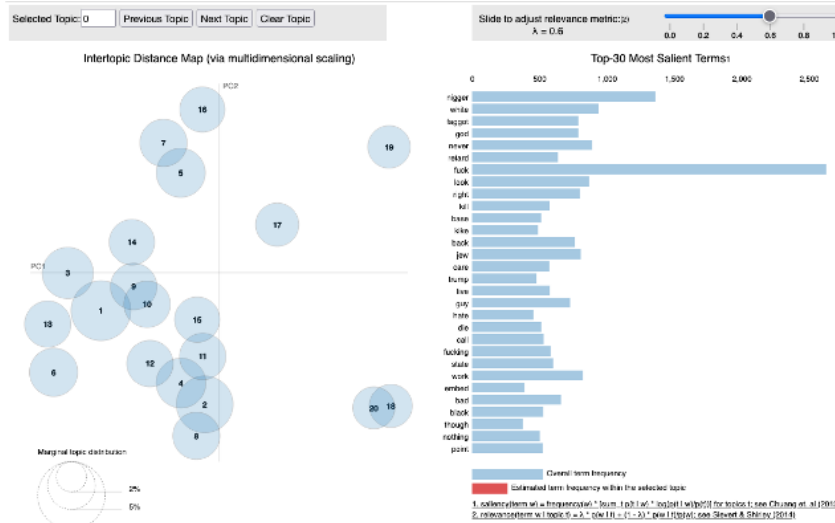


Figure 6.9: Deleted Posts Topics

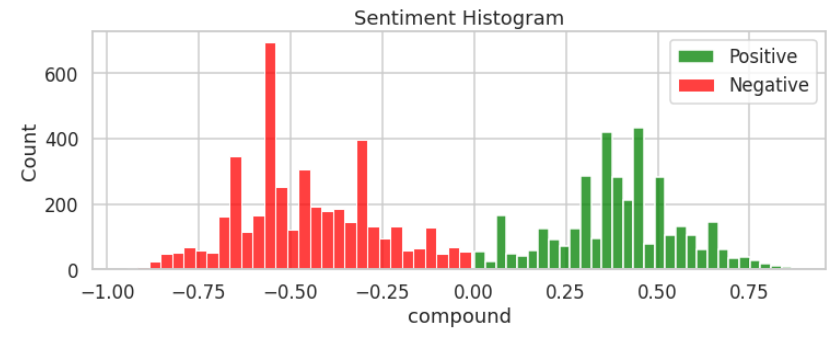


Figure 6.10: Deleted posts sentiment

how they change over time) of the 4chan live board and the transition to the current board from t_{-x} where x is the number of times the entire live board had its threads archived. For example, the live board, t_0 , is composed of the 200 threads that are currently live, and t_{-1} is the board composed of the last 200 threads that were archived, as illustrated in Figure 6.11.

Figures B.3 B.4 show how the topic distribution transitions from one set affinity to another as the live board is generated. The sentiment distribution shows a higher affinity towards extreme negative sentiments. In comparison, somehow expectantly, due to the distribution over time, the inter-topics distance map of the deleted posts is different from the live boards, with topics being more widespread. Unexpectedly, though, the overall sentiment of deleted posts, though with a higher tendency towards the negative sentiment,

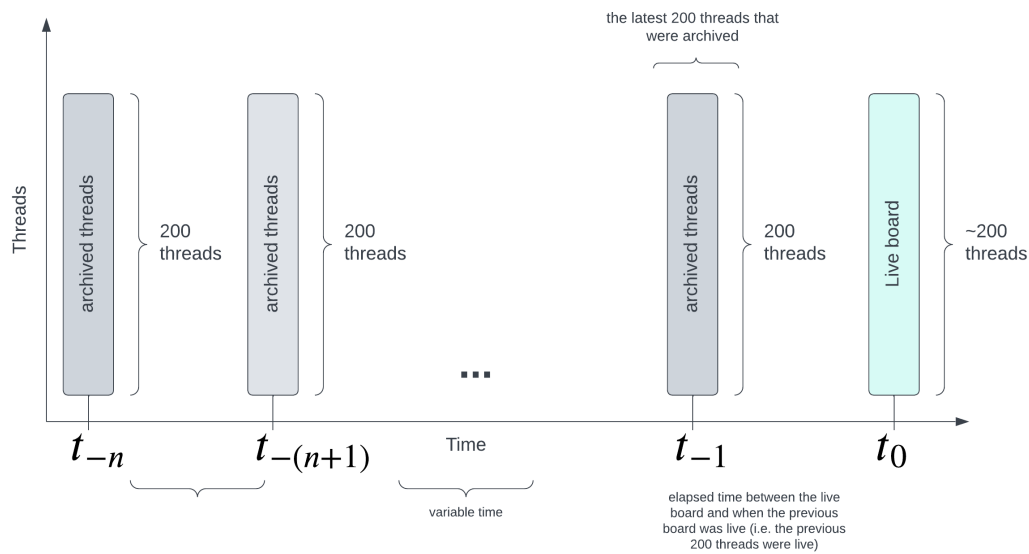


Figure 6.11: The observed distribution of 4chan Live Board and archived boards transitions

is less balanced towards the extreme negatives than the sentiment on the live boards. The extremity of the language used is unlikely to determine the deletion and removal of posts. At this point, we wanted to understand whether the transition from one board to another translated into any significant shifts in the sentiment distribution. We run a Kolmogorov–Smirnov test between the sentiment distribution of each board compared to the sentiment distribution of all the other boards as illustrated in Figure 6.12. For example, the scattered values under Lot_0 are the compound values of the result of the KS test between sentiment distribution of t_0 and t_{-1} , t_{-2} and so forth.

This graph is an indication of whether we have sentiment affinity or if we have a substantial change in the sentiment distribution, so much so that this is now a different distribution. We can see from image 6.12 that some lots (for example, Lot 0, 6 and 9) are central with the most affinity to the sentiment of the topics discussed in the other lots, whilst some others (for example, lots 1 and 2 and 7) offers the most significant transition with very few lot distribution affinity and mostly very different from the others.

6.2.3 Case study limits and conclusions

4chan makes, of its ephemeral characteristics, its core feature for its audience. Anonymity and ephemerality are the fundamentals over which each thread, post, and exchange happens on 4chan. There are no ways to guarantee that any data retrieval algorithm can

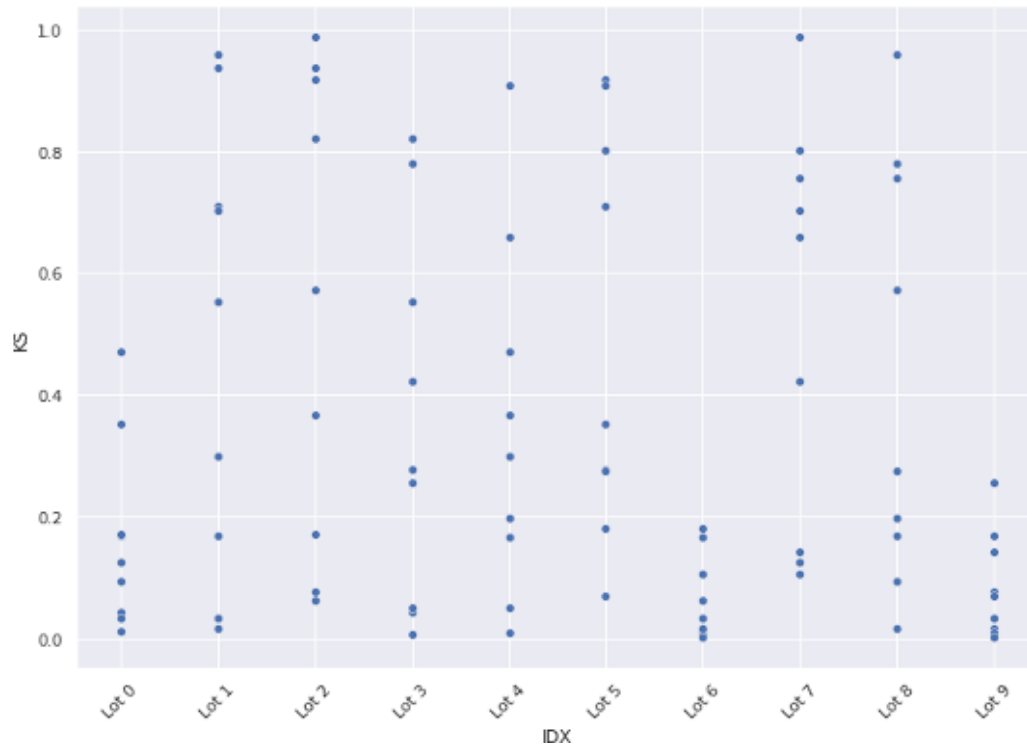


Figure 6.12: Kolmogorov-Smirnov sentiment analysis distribution comparison

capture 100% of the data. In our three-way comparison, we could undoubtedly find threads using the OSNNs-ScrA system and the OXPath query engine, which we couldn't find on the data retrieved from the 4chan API or the 4plebs API. Whilst no conclusion could be made for the data missing from the 4chan API since these could be due to ephemerality, we could hypothesise that these data were never made available via 4chan API. Hence 4plebs archiving service has missing threads.

Whilst the three-way comparison is not symmetric²³, because of the ephemerality of threads and posts, we can derive some conclusions when comparing the data we have captured from all three sources. The comparison, $\approx 100K$ threads extracted from all three sources over a total of $\approx 700K$ threads discovered in the 6-month period, offers a statistically relevant sample for comparison. The results provide a novel perspective (i.e. to the best of our knowledge, this is the first of such analysis in literature) with some important conclusions:

²³Any thread found via the 4chan API scraping and not found in the other sources is a clear indication of missing data capture, the same is not true when looking at the opposite case. Missing data from the scraping of the 4chan API offers no indication if the data was never there or was deleted due to its ephemerality and hence not scraped in time.

1. The 4chan archiving service, 4pleb.org, offers a holistic and reliable view of the threads and posts of 4chan. This defies the ephemerality characteristic altogether since threads, posts, and images are stored forever by 4pleb.org.
2. We saw clear signs of deleted posts at a rate of 0.5% of the total posts, we couldn't see any affinity in the topics of the posts deleted and the sentiment of the deleted posts is not as extreme as the topics of the threads in general posted in the live board.
3. Topics of threads on the live board tend to be clustered by inter-topic distance and a board transition (once all live threads are archived and new threads are live), which translates into a new clustering of inter-topic affinity.
4. The sentiment of the live threads tends to be biased toward the extreme negatives. This remains valid as the board transition into new threads.
5. Whilst the bias towards the negatives characterises the live board, the sentiment distribution changes and Kolmogorov-Smirnov tests indicate the tendency for a major transition in sentiment distribution.

6.3 Data collection campaigns and datasets

Regarding Table 3.1 and 3.2, we ran several data collection campaigns and produced an even more significant number of datasets made available to the research community. This section explores the collection campaigns, each dataset built and their data structures.

6.3.1 High trust datasets

We run four data collection campaigns in the high trust demand online social networks of needs. Specifically, the four data collection campaigns were focused on four different high-trust areas:

1. Childcare: *childcare.co.uk*
2. Elderly and nursing care: *carehome.co.uk*
3. Health and medical care: *doctify.com/uk*, and
4. Pet care: *rover.com/uk*

Whilst the campaigns are generally very similar to the case study in terms of query creation and data collection execution; there are differences due to the way these platforms work. The discovery queries start by performing a search that has its main input, the geographical location in the UK. The discovery queries are dynamically generated via API integration. UK location information is dynamically baked into the queries. In most cases, geolocation information is passed as UK postcodes, by city names, longitude, and latitude can also be input parameters ²⁴. Additionally, the search results have useful aggregated information we capture during discovery. For example, for the pet care source, the API integration is simple, and the configuration for the integration is a single JSON line of code: { "endpoint": "https://extractor-api.azurewebsites.net/rover-discovery" ²⁵ }. The endpoint, in turn, generates the XPath shown in the code snippet A.7. The resulting XPath will perform a web search and capture all deep links from the resulting page, together with a payload of aggregated information shown in the search result. The search result is shown in Figure B.5, and the structure of collected data as a result of executing the discovery queries is shown in Figure 6.13

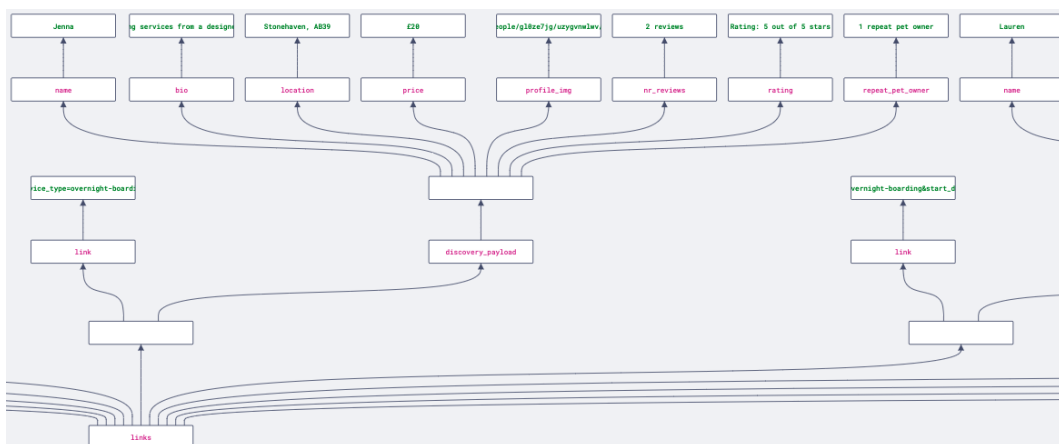


Figure 6.13: Rover.com discovery collection data structure visualisation

We ran the High Trust data collection campaigns for one month, performing multiple scans for each platform. The number of service providers, reviews, and size of the data for the last collection for each campaign is shown in Table 6.4.

²⁴For example, the discovery queries for doctify.com take city name, longitude and latitude as input parameters

²⁵This is a pay-as-you-go azure cloud web service, and resources might run out. Alternatively, commodity servers were used as a backup and reachable under <http://srv-us.prifti.us/rover-discovery>

Dataset	Service Providers	Reviews	Storage Size
Childcare	85 883	23 468	305MB
Carehome	17 488	79 677	550MB
Doctify	8 985	132 247	685MB
Rover	2 139	53 691	315MB

Table 6.4: High Trust dataset characteristics

6.3.2 Medium trust datasets

We run four data collection campaigns in the medium trust demand online social networks of needs. Specifically, the four data collection campaigns were focused on two different Categories of trust demand, **Property Care** and **Property Sharing**:

1. Property Care: *checkatrade.com, trustatrader.com*
2. Property sharing: *getaround.com, airbnb.com*

Whilst we collected information about the property networks, we failed to complete the task for the networks in the property sharing category. This was due to technical limitations where the query engines need further work to support more modern technologies.

6.4 Conclusions

In this section, we describe in detail our data campaigns. We started with a challenging case study to prove the universal applicability and some advantages of the system designed and implemented in the previous chapter. Thanks to the granularity of the data extracted from 4chan, we could prove the following important hypothesises:

- we could verify most of the behaviours expected and discussed by previous research on 4chan.org. This is an important confirmation of the 4chan.org API since the data source for our study is different.
- 4chan ephemerality if not sustained and web archiving services retain a full cover of the activity on 4chan.org/pol
- we can track moderation in 4chan, generally not visible via the 4chan.org API. We studied the topics of the deleted threads and projected the sentiment over six

months. We compared the sentiment of the deleted threads to the rolling live board and found that the sentiment trend of the deleted threads is different from the sentiment distribution in the live board.

- thanks to the live extractor, we can monitor the topics and sentiment of the live board and see how it transitions from one state to another to indicate a change in dominating topic or sentiment.

Chapter 7

Trust evaluation and trust models emerging from data

Following the trust literature discussion in Section 2.2, in the previous chapters, we discussed, implemented and applied a new system for trust information collection. In the literature review, Figure 2.3 highlights the importance of network theory in social network analysis and trust models. Out of 18 trust models reviewed, over 65% use graph-based evaluations. Network Centrality measures are at the core of social network analysis and often the basis for trust metrics [99]. Google's PageRank, Trust Centrality [99], and EigenTrust are all trust metrics derived from network centrality measures.

We now have social trust data from multiple sources and can follow the theoretical concepts discussed in Chapter 3. More specifically, in this chapter, we are going to:

- use data from childcare.co.uk to build graphs and run social network analysis discussed in Section 3.3.
- use data from high trust datasets, discussed in Section 6.3.1, and create multiplex networks discussed in Section 3.3.1
- calculate the value of the trust (Equation 3.3) for agents that have received reviews, use factorisation machines to predict trust for agents without reviews and evaluate its predictive power over time
- discuss the trust relations between different online social networks of needs, the total trust and its variations over time

The first part of this chapter is focused on analysing existing trust models, measures and values emerging from trust expressions in the form of reviews. As discussed in Chapter 3, it is common in the literature to analyse reviews and derive trust metrics. There are well-known limits to this approach since reviews and recommendations are known to suffer from positive bias of opinions [100]. In Section 7.3, we further discuss the low number of reviews and the negative correlation between trust demand and the number of reviews. We have defined and tested a novel approach to trust that is attribute-based, does not rely on reviews and is more comprehensive in the network evaluation. In the later sections of this chapter, we compare the results between our attribute-based trust models and models derived by analysing the network characteristics of the reviews.

7.1 Network analysis for the online social network of needs

When landing on the homepage of childcare.co.uk, you get presented with the following sentence:

*Childcare.co.uk is an award-winning online social networking platform for parents, childcare providers and private tutors with over 2.5 million members. Start your childcare search or childcare job search today.*¹

Apart from efficiently describing the utility of the social network, this sentence also highlights the nodes and relations of the graph we built. In fact, we have: • **providers**, • **parents** and • **jobs** as nodes and • **awards job** and • **receives service** as links. The jobs are represented by the reviews received by the service providers, and they link parents to providers.² In fact, this is a common pattern among all OSNNs. Figure 7.1 shows a visual representation of nodes and links in childcare.co.uk.³

We build the graph using a distributed graph database, specifically Neo4j [101]. The result

¹As of 25th of November 2022

²We will be using *review* instead of *job* and [*leaves review*, *receives review*] instead of [*awards job*, *receives service*] going forward.

³The Jupyter notebook for the creation of the graph can be found here: *Childcare Network Creation Notebook*. The Jupyter notebook includes read-only access to source data but requires a new Neo4j server. Neo4j offers a free tier for its cloud-native Distributed Graph Database that can be used to replicate the research in this chapter.

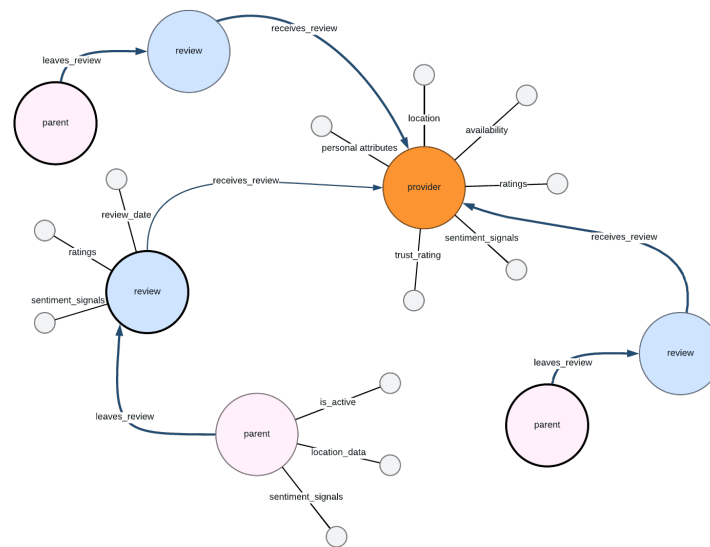


Figure 7.1: Childcare.co.uk network

is a disconnected directed bipartite multi-graph made of 80,325 provider nodes, 22,885 nodes representing parents, 24,087 nodes representing reviews, 2,512 nodes representing UK locations, 9 region nodes, 3 country nodes, and a total of 196,220 links. The graph is directed because there is clear direction from parents leaving reviews to providers receiving reviews. The graph is bipartite because you can separate parents and provider nodes from review nodes. Furthermore, there are no cycles since only parents can initiate and leave a review, and providers can only receive reviews once they have delivered their services.

```
1 match (p:parent) - [lr:leaves_review] - (r:review
   ) - [rr:receives_review] - (pr:provider) return p, lr, r, rr, pr
```

Most of the providers haven't received any reviews. There are 80,325 active providers in the network, but only 7,143 (i.e. 8.9%) have received a review. There are 24,087 reviews in the network.

The chart in Figure 7.2 shows the frequency and cumulative distribution of the number of reviews received by each provider. About 95% of the providers that have received reviews have received less than 10 of them. Relations in high trust-demanding networks are harder to form and supposedly long-lasting. The graph density is only 3.29×10^{-5} and average degree of 1.78. ⁴

⁴The graph density is slightly higher, 4.13×10^{-5} , when you include locations, regions and respective

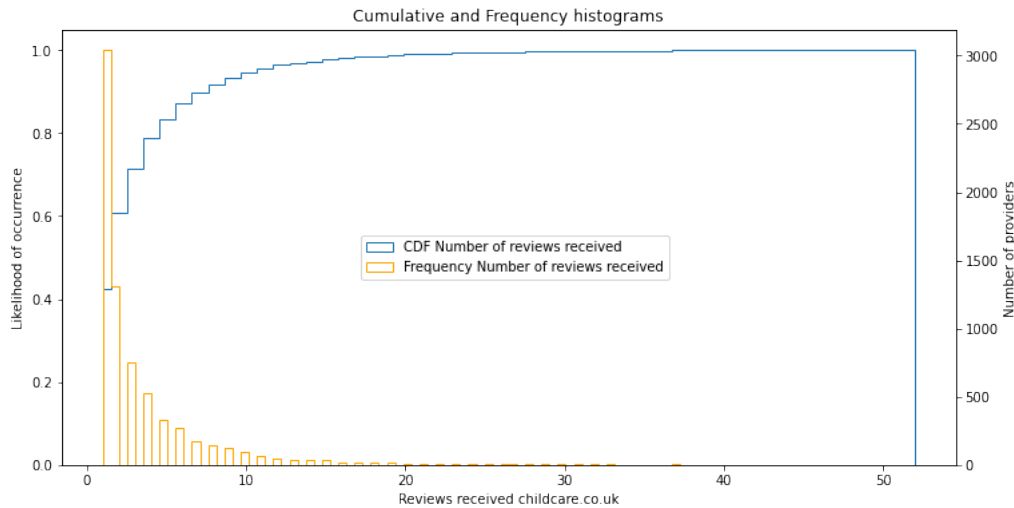


Figure 7.2: Childcare provider received reviews distribution

A visual representation (Figure 7.3) of a random walk of 3000 provider nodes and their review and parent links shows a highly clustered and disconnected graph. Unsurprisingly, the graph is provider-centric and shows characteristics of an egocentric network. Providers create loyalties for long-term relations, and parents will only work with a few trusted providers. The code sniped in 7.1 shows the Neo4j cypher query used to generate the graph in Figure 7.3.

We can expand the network analysis to include the locations where the services are provided and where job exchanges (i.e., reviews) have happened. Locations are represented by the outcode of a UK postcode⁵, their links into region nodes, districts, and countries. Whilst the network of parents and reviews is disconnected (i.e. not enough parents leave reviews to distinct providers to form a connected network), the location-centric graph is connected since most providers offer services in multiple locations. In fact, the results of community detection [102], on the two different views of the graph, show the different natures of the two. There are 6403 communities detected in a provider-centric network but only 456 in a network that additionally features links to locations.

The betweenness centrality on the connected network that includes the locations, is greatly dominated by the geolocation structure and association of postcode locations and regions, featured in the new graph. The eigenvector centrality, instead, weights the

links

⁵https://en.wikipedia.org/wiki/Postcodes_in_the_United_Kingdom

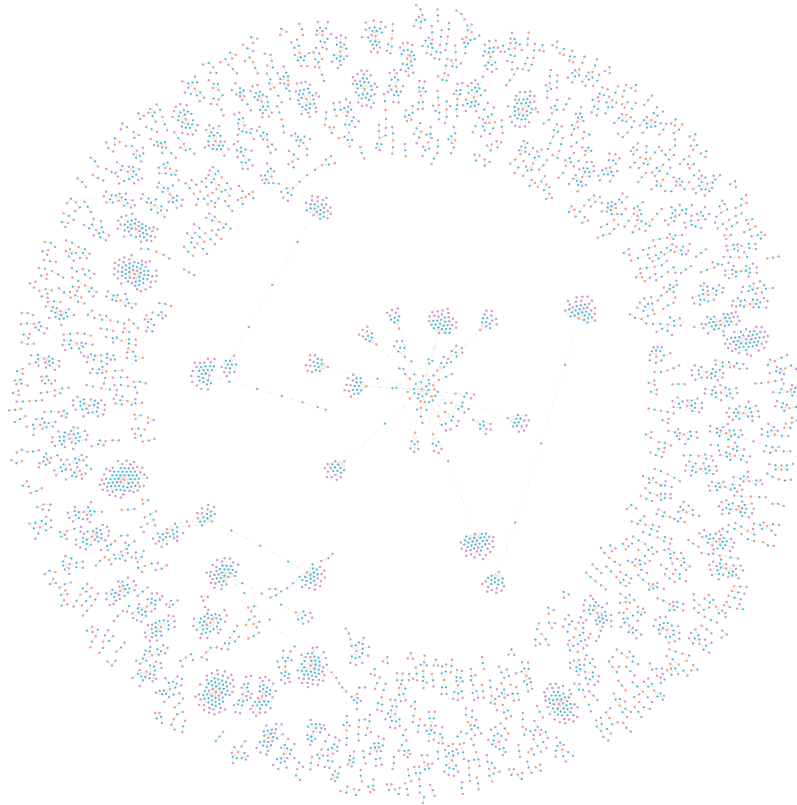


Figure 7.3: Random graph view of 3000 parents, reviews, and provider nodes on childcare.co.uk

in-links and weight of the connected nodes. The providers with the highest page rank are the ones that are connected to reviews from parents that have worked with other providers and connected to locations where many providers with high page rank offer their services.⁶ The two figures in 7.5 show the differences between the two centrality measures. Betweenness centrality is, in general, dominated by locations, however, there is a direct correlation between providers with the highest betweenness centrality and providers with the largest numbers of reviews received.

The same is not true for the providers with the highest page rank. In Table 6.2, we show the top 10 providers with the highest betweenness and page rank scores. Interestingly, none of the providers with the highest betweenness centrality features in the list of the providers with the highest page rank centrality. The top 3 providers with the highest closeness centrality feature in the top 10 providers with the highest page-rank centrality. This pattern didn't change when we ran the same calculation on other UK regions or

⁶Page rank and eigenvector centrality are being used interchangeably

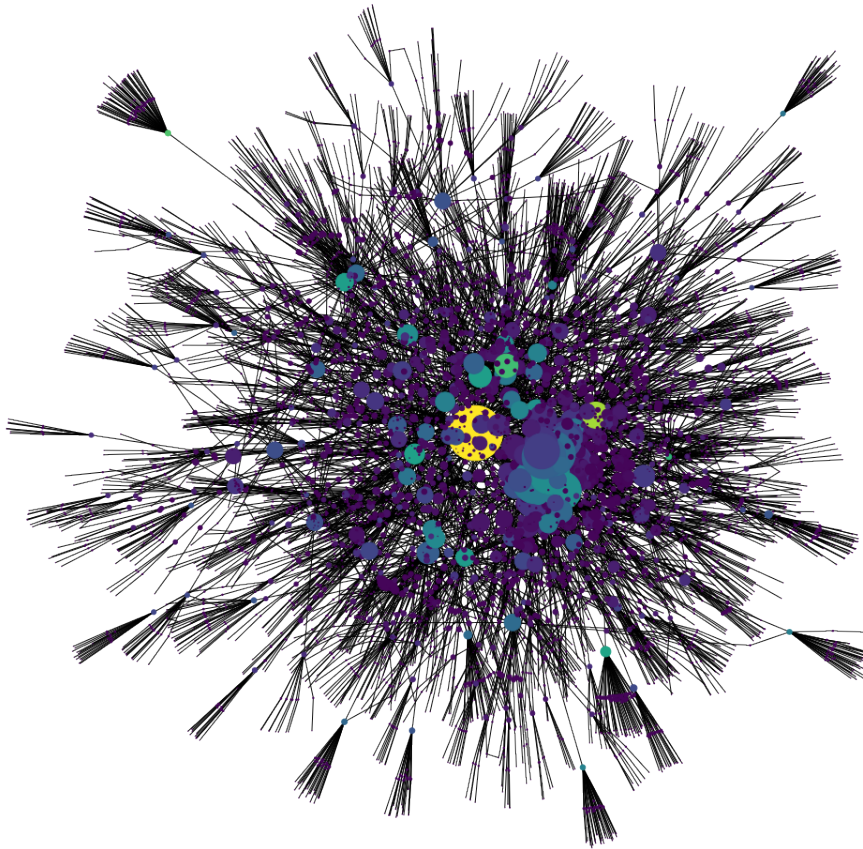


Figure 7.4: Eigenvector Centrality

when running on the full network rather than using sampling. The betweenness centrality is more closely associated with a higher number of reviews, whilst the page-rank centrality seems not to feature any of the providers with the highest number of reviews. The closeness centrality values are at an order of 10 higher than the other two centrality measures, as shown in Figure 7.5

Another pattern emerging from the regional networks is that the eccentricity measure is always 8. This number is a network representation artefact, which is the total number of links between the two most distant nodes (by definition).

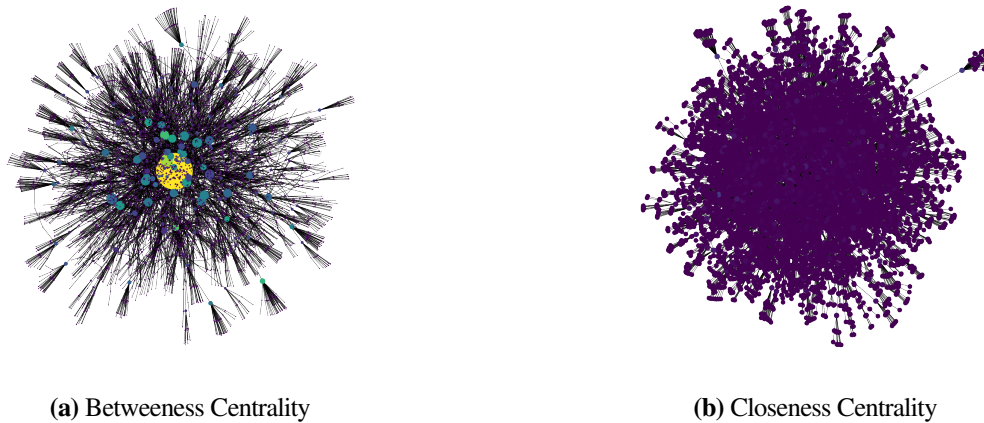


Figure 7.5: Centrality measures for a sample of 5K nodes from the childcare network in London

In fact, 8 is the number of links in the following path: *parent* \rightarrow_1 *review* \rightarrow_2 *provider* \rightarrow_3 *location* \rightarrow_4 *region* \rightarrow_5 *location* \rightarrow_6 *provider* \rightarrow_7 *review* \rightarrow_8 *parent*. These characteristics can be seen in the Figure 7.1 Section (b). Unlike the other centrality measures, closeness centrality has a much smaller standard deviation from the mean. The normalised standard deviation of the closeness centrality values is 0.00132. By comparison, the standard normal deviation for the betweenness centrality and page rank centrality are larger by a factor of 8. These values are 0.009613 for page rank and 0.00974 for betweenness centrality.

The childcare network is disconnected in its triad of nodes, parents, reviews, and providers. It is, however, connected by locations where services are provided, and this allows us to derive some conclusions from the centrality measures. Yet, most of the providers that haven't received any reviews are left out of these calculations. A trust measure that weights the features of the individual providers would provide bigger insight into the trustworthiness of each provider.

7.2 Multiplex care network

In this section, we analyse holistically the care network of careers, jobs, and providers segmented by locations. In the previous chapter, we described the data retrieval from four

Betweenness Centrality				Eigenvector Centrality				Closeness Centrality			
Prov	Cent	R	L	Prov	Cent	R	L	Prov	Cent	R	L
73286	.0.016	41	1	2449523	0.011	3	4	1093188	0.194	14	2
109854	0.016	44	1	848517	0.009	10	3	1252990	0.194	2	2
70763	0.014	38	1	340251	0.009	2	3	1095304	0.194	1	2
1073016	0.011	28	1	2313178	0.009	2	4	970779	0.194	1	2
14005	0.010	25	1	2255913	0.008	1	3	837197	0.193	18	2
212922	0.009	22	1	1402304	0.008	4	3	109854	0.193	44	1
339021	0.008	21	1	214141	0.008	8	3	1904057	0.193	11	2
105020	0.007	18	1	1983931	0.008	5	3	130340	0.193	9	2
300229	0.007	18	1	1093188	0.006	14	2	468559	0.193	9	1
123798	0.007	16	2	1252990	0.006	2	2	2623136	0.193	9	2

Table 7.1: Top 10 providers by centrality measures. R: number of reviews received and L: number of locations where they provide their services.

different high-trust demanding networks. These were: *childcare.co.uk*, *carehome.co.uk*, *doctify.com* and *rover.com*^{7 8}

The chart in Figure 7.6 shows the composition of the multiplex network in terms of its nodes and links. The network can be represented by:

$$G = (V, E_{type})$$

Where: $V = \{ \textit{parent}, \textit{carehome_carer}, \textit{petowner}, \textit{provider}, \textit{carehome_provider}, \textit{doctify_provider}, \textit{review}, \textit{carehome_review}, \textit{doctify_review}, \textit{rover_review}, \textit{location}, \textit{district}, \textit{region}, \textit{country} \}$ and

$E_{type} = \{ \textit{parent}_{\textit{leaves_review}} \textit{review}, \textit{review}_{\textit{receives_review}} \textit{provider}, \textit{petowner}_{\textit{leaves_review}} \textit{rover_review}, \textit{rover_review}_{\textit{receives_review}} \textit{rover_provider}, \textit{carehome_provider}_{\textit{provides_in}} \textit{location}, \dots \}$

Each subgraph of the multiplex network represents the care network defined by its source. In each dimension, the care network nodes are specific to that level. For example, a petcare provider node is called *rover_provider* whilst a healthcare provider

⁷Some of these networks offer their services in different countries. Our data retrieval campaigns were focussed on the network in UK.

⁸Whilst three of the four networks in question focus on the individuals providing services, *carehome.co.uk* takes a slightly different approach, and most of its providers are Care Homes. However, the trust relations, reviews, and feedback are focused on the individuals providing the service rather than the Care Home entity.

is called *doctify_provider*. Each layer denotes a different type of care, and generally, there wouldn't be any shared providers because a scenario of a provider offering pet care services on rover.com and offering healthcare services on doctify.com are highly uncommon in real life.

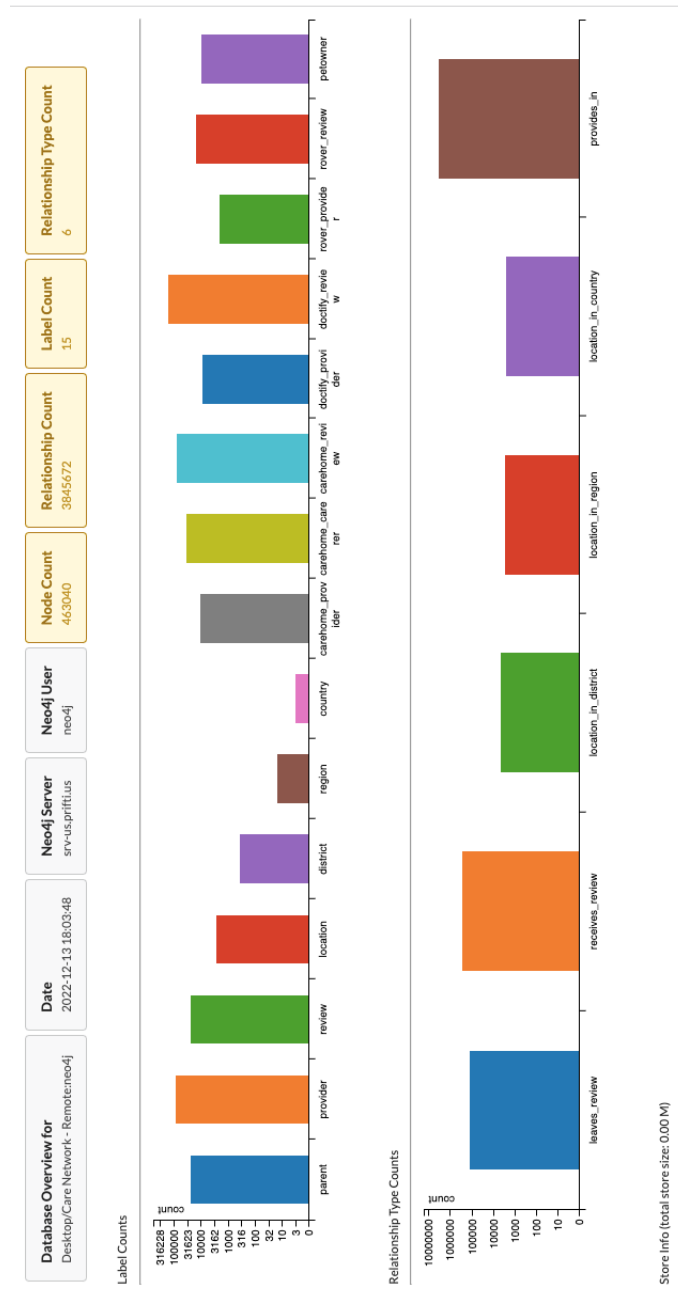
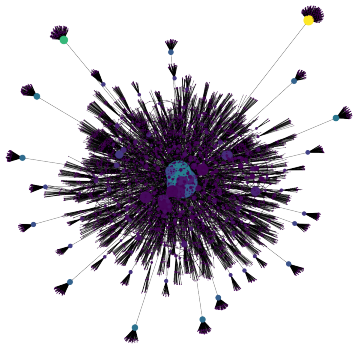


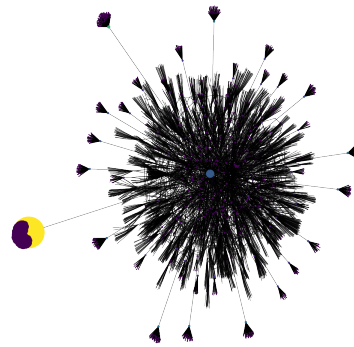
Figure 7.6: A detailed view of the care network

It is not uncommon, however, that providers offer their services on multiple

platforms. Whilst there is an opportunity to compute trust based on aggregated data from multiple platforms, entity resolution presents another challenge since there are no direct links between providers on multiple platforms. We analysed two networks from the “Home Care” group of Online Social Network of Needs. More specifically, we looked into the providers of trustatrader.com and checkatrader.com. Most providers are sole traders that trade with their trade name. The trader name is unique across all platforms, making it slightly easier and more predictable to match them throughout the different platforms.



(a) Betweenness Centrality for the Multiplex care network



(b) Eigenvector Centrality for the Multiplex care network

Figure 7.7: Centrality measures of multiplex care network of 5K nodes in top 10 UK locations with the highest households

By comparison, most providers on childcare.co.uk or rover.com can choose a screen name that can be their name, their full name or anything else. These characteristics make it impossible to match these providers by name on other platforms.

There are 4,731 unique providers (i.e. based on their trade name) on trustatrader.com, and there are 23,759 unique providers on checkatrader.com. There are only 68 traders that can be matched by their name on both platforms. This number can be slightly higher because of name variations (for example: using or not using

Provider Name	CheckATrade URL	TrustATrader URL
t carey's landscaping	TCareysLandscaping	t-careys-landscaping-blockpaving-driveways-st-albans
g j smith roofing	GJSmithRoofing	g-j-smith-roofing-fascias-soffits-guttering-watford
hale electrics ltd	HaleElectricsLtd	hale-electrics-ltd-electrical-inspection-testing-bromley
fab moves	FabManandVan	fab-man-and-van-removals-three-rivers
oakenshield construction ltd	OakenshieldConstruction	oakenshield-construction-ltd-extension-specialists-wisbech
t l contractors limited	TLContractors	t-l-contractors-limited-blockpaving-driveways-flintshire
zion driveways & patios	ZionDrivewaysAndPatios	zion-driveways-patios-blockpaving-driveways-bury
l.t.d maintenance	LTDMaintenance	l-t-d-maintenance-double-glazing-suppliers-sutton
nat handyman services uk	NatHandymanServicesUK	nat-handyman-services-uk-dartford
hammer and halo ltd	HammerAndHalo	hammer-and-halo-ltd-handyman-wandsworth

Table 7.2: Ten matching providers on two different platforms

“Ltd” in their name); however, this is very low for what would be expected from the two biggest platforms for home care traders in the UK. Table 7.2 shows ten matching providers on both platforms and the links to their pages on the different platforms^{9 10}.

Another obvious extra dimension link in the care network is the location that is shared among all providers. We looked at the combined reviews and providers in the top 10 locations¹¹ with the highest number of households in the UK and build the multiplex network initially limiting the view to only 5000 nodes as

⁹Base URL path for checkatrade.com is "https://www.checkatrade.com/trades/"

¹⁰Base URL path for trustatrader.com is "https://www.trustatrader.com/traders/"

¹¹Whilst the query selects nodes from the top 10 locations with most households in the UK, on some of the levels, the limit of 5000 nodes is reached before all 10 locations can contribute to the graph.

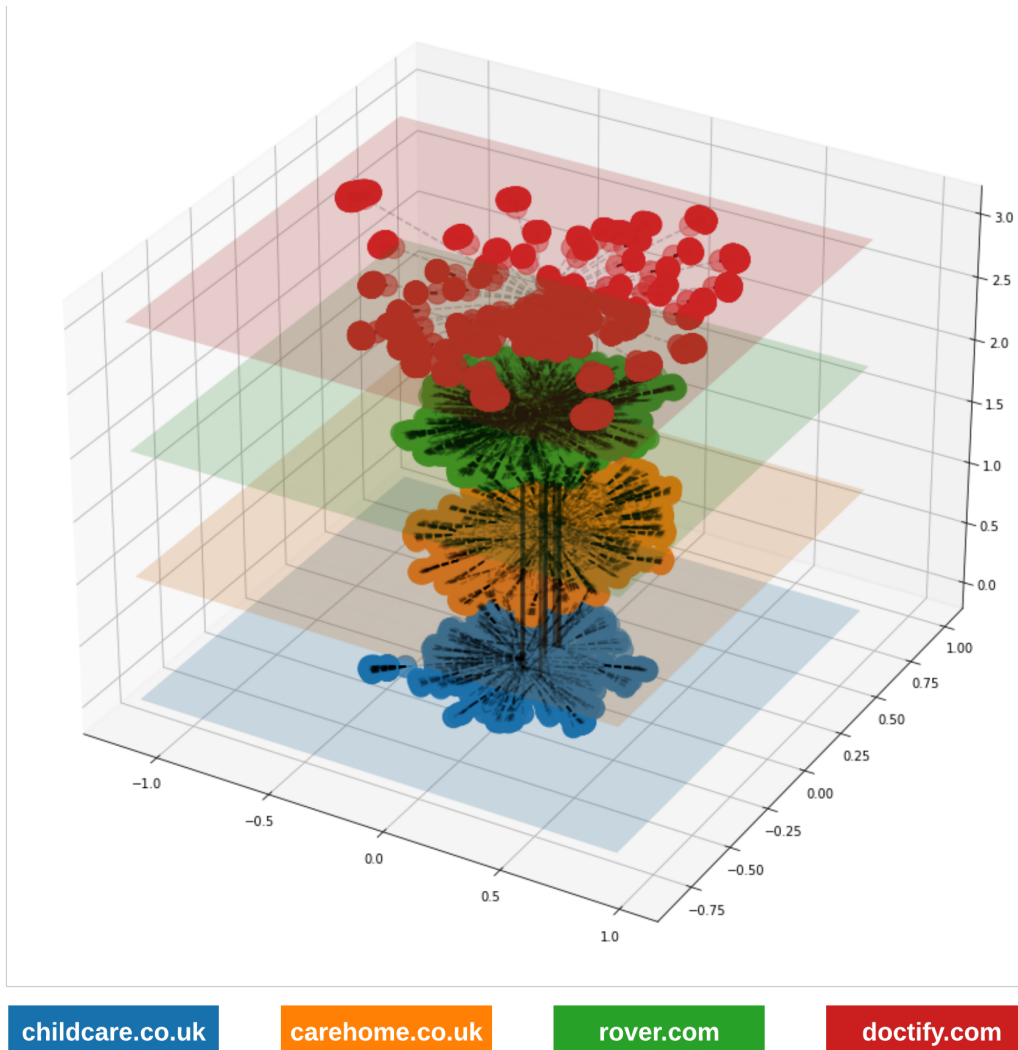


Figure 7.8: Multiplex care network simplified view of 5000 nodes from the top 10 locations with the highest households in the UK

shown in Figure 7.6. The projection shows that pet care and healthcare have a more dense coverage per location than child care and elderly care. Yet, there are more childcare providers in the network than any other providers. This highlights the differences in the radius where the services are offered for each type of care. Healthcare services have the most outreach, and childcare the least.

Table 7.3 shows the extradimensional links between locations. There are 1,065 childcare providers providing services in the top 10 locations with most households in the UK,

2,910 care homes, 4,941 petcarers, and more than 5,104 healthcare service providers. The high number of healthcare providers per location is generally explained by the patterns of the service provided. The healthcare providers are, in general, general practitioners or specialised doctors that associate themselves with multiple medical centres that in turn offer their services in multiple locations. For example, some doctors are associated with clinics in London, Brighton, and Leicester and these clinics offer their services to a vast radius in these locations. This means that when we search by postcode, the same doctors will appear as available in numerous postcodes in all these locations. By contrast, childcare providers only offer their services within a few miles of where they live.

Level out	Level in	Node type	Node detail
childcare.co.uk	carehome.co.uk	location	N1
childcare.co.uk	carehome.co.uk	location	BN2
childcare.co.uk	carehome.co.uk	location	BN3
childcare.co.uk	carehome.co.uk	location	BN1
childcare.co.uk	carehome.co.uk	location	LE2
childcare.co.uk	carehome.co.uk	location	E17
childcare.co.uk	carehome.co.uk	location	NG5
childcare.co.uk	carehome.co.uk	location	CR0
childcare.co.uk	carehome.co.uk	location	LE3
childcare.co.uk	carehome.co.uk	location	LE4
carehome.co.uk	rover.com	location	BN2
carehome.co.uk	rover.com	location	BN3
carehome.co.uk	rover.com	location	BN1
carehome.co.uk	rover.com	location	LE3
carehome.co.uk	rover.com	location	LE2
carehome.co.uk	rover.com	location	E17
carehome.co.uk	rover.com	location	NG5
carehome.co.uk	rover.com	location	CR0
carehome.co.uk	rover.com	location	N1
carehome.co.uk	rover.com	location	LE4
rover.com	doctify.com	location	CR0

Table 7.3: Extradimensional links

Figure 7.6 shows different perspectives of the same network and highlights similarities, differences, and extradimensional links.

We computed the betweenness and page-rank centrality measures on the multiplex network. These show different patterns compared with the centrality measures seen on

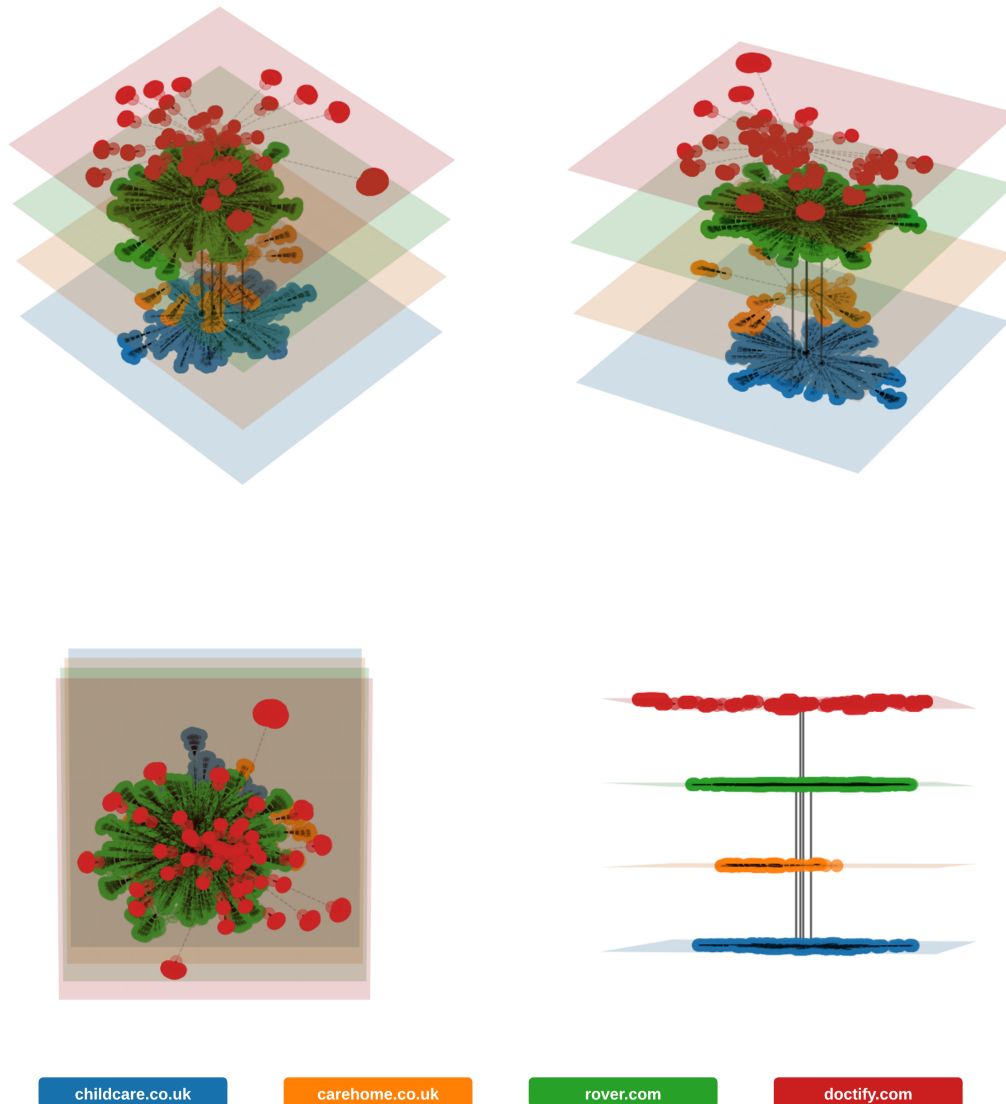


Figure 7.9: Different perspectives of the Multiplex care network of 2000 nodes from the top 3 locations with the highest households in London

chaldicare.co.uk. Figure 7.7 shows the multiplex centrality measures and the various patterns appearing in the multiplex. At the very least, these networks and the more central node patterns look entirely different. Whilst it isn't easy to be conclusive in terms of what the different patterns, shapes, linking behaviours, and density of the multiplex network translate into, something that can be conclusively stated about the analysis so far: we have analysed providers where a link existed between reviews received and Trustee (i.e. a parent or a petowner or other types of someone requesting a service). However, these analyses

have excluded all the provider candidates that might not have received any reviews. We need a model based on potential and can evaluate trust independently of existing links.

7.3 An attribute-based trust model

Most of the trust measures in the literature, and those analysed in the previous section, are based on networks, centrality measures and an evaluation of probability and likelihood based on previous experiences expressed as reviews and recommendations. However, trust is much more than subjective probability [103]. Quoting Castelfranchi:

...trust is about somebody: it mainly consists of beliefs, evaluations, and expectations about the other actor; his capabilities, self-confidence, willingness, persistence, morality (and in general motivations), goals and beliefs, etc. Trust in somebody basically is (or better at least includes and is based on) a rich and complex theory of him and of his mind. Conversely, distrust or mistrust is not simply a pessimistic esteem of probability: it is diffidence, suspect, negative evaluations relative to somebody

Equation 3.3 represents the degrees of trust evaluated from its non-reductionist components: opportunity, ability and willingness (or in the extended form: "opportunity", "ability and competence", "intent and persistence"). In this section, we evaluate trust scores for childcare providers. We intentionally do not use reviews in the calculation because reviews represent an expression of trust 3.2 and are a way to validate the results. The number of reviews received is used as the independent variable to validate the results. We expect a positive correlation between the computed value of Attribute-Based Trust (aka CF-Trust) and the number of reviews received. Further, when using a non-linear solver to calibrate the belief parameters, we use the number of reviews received as the (solver) objective.

Trust is situational and subjective. In fact, in 3.3, we write $DoT_{x,y,\tau}$ to indicate the representation of the degree of trust for the cognitive agent x (or the *Trustor*, or in our case *parents*) toward the cognitive agent y (or the *Trustee*, or in our case *provider*) for the situation τ (in our case the *care service*). We are computing the degree of trust for care providers as (hypothetically) perceived by parents looking for care services. The reviews in the platform are seen as an expression of trust for a situation and are subjective since

both a parent and a provider are involved. Our computation is objective since we aim to generalise the perceived trust for all parents. The parameters $\alpha_{i,x,y}$, $\beta_{i,x,y}$ and $\gamma_{i,x,y}$ are the subjectivity parameters. Each parent would have different beliefs, hence different values of how much, for example, the number of qualifications (i.e. $\beta_{qualifications,x,y}$) contributes in the evaluation of the ability and competence a provider has to offer their services, or when evaluating its willingness and persistence (i.e. $\gamma_{qualifications,x,y}$). The value of "number of qualifications" for provider y is the $T_{qualifications,y}$ in Equation 3.2

Starting from a position of no knowledge, we can only decide whether an attribute contributes to the value of opportunity, ability, and willingness based on common sense. Whilst this decision can be subjective, it doesn't undermine the trust evaluation since we calibrate the parameters as the next step of this evaluation. Table 7.4 shows the attribute contributions to the degrees of trust calculation.

Attribute	Derived Attribute	Opportunity	Ability	Willingness
About Me	Length	-	-	X
	Positive Sentiment	-	X	X
Profile Image	Happy Emotion	-	X	X
	Age	-	-	-
	Gender	-	-	-
	Race	-	-	-
My Qualifications	Length	-	X	X
My Local Schools	Yes/No	X	-	X
Availability Timetable	Yes Count	X	-	X
Last Updated Timetable	Days Ago	X	-	X
My Documents	Length	-	X	X
Rating Count		-	-	-
Rating		-	-	-

Table 7.4: Attribute contributions

Attributes are retrieved from the provider pages as exposed in the childcare platform. Derived attributes are used in the calculations. We have intentionally left out *Rating Count* and *Rating*, since these will be used for model validation. *Positive Sentiment* is derived from analysing the sentiment of the *About Me* text and using the positive score. We used Vader¹², a NLTK [96] wrapper, for sentiment analysis.

We used DeepFace [104] to analyse the profile image and derived several attributes

¹²https://www.nltk.org/_modules/nltk/sentiment/vader.html

including *Happy Emotion*, *Age*, *Gender*, *Race* etc. We are only using *Happy Emotion* for trust evaluation purposes; however, the other attributes can be useful for trust segmentation and biases analysis.

We have intentionally included attributes that positively contribute to trust. The degree of trust is a number between $[0,1]$. The calculation can be expanded to include negative trust in a range between $[-1,1]$. Other attributes, such as *Anger Emotion* can be included with a negative contribution. Missing values from some attributes or out-of-range values can also have a negative contribution rather than 0.

Each attribute is normalised as follows:

$$N_i = \begin{cases} \frac{V_i}{\text{Max}(V_i) \times \text{Max}(i)} & 0 \leq \text{Max}(V_i) \leq 1 \\ \frac{\log(V_i)}{\log(\text{Max}(V_i)) \times \text{Max}(i)} & \text{Max}(V_i) > 1 \end{cases} \quad (7.1)$$

where:

N_i is the normalised value between 0 and 1 for the i -th attribute,

V_i is the value of the i -th attribute,

$\text{Max}(V_i)$ is the highest value for V_i among all providers for the i -th attribute, and

$\text{Max}(i)$ is the number number of attributes. \log is used on values greater than one to limit the impact of out-of-range values.

Equation 7.1 encapsulates attribute value normalisation $V_i \div \text{Max}(V_i)$ and coefficient normalisation $1 \div \text{Max}(i)$, since the coefficients are equal among all attributes.

The python code for the degrees of trust calculation is shown in the code sniped in 7.1

Listing 7.1: Degrees of Trust Calculation

```

1 def opportunity(item) -> float:
2     return (get_local_schools(item, NR_OPP)
3           + get_my_timetable_avail(item, NR_OPP)
4           + get_about_me(item, NR_OPP))
5
6 def ability(item) -> float:

```

```

7     return (get_my_qualifications(item, NR_ABB)
8             + get_about_me(item, NR_ABB)
9             + get_my_docs(item, NR_ABB)
10            + get_image_score(item, NR_ABB))
11
12 def willingness(item) -> float:
13     return (get_my_timetable_avail(item, NR_WILL)
14             + get_about_me(item, NR_WILL)
15             + get_timetable_last_update(item, NR_WILL)
16             + get_my_docs(item, NR_WILL)
17             + get_image_score(item, NR_WILL))
18
19 def cf_trust(provider):
20     return ( ability(provider)
21             * willingness(provider)
22             * opportunity(provider))
23
24 cf_ratings = [cf_trust(provider['l']) for provider in provider_list]

```

The inner functions are the normalisation calculations for each of the trust components. For example, *get_about_me* normalisation function is shown in 7.2 where *get_pos* return the positive sentiment of the *about_me* text, *MAX_ABOUT_ME* is the maximum value the positive sentiment has in all the population, and *contr* is the number of attributes for the current contribution ¹³.

Listing 7.2: Normalisation function on about_me text sentiment

```

1 def get_about_me(item, contr) -> float:
2     return (get_pos(item['about_me']))/(MAX_ABOUT_ME * contr)

```

The left columns of Table 7.5 show the top 10 providers with the highest Castelfranchi Trust (column "CF") and the corresponding aggregated rating and number of reviews.

¹³The Jupyter Notebook with the complete calculations of the Castelfranchi Degrees of Trust model applied to the childcare network can be found here: <http://srv-us.prifti.us:28888/lab/tree/CastelfranchiTrustCalculation.ipynb>

Prov	CF	R	C	Prov	CF	R	C	Prov	Rand	R	C
143427	0.436	5	3	2273546	0.049	5	53	3087320	1	0	0
2049616	0.411	5	7	20688	0.032	5	46	3215773	1	0	0
234392	0.405	0	0	109854	0.064	4	44	3092006	1	0	0
2875684	0.402	5	2	2354656	0.179	5	41	3222468	1	0	0
2152564	0.395	4	19	1141164	0.085	4	41	1472774	1	0	0
189455	0.372	5	18	73286	0.152	4	41	1193106	1	0	0
2739757	0.370	0	0	103071	0.047	4	39	1321398	1	0	0
682981	0.369	0	0	70763	0.077	4	38	3303579	1	0	0
3223827	0.367	5	4	1862858	0.301	4	37	2228202	1	0	0
2302155	0.364	0	0	3146887	0.286	5	37	3359468	1	0	0

Table 7.5: Top 10 providers with the highest CF-Trust and the number of reviews received. Prov: Childcare Provider, CF: Castelfranchi Trust Score, Rand: Random Trust Score, R: Rating, C: Review Count

The middle columns show the top 10 providers with the highest number of reviews, their ratings and the corresponding Castelfranchi Trust. The four columns on the right show the top 10 providers with the highest random trust (i.e. a random number between 0 and 1 assigned to each provider) and their corresponding rating and review count.

The probability distribution functions for the three measures, CF Trust, Number of reviews and Rating, are shown in Figure 7.10.

Along with calculating the CF Trust, we generate a random number (referred to as 'Random Trust'), which is assigned uniformly at random by using *random.uniform*¹⁴ from the NumPy package in Python. It generates random numbers between [0, 1) with a probability density function of $p(x) = \frac{1}{b-a}$. We compare CF Trust to random trust as a performance baseline measure between the calculated trust and randomly generated numbers. As a first result, we can easily see CF Trust performs better than random.

All numbers in Table 7.5 are rounded to 3 decimal points. There are over 80K childcare providers in the network, and the 10 with the highest random trust score all have a score with five 9s after the decimal point; hence, all are rounded to 1. However, it is interesting that they have not received any reviews (i.e., they have not been trusted by parents yet to provide childcare services). The fact that by a random selection, there is a higher probability of selecting childcare providers that parents have not yet trusted is supported

¹⁴Please see: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html>

by the fact that there are 73182 providers that have not yet received any reviews, and only 7143 providers have received reviews. Less than 10% of the childcare providers in the network have received any reviews. The number of reviews received is an important measure, independent of the rating the parent leaves afterwards. In definition 3.1.1, we treat the rating as an adjustment of status after the decision to trust and collaborate.

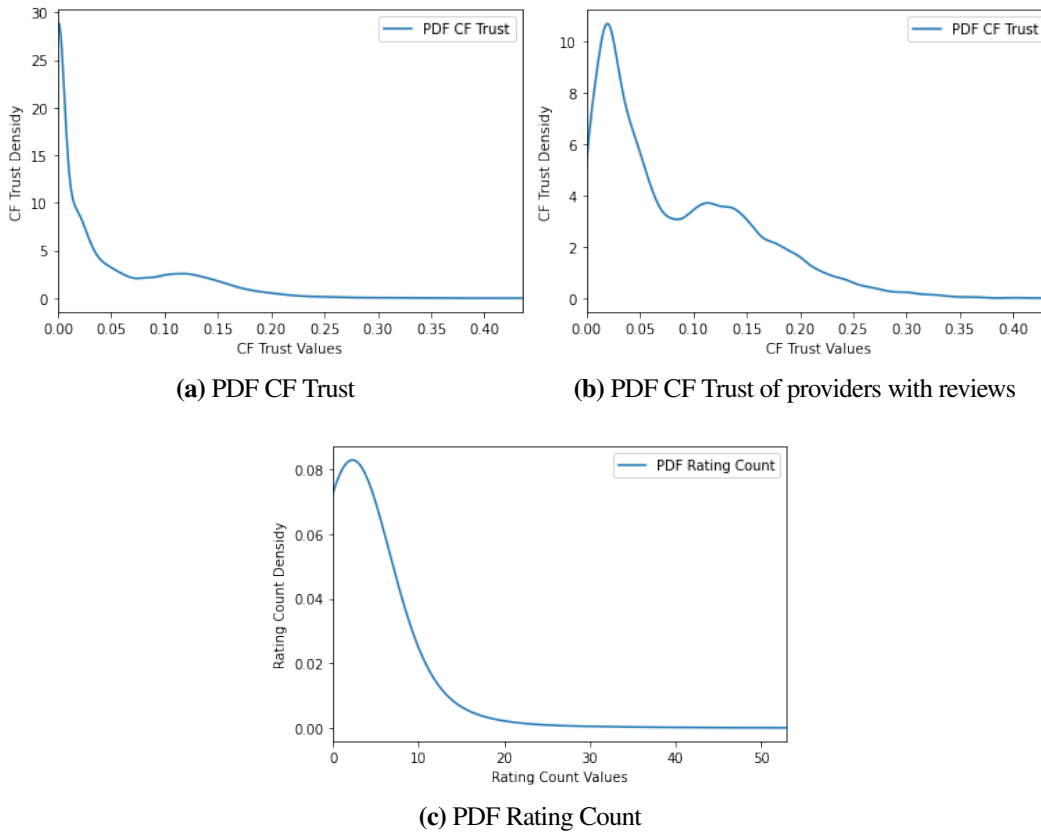


Figure 7.10: Probability distribution functions for CF Trust, Rating and Number of reviews received

The CF-Trust¹⁵ calculation is independent of the number of ratings received or the rating values. These were intentionally not included in the degree of trust calculation. This differs from the centrality measures that mainly depend on the number of reviews received since reviews are the incoming links of the care providers' network. The CF Trust performs substantially better than a random measure (i.e. random trust). For example, the top 100 higher CF-Trust rated provider account for 330 reviews received

¹⁵We use the denotation "CF-Trust" (as in Castelfranchi Trust) to indicate the calculated value of a mathematical representation of Castelfranchi's Degrees of Trust

(i.e. they have been trusted 330 times), whilst the 100 higher rated providers with random trust account only for about 23 reviews. CF-Trust performs as well as Page Rank. The number of reviews received (i.e. the number of times providers have been trusted) for the highest-ranked providers is comparable between CF-Trust and Page Rank. Yet, the CF-Trust measure is independent of the number of ratings, whilst the number of ratings received is embedded and positively influences the page rank calculation.

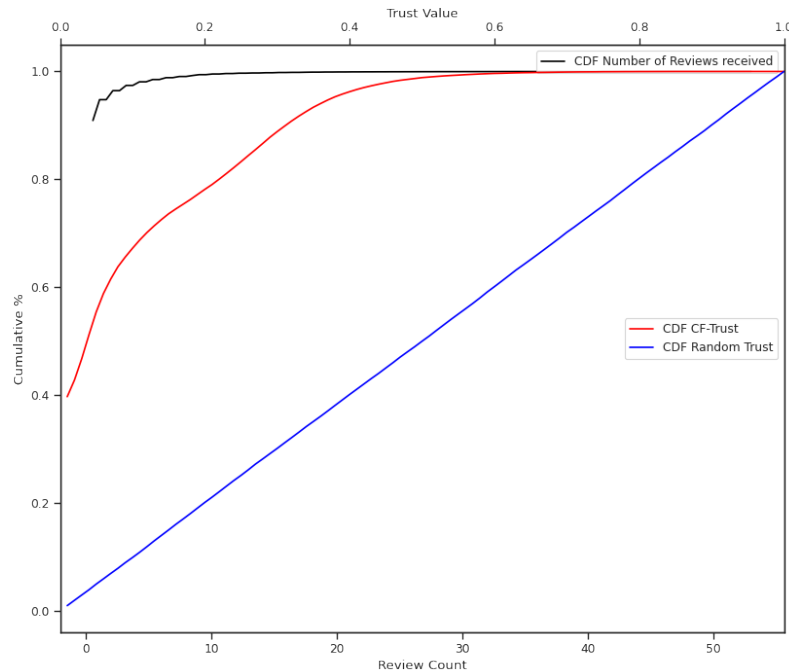


Figure 7.11: CDF comparison of Random Trust, CF-Trust and reviews received

More importantly, the centrality measures do not offer any information about providers that have not received any reviews since the network for calculating the centrality measure focuses on the triad $parent \rightarrow_1 review \rightarrow_2 provider$. On the other hand, CF Trust depends solely on provider attributes, and every provider participates in the trust scoring. There are four providers in the top 10 highest CF-Trust providers that have not received any reviews.

We have trust data represented by the number of reviews each provider has received (the more reviews a provider receives, the more times they have been trusted with care work). We also have computed CF-Trust, attributed-based trust calculations that are independent of the reviews received. We want to validate if attribute-based trust is a good predictor of

reviews received (potentially a good predictor for a trusted provider). KS was introduced as a test for underlying exponential distributions. To validate this correlation, we start by comparing the distribution properties of PDF and CDF. The CDF in Figure 7.11 shows the cumulative distribution of values between CF-Trust and the Number of Reviews received. About 90% of providers have not received any reviews, whilst about 90% of providers have a CF-Trust value of 0.4 or less. About 99% of providers have 30 or fewer reviews and have a CF-Trust value of 0.6 or less.

As the next step, we compute Spearman's correlation coefficient to see if the two distributions are similar and correlated.

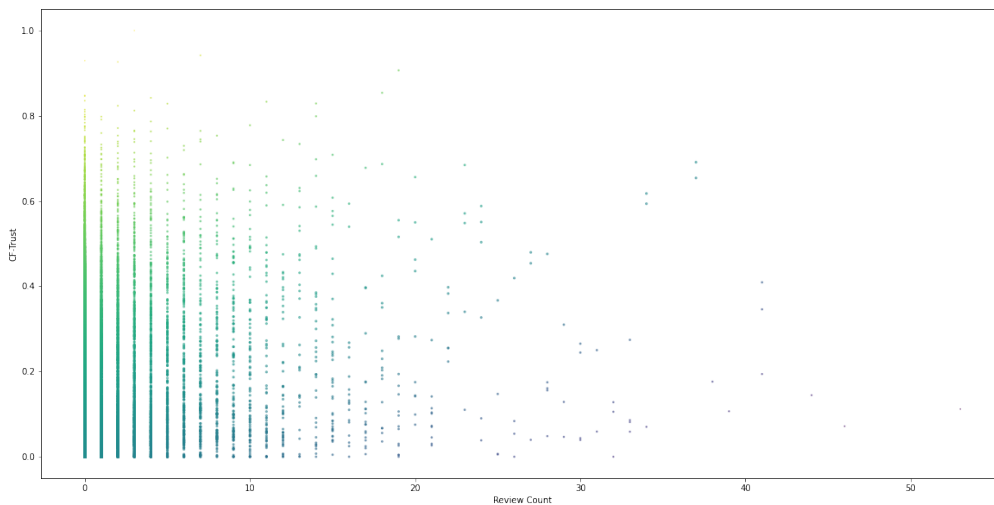


Figure 7.12: Scatter Plot CF-Trust vs Number of reviews received

Spearman's correlation coefficient between the two distributions "*Number of Reviews Received*" and "*CF-Trust*" is 0.238 with a P-value of 0. These results indicate that the two distributions are positively correlated with high confidence (the P-value is very small). In Figure 7.12, results along the $x = y$ diagonal are the highly-correlated results indicating providers that perform similarly with both CF-Trust and the number of reviews received. Points on the lower right-hand side are the providers where CF-Trust calculation indicated low trust yet have received many reviews and are highly trusted in the platform. These providers indicate an opportunity to improve and better calibrate the belief parameters. Results in the scatterplot where the number of reviews received is 0 but the CF-Trust is between $[0, 1)$, by far the majority of providers, indicate results where, even though we have a lack of information when trying to rank them based on number of reviews

received, we can assign trust values based on attribute-based trust computation.

We have an opportunity to calibrate the belief parameters $\alpha_{i,x,y}$, $\beta_{i,x,y}$ and $\gamma_{i,x,y}$ based on observations.

We used ordinary least squared¹⁶ optimisation for linear regression in python [105] to search and optimise the parameters.

We used 10% providers sample from the entire network to run the optimisation search.

We used 50% of providers with ratings from the 10% sample to create our training set.

We normalised the rating count between 0 and 1 and set $\mathbf{Y}_{n \times 1}$ as the optimisation target. For each provider in the training set, we created an array of 13 parameters that participate in the Castelfranchi Trust calculation. These were used as the training matrix of dimension

$\mathbf{X}_{n \times 13}$, where n is the number of sampled providers with ratings.

After each iteration, the belief parameters (i.e. $\alpha_{i,x,y}$, $\beta_{i,x,y}$ and $\gamma_{i,x,y}$) are derived from the OLS model and used in the calculation of CF Trust. Within only ten iterations run over a 10% sample, we can visually see the convergence and get a new set of belief parameters that perform better than the belief parameters result of the normalisation process. Whilst the convergence is embedded in the training model via the target array $\mathbf{Y}_{n \times 1}$; the rating distribution pattern isn't.

Observing Figure 7.13, we notice some interesting facts about the belief calibration:

- In *round 1* the CF Trust looks the closest to the observed expressed trust on the childcare.co.uk platform
- The belief coefficients are not bound; however, the best performing results show coefficients within $[-1,1]$
- R^2 shows the model only explains about 6% of variability. Independently, the adjusted belief parameters yield results that converge closer to observations.
- We started from a position of no knowledge with normalised belief coefficients and now have a set of coefficients that improve the CF Trust match observations.

¹⁶<https://www.statsmodels.org/dev/examples/notebooks/generated/ols.html>

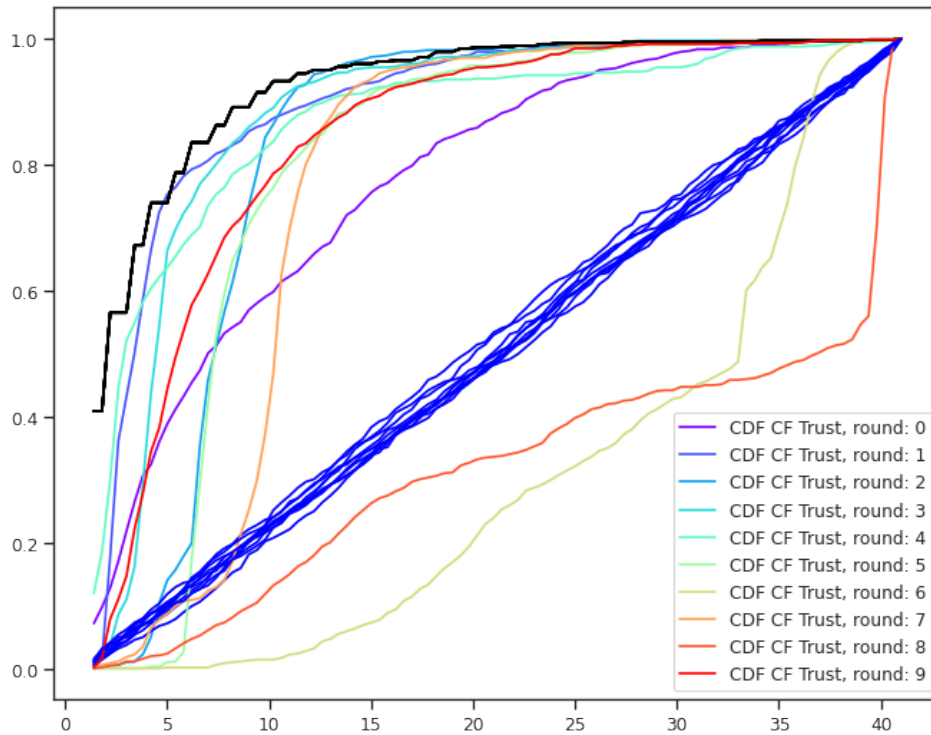


Figure 7.13: CDF Convergence over ten iterations of belief calibration

The linear regression used is bound to have strong limits in optimising the coefficients in a non-linear distribution. For better results for coefficient optimisation, we used non-linear optimisation methods. More specifically, differential evolution¹⁷ and its python implementation in python¹⁸

We ran two different experiments:

- Progressive optimisation with different sample size
- Random walk with the same sample size

The purpose of the first experiment is to evaluate how the parameter optimisation progresses as we run multiple experiments with increased sample size, where each stage uses the output parameters from the previous step. The results are presented in Table 7.6. The results show that the first optimisation, even with a minimal sample size, improves the outcome about ten times. The total drift of the CF Trust, calculated using random belief parameters in input, is ten times bigger than the total drift when using the differential

¹⁷https://en.wikipedia.org/wiki/Differential_evolution

¹⁸https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html

Table 7.6: Rolling Parameter Optimisation with different sample size

Experiment	1	2	3	4	5
Execution Time (Minutes)	10	18	13	36	41
Sample Rate (%)	0.2	0.4	0.6	0.8	1
Sample Size (Number of Records)	14	28	42	57	71
Objective Function Value	0.936	2.893	6.010	6.802	7.748
Total drift with input parameters	45.03	12.62	18.87	24.44	30.109
Total drift with optimised parameters	5.55	12.62	18.87	24.44	30.109

evolution optimised parameters. After that, however, all successive executions that use the parameters optimised in the previous step do not improve the total drift further. This is despite the increased sample size, which results in a significant increase in execution time.

Table 7.7: Random walk parameter optimisation

Experiment	1	2	3	4	5
Execution Time (Minutes)	10	11	11	11	11
Sample Rate (%)	0.2	0.2	0.2	0.2	0.2
Sample Size (Number of Records)	14	14	14	14	14
Objective Function Value	1.144	0.472	0.847	0.534	1.051
Total drift with input parameters	77.94	79.70	53.36	63.07	37.49
Total drift with optimised parameters	6.612	5.845	5.93	6.512	5.831

In the second experiment, we ran the optimisation five times with a sample size rate of 0.2%, starting from a random starting position (i.e. using random belief parameters). We compared the results shown in Table 7.7. In all five experiments, across both methods, even though both the parameters and the sample data were different, the resulting optimised belief parameters converge to the the same values. These results were confirmed over hundreds of executions of the experiments. Whilst the optimisation method does not guarantee the parameters are a global minima of the drift, there is high confidence they represent a solid solution. The resulting belief parameters are shown in Table 7.8

Looking at the table above, we notice that, in general, the chosen attributes have little influence on the Opportunity belief. Providers that have uploaded personal documents are likely to be judged as having higher opportunities to do the job by a factor of 19%. The other parameters influence the opportunity belief close to 0. On the other hand, providers with a positive sentiment on their about me section are likely to be judged high on their ability by a factor of 24% (i.e. the about me sentiment contributes 24% to the overall

Table 7.8: Optimised Belief Parameters

	Opportunity Belief	Ability Belief	Willingness Belief
About Me Sentiment	-3.47e-06	0.2411	-0.0083
Profile Picture Happiness Score	-1.01e-07	0.0103	-0.0031
Services Local Schools	-1.23e-07	0.0292	0.0009
Has Qualifications	7.16e-06	-0.2678	-0.0118
High Timetable Availability	-2.31e-06	0.0185	0.0218
Recent Updates Timetable Availability	9.61e-07	0.0024	0.0024
Has Uploaded Personal Documents	0.1906	-0.0358	-0.0358

judgment of ability in the trust evaluation).

The belief parameters are shown in table 7.8 are the result of observation based on how the attributes contributed to trust for providers that the community has already trusted. Whilst we only have less than 10% of providers having received any feedback on the platform, we can now calculate the CF Trust values for all providers. Figure 7.14 plots Trust Observations and CF Trust calculated with random parameters and CF Trust calculated with the optimised parameters.

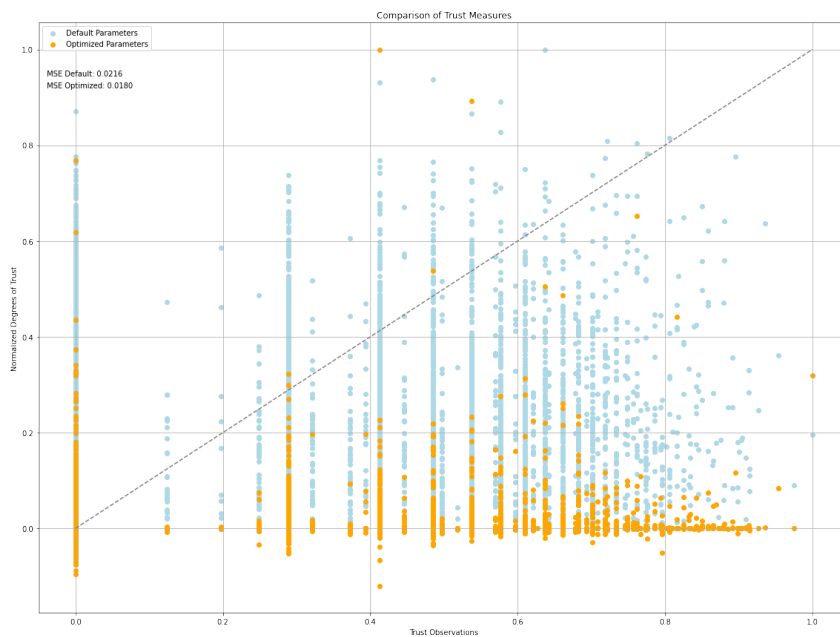


Figure 7.14: Comparison of CF Trust with random and optimised parameters

Whilst regression models and non-linear optimisation help calibrate belief coefficients, we use factorisation machines to look at similarities among attributes and determine if trust patterns can emerge for providers with similarities.

Households	Region	TrustCapital	TrustCapital/Household
3521483	London	6158.859684	174.893921
3934639	South East	1402.256877	35.638768
2652878	East of England	625.644869	23.583628
2383082	South West	531.846623	22.317596
3103743	North West	639.411579	20.601306
2530062	West Midlands	419.348362	16.574628
2361567	Yorkshire and The Humber	382.643381	16.202944
1142240	North East	167.124897	14.631329
2185457	East Midlands	252.483454	11.552890
598514	South East Wales	66.912271	11.179734
693329	Northern Ireland	66.088602	9.532070
315757	South West Wales	12.123391	3.839469
290634	North Wales	5.647836	1.943281

Table 7.9: Social Capital score per 100k households

7.4 Degrees of Trust predictability

In the previous section, we used the mathematical representation of Castelfranchi's Degrees of Trust to calculate its values across the childcare service provider network. We also used the existing knowledge about expressed trust on the network to calibrate the belief parameters. Finally, we used the best-performing belief parameters closely related to the number of received reviews on the network to recalculate and eventually normalise CF Trust scores between 0 and 1. This section explores whether the CF-Trust scores are good predictive scores for future jobs.

In Table 7.9, we can see the ranking of social capital observed for every 100k households grouped by region. An interesting observation in this table is that the ranking closely matches the GDP per person for each region. In fact, according to gov.uk¹⁹ the top 8 regions by GDP pro capita yield are: 1. London - 55,974 2. South East - 34,516 3. East of England - 29,176 4. North West - 28,257 5. South West - 28,012 6. West Midlands - 26,281 7. East Midlands - 25,956 8. Yorkshire and The Humber - 25,696

The strong correlation is confirmed by the Pearson Correlation Coefficient [106] comparing the two arrays of CF-Trust and GDP per person ordered by UK Regions.

¹⁹ons.gov.uk/economy/grossdomesticproductgdp

The results show a correlation value of 0.98 and a P-Value close to 0²⁰. The small P-Value is an indication we should reject the null hypothesis that the correlation is due to random noise. This is an important independent quantification and validation of the social science concepts discussed by [15] about Social Capital, its effect on economic welfare, and how trust is a composing component of Social Capital.

One of the characteristics of the high trust demand social networks of needs is their sparsity in reviews. The childcare network claims to be trusted by millions of parents, and there are over a hundred thousand service providers, yet we only found about 24k reviews on the platform.

Given these factors, we use factorisation machines, known to perform well in sparse conditions, to run an experiment on CF-Trust predictability. Unlike common uses of factorisation machines, where every pair user/product is considered possible, our FM model has to overcome the fact that a parent requesting childcare in Brighton will not select a provider in Manchester, however high its trust score might be.

Adjacent locations, however, share providers (i.e. it is common for providers to offer their services in more than one adjacent location).

There are 603 childcare service providers in W11 and 653 providers in W12 in west London. Of these, 114 offer their services in both locations. We build a factorisation machine model that uses data from one location (for example, W11) to train the model. Parents have a relationship with providers they have left a review, and the provider CF-Trust score is used as the training target. We used pyFM²¹ library, a python wrapper that uses polylearn from scikit-learn.org [107] [108]²², to build and run the childcare FM [41] model.

We created around 300 random bigrams of postcodes in London and used the first postcode to train the factorisation machine model and the second one to test the model. We used the review information as a feature. More specifically, each review indicates a relation, the review overall score is a feature together with the review body sentiment

²⁰More specifically, the Pearson Correlation Coefficient calculation results are: `PearsonRRresult(statistic=0.9806843322334651, pvalue=1.1014217085859774e-07)`

²¹<https://github.com/coreylynch/pyFM>

²²<https://contrib.scikit-learn.org/polylearn/index.html>

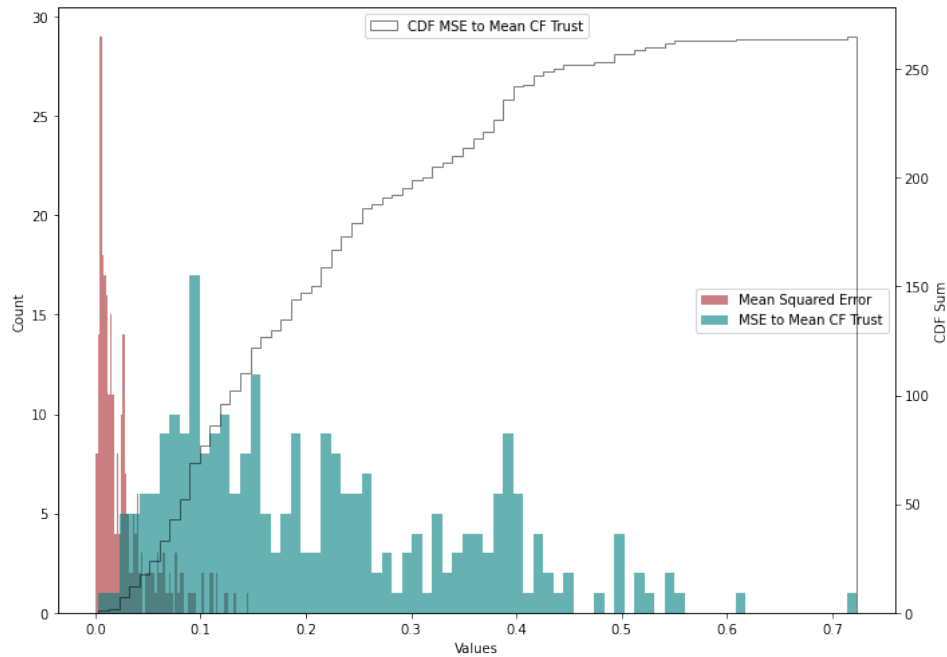


Figure 7.15: MSE distribution for FM Predictions

Table 7.10: Data for building the factorisation matrix

parent	provider	review	review_positivity	ts	Y-Train
Nina-1657134	1422990	4.0	0.171	0.321622	0.373482
Chantell-2093653	1422990	5.0	0.341	0.385405	0.373482
Miss Fig-107663	1422990	5.0	0.193	0.385946	0.373482
Siobhan-3195215	1949846	5.0	0.535	0.893514	0.015453
Tracy-3428907	1949846	5.0	0.423	0.991081	0.015453
Nne28-3428971	1949846	5.0	0.347	0.991081	0.015453
Jozza-3187611	3187213	5.0	0.252	0.869730	0.042095
Joline-3325786	3187213	5.0	0.228	0.939459	0.042095
claudia-2403994	77247	5.0	0.354	0.622432	0.047503
Jenny-2008565	1935388	5.0	0.184	0.535405	0.041760

positivity score. The review date is the time dimension.

Table 7.10 shows 10 records of reviews from parents to providers in the childcare network, the respective overall review score, the positivity sentiment of the body of the review, the normalised timestamp and the target CF-Trust used for training the model. Figure 7.16 shows 10 records of the resulting factorisation matrix.

We captured the results as mean squared error deviation from the actual value of CF Trust in the test set. Figure 7.15 shows the distribution of MSE, with the vast majority

	0	1	2	3	4	5	6	7	8	9	...	87	88	89	90	91	92	93	94	95	96	
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.472	0.980611
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.345	0.980611
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.313	0.826498
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.246	0.847626
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.260	0.949043
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.336	0.949043
6	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.422	0.982849
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.128	0.969923
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.214	0.819538
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.184	0.831220

10 rows x 97 columns

Figure 7.16: Factorisation Matrix

Table 7.11: MSE of FM Predictions for the top 10 postcodes with the highest combined households

TR	TE	TR P	TE P	MP	MSE	CF	H
CR0	N7	103	719	0	0.029065	0.126612	82733
IG8	CR0	113	103	0	0.031152	0.119070	76139
W11	N1	653	745	1	0.014055	0.131662	56586
E17	W11	341	653	0	0.015022	0.100584	54047
E14	IG11	372	107	0	0.001329	0.054233	52607
N1	W7	745	258	0	0.023297	0.107300	52197
NW1	E1	448	570	2	0.026743	0.116833	51460
SE9	NW3	131	415	0	0.015527	0.115321	49097
WD23	E17	53	341	0	0.007352	0.083552	48749
SW19	SM6	284	156	0	0.036302	0.090174	47852

of values falling between 0 and 0.1. In addition, Figure 7.15 also shows how big in % MSE is compared to the Mean value of CF Trust for the test data. The results indicate that most predictions predicted a CF Trust value that deviates no more than 20% of its actual calculated value.

In the following tables we show the MSE of FM prediction ordered by number of combined households in the training and testing postcodes,

For each table, the column names have the following meaning:

- **TR:** Postcode for the training data
- **TE:** Postcode for the testing data
- **TR P:** Number of providers in the training postcode

Table 7.12: MSE of FM Predictions for the top 10 postcodes with the highest matching providers

TR	TE	TR P	TE P	MP	MSE	CF	H
EC4P	EC1R	519	554	470	0.002242	0.099345	2426
EC4N	EC3P	317	345	215	0.000653	0.198210	11
EC4Y	WC1V	292	377	200	0.007716	0.129610	280
W1G	WC1E	291	373	187	0.002419	0.067704	2365
WC1X	EC2P	568	340	185	0.069202	0.188852	3308
WC2B	W1W	265	309	124	0.000888	0.049380	2876
WC1A	EC1V	314	576	93	0.017892	0.140215	6604
W1S	WC2A	188	315	67	0.002773	0.086158	298
W1H	SW1Y	280	220	32	0.004620	0.047597	3676
NW1W	EC2Y	478	395	5	0.064173	0.170527	1840

Table 7.13: MSE of FM Predictions for the results with the 5 highest and 5 lowest MSE

TR	TE	TR P	TE P	MP	MSE	CF	H
EN1	HA8	198	158	0	0.145398	0.344660	40272
SE17	N6	613	215	0	0.132707	0.316327	22129
HA5	IG6	74	74	0	0.124925	0.292546	28195
SW1Y	TW9	220	205	1	0.123200	0.248717	10280
SW11	CR3	571	45	0	0.115857	0.386293	43040
...							
EC4N	EC3P	317	345	215	0.000653	0.198210	11
WC2B	W1W	265	309	124	0.000888	0.049380	2876
RM1	W1B	89	241	1	0.001015	0.041394	8901
HA0	W1A	193	295	0	0.001311	0.049380	14985
E14	IG11	372	107	0	0.001329	0.054233	52607

- **TE P:** Number of providers in the testing postcode
- **MP:** Matching providers offering services in both training and testing postcodes
- **MSE:** Mean Squared Error of FM predictions compared with the calculated Castelfranchi Degree of Trust Score
- **CF:** Mean value of calculated Castelfranchi Degree of Trust for all providers in the test postcode
- **H:** Total number of households in the combined postcodes

Prediction seems to be best (i.e. lowest MSE) for postcodes that have some providers providing in both postcodes, whilst the highest MSE values are observed in postcodes without any matching providers.

One of the best performing prediction is observed when the model is trained with data from *WC2B* and tested on *WIW* with MSE to observed CF-Trust score ratio of only 0.0009 . There are 265 providers in the training set and 309 providers in the testing set with 124 matching providers (i.e. providers offering services in both locations). There are a combined 2876 households in these two postcodes. One of the next best predictions with similar MSE performance, is trained with providers in east London (Poplar and Canary Wharf, E14) and tested on postcode IG11 (Barking) with a combined 52607 households and no providers offering services in both locations. There is a distance of 12 miles (ca. 19 km) between the two locations.

The top 5 worst performing predictions, with MSE values between 0.145398 and 0.115857 and MSE to CF-Trust score ratio between 0.422 and 0.300, are seen when training and testing postcodes that do not have any providers offering services in both postcodes (i.e. there is only one provider shared among all 5 tuples) and in general with a high number of Households per location.

7.5 Conclusions

In this chapter, we run a social network analysis for four high-trust-demanding social networks of needs. We found that these networks composed of triads *caretaker*, *review*, *provider* are disconnected networks with a high number of communities detected and centred around providers. The network becomes connected when location nodes are inserted. Closeness centrality has the lowest variability with a standard deviation of 0.0132, backed by a consistent eccentricity measure of 8 hops: *parent* \rightarrow_1 *review* \rightarrow_2 *provider* \rightarrow_3 *location* \rightarrow_4 *region* \rightarrow_5 *location* \rightarrow_6 *provider* \rightarrow_7 *review* \rightarrow_8 *parent*. Betweenness centrality scores high for providers with high number of reviews received. Eigenvector centrality instead scores higher in providers that have received reviews and

provide their services in multiple locations.

The multiplex care network seen in Figure 3.5 highlights differences between the different care networks. Doctify is denser by location, and most providers offer their services in numerous locations, whilst by contrast in childcare networks, providers only offer services in a few adjacent locations. The centrality measures patterns also look different for the multiplex network, highlighting the heterogeneity of the network.

Existing measures, however, are centred on relations and received reviews. We used Equation 3.2 to calculate a trust measure based on Castelfranchi's Degrees of Trust. Its main components are: Opportunity, Ability and Competence, Intent and Persistence. Such measure does not need prior knowledge on existing reviews. This overcomes the discrepancy observed where the care network tends to have a low review ratio, and most providers have not received any reviews.

Yet, the existing reviews are an indicator of expressed trust and a good validator for such measure. DoT performs substantially better than random trust. We used ordinary least squares regression to calibrate belief coefficients and derive a better performing trust measures, when compared to the observed expression of trust (i.e. reviews received).

We used factorisation machined to evaluate the predictability of the CF Trust and in general found that the model and measure is a great predictor with consistent low mean squared error values.

Chapter 8

Conclusions, limitations, and future work

A non-reductionist model of trust composed of one's beliefs about other agents' opportunity, ability and competence, intent and persistence to achieve a certain goal τ , can be represented mathematically by Equation 3.3

$$DoT_{x,y,\tau} = \sum_{i=1}^n \alpha_{i,x,y} T_{iy} \cdot \sum_{i=1}^n \beta_{i,x,y} T_{iy} \cdot \sum_{i=1}^n \gamma_{i,x,y} T_{iy} \quad (8.1)$$

In the high trust demanding care social network of needs (3.1.1), where τ is the action of taking care, we showed that such calculation yields trust prediction results that perform substantially better than random and the results are comparable with other centrality measures.

Unlike other centrality measures, however, the DoT trust score (Degree of Trust or other times referred to as CF-Trust - an abbreviation for Castelfranchi Trust) is only based on provider (i.e. Trustee) features. It does not presume previous knowledge or experience of trust exchange. In the care network, an earlier decision to trust is expressed in caretakers (i.e. Trustors) leaving reviews to providers. Hence, unlike other centrality measures, the CF-Trust score scores all providers in the network, regardless of whether these have previous reviews.

This is an important outcome for high trust demanding networks where reviews are rare and sparse. In the childcare network composed of over 100thousands¹ of providers and

¹We only have observed about 80k providers; however, these are dynamic, and accounts are closed

millions of parents, only 24k reviews are found. We consistently saw the number of reviews and interactions decrease when the demand for trust presumably increased.

In Equation 3.3 the coefficients $\alpha_{i,x,y}$, $\beta_{i,x,y}$ and $\gamma_{i,x,y}$ are the belief parameters. That is, they represent the agent's x belief of how much the i -th attribute of the agent y contributes to the overall trust, in its non-reductionist part: opportunity, ability and competence, intent and persistence.

While we initiated these parameters with normalised values, using non-linear differential evolution and the expressions of trust on the childcare network, we were able to derive better-performing belief parameters. On the childcare network, empirical evidence indicates that parents believe providers serving local schools are more trustworthy, and the belief coefficient contributes higher to the opportunity component of trust. The ability belief coefficients (i.e. β) for an image with a dominant happy feeling is higher than the willingness belief (i.e. γ). The sentiment of the 'about me' text consistently contributes negatively to all three trust components but does so with less strength for the willingness component. When used to re-calculate CF Trust, the calibrated belief parameters yield results that match closer expressions of trust seen on the platform as the number of reviews received by each Trustee.

According to previous research, social trust and social capital concepts are closely related. The latter can be derived from the former, and social trust is a crucial component of social capital. When aggregating the total CF-Trust calculated for each region in the UK and calculating a per 100K household value, the ranking closely relates to the UK GDP per head ranking 7.9.

The CF-Trust is a good predictor of trust in adjacent locations when used in factorisation machines with consistently low mean squared error results. In an experiment with bigrams on London postcodes, we found that among over 250 pairs where one location is used for training the FM model and the other for testing, adjacent locations with some providers offering services in both locations yield better predictions with lower MSE.

We retrieved data from several diverse locations that conformed to the characteristics of online social networks of needs. We categorised these by the level of trust needed to cooperate. We combined data from 4 high trust demanding networks (i.e. childcare.co.uk for childcare, rover.com for pet care, doctify.com for health care and carehome.co.uk for elderly care) and created a multiplex network. Each layer of the multiplex does not have shared providers, but they share the locations; hence, the locations were the extradimensional links. On other networks of needs, for example, the home care networks, providers are shared and detectable across the different networks 7.2. The heterogeneity of the selected care networks dictates the unlikelihood of shared providers (i.e. it is unlikely a provider offering pet care services will also provide health care services). The multiplex network analysis highlights some key differences even among networks where the cooperation thresholds are similar.

There are significant differences in the distances healthcare providers offer their services compared to childcare providers. There are differences in the number of reviews received; the network density and multiplex centrality measures results look substantially different compared to the childcare network centrality measures.

Apart from data from the care network and the home care network (checkatrade.com and trustatrader.com), we run a data collection campaign towards the infamous 4chan.org/pol board. Whilst this is a non-OSSNs conforming network, it has characteristics (ephemerality, anonymity, slang) that make for a challenging task for data retrieval. We collected 4chan data from three sources: the web, the 4chan API, and the 4plebs API (a web archiving service for 4chan). The data and patterns observed conformed to previous knowledge about 4chan. Frequency of publications, threads' time to live, activity by time of the day and day of the week and length of thread bump were observable and confirmed in previous research. In a three-way data comparison, we showed that 4plebs API has accurate and comprehensive copies of 4chan data. This partially defies the ephemerality claim of 4chan. Whilst threads continue to disappear from 4chan, a copy is permanently saved on 4pleb and is always retrievable.

Further to testing our system for data retrieval, we showed that the high-frequency retrieval of data from the web (as seen by end users) captured threads and posts that only existed on the web and were not observed on the 4chan API or 4plebs API. We believe these to be short-lived threads that were deleted/removed by 4chan janitors before being observed by the polling services of 4plebs. Over six months, we found that about 0.5% of threads and posts were deleted. Further, we found that the sentiment of the deleted threads and posts was more moderate compared to what the sentiment of the content of the live board looks like.

This work, among others, proved the worthiness of a query-based architecture, system design and implementation for a distributed system capable of collecting large amounts of data for extended periods. This system was consistently used across the work of this research and was one of our early goals when looking to solve one of the dimensions and challenges when working with trust: trust information collection: 2.4. The ability of user queries to extract data is guaranteed by two query engines: OXPath and DR-Web-Engine. OXPath, developed at Oxford, is an extension of XPath, and the query structure is similar to XPath. It is implemented in Java and wraps around the Selenium web driver.

DR-Web-Engine (<https://pypi.org/project/dr-web-engine/>), developed as part of this research, conforms to the OXPath design, is implemented in Python, and is a wrapper of multiple underlying drivers (including Selenium web driver). However, unlike OXPath, it has a JSON-like query structure, which is closely related to the output.

8.1 Limits and Future work

The distributed data retrieval system

During this research project, we identified many social networks of needs in each category discussed in Chapter 3. The underlying query engines present two fundamental limits that sometimes make it difficult to have a wide spread of data from different networks. OXPath uses a fixed version of Selenium with an embedded fixed version of Firefox. Both these are now obsolete, and most social networks use newer technologies and have stopped support for older browsers. DR-Web-Engine was developed to overcome this issue; however, its implemen-

tation does not yet support all XPath keywords and needs further development.

Another challenge to the system's scalability and sustainability is its reliance on manual analysis, attribute identification, search paths, form filling, and finally, the building of web queries for data extraction. These activities are time-consuming and challenging to scale. Furthermore, as web page structures are updated, the web queries need maintenance. A possible approach to addressing this issue is adopting automatic full-site extraction (also known as AFE) techniques. Whilst a complex problem, XPath - our underlying query engine, has been used in AFE systems where extraction queries have been generated with up to 97% attribute accuracy discussed by Furche et al. [80]. Another AFE approach uses search/results/record patterns. In our data extraction queries, we largely use this pattern. Adopting redundancy-driven data extraction [81] for automatic query generation can address the scalability and sustainability of the distributed system.

The system deployment and monitoring used the free tier of platform9.com services. From November 2022, Platform9 withdrew its free tier. Whilst these services are not strictly necessary to deploy and run the data retrieval projects, they offer observability, monitoring and easy deployments. In future work, these services should be replaced with alternative open-source services such as Rancher ², Grafana ³ and Prometheus ⁴.

Whilst most system operations are achievable via CLI, the system usability would benefit from a web interface for widespread adoption.

Data and datasets

We only collected data from 6 online social networks of needs. Four are in the high trust category, and two are in the medium trust. While we have already published a few of the datasets, and the rest are accessible via MongoDB and Neo4j on university servers, the remainder of the datasets need to be cleansed and anonymised, prepared for publication, and published via the university ePrint services. While we analysed the last copy of

²<https://www.rancher.com/>

³<https://grafana.com/>

⁴<https://prometheus.io/>

the data, we ran the data campaigns for extended periods, with multiple copies offering insight into how data evolves.

Data retrieval work should be extended to the remaining identified OSNNs or when new ones are identified. In the future, a multiplex network comprised of all online social networks of needs can confirm and give a higher confidence in some of the results seen with social capital 7.9.

boards.4chan.org/pol analysis

Whilst there is high confidence in the validity of the three-way comparison and that 4plebs offers holistic coverage of the 4chan threads, the results on the deleted threads and posts present some further challenges. 4plebs API comes with strong throttling limits. These are managed on the data retrieval code; however, false positive cases can happen where a thread is not returned because of throttling rather than not being present on the archiving services. Even more so, when retrieving threads from the 4chan API, there is only a limited window when a thread is available for retrieval before its deletion. The number of deleted threads and posts might contain errors. The communication on 4chan is picture and meme-driven. Only short sentences are used in thread exchanges, and previous work has identified that 4chan uses its slang [93]. These characteristics make topic modelling and sentiment analysis challenging. Future work should validate the sentiment analysis outcomes on 4chan deleted threads.

Network Analysis of the care network

We analysed the childcare network in-depth and built a multiplex graph with the four sources from the care network. In future work, the network analysis should be expanded to all four sources of the care network. Comparing the social network analysis results can provide further insights.

Appendix A

Code

In our deployment the Kubernetes Cluster was composed of one master node and six slave/worker nodes¹ as shown in detail in **table A.1**. Kubernetes is also available on all major public cloud providers² with advanced web interfaces management as well as CLI directly connected to the cloud instance.

A typically project deployment is shown in **figure A.1**

¹Due to the low workload on the master node, we added the master to the worker nodes

²Microsoft Cloud (Azure): <https://azure.microsoft.com/en-gb/services/kubernetes-service/>, Google Cloud Platform (GCP) : <https://cloud.google.com/kubernetes-engine>, Amazon Web Services (AWS): <https://aws.amazon.com/kubernetes/>

Table A.1: Kubernetes Cluster

Kubernetes Cluster Nodes				
Name	Version	OS Image	Kernel-Version	Container-Runtime
cspc537-lx	v1.17.2	Ubuntu 18.04.4 LTS	4.15.0-88-generic	docker://18.9.7
k8s-master	v1.17.0	Ubuntu 18.04.3 LTS	4.15.0-72-generic	docker://19.3.5
k8s-slave	v1.17.0	Ubuntu 18.04.3 LTS	4.15.0-72-generic	docker://18.9.7
k8s-slave2	v1.17.0	Ubuntu 18.04.3 LTS	4.15.0-72-generic	docker://18.9.7
k8s-slave3	v1.17.0	Ubuntu 18.04.3 LTS	4.15.0-72-generic	docker://18.9.7
k8s-slave4	v1.17.0	Ubuntu 18.04.3 LTS	4.15.0-72-generic	docker://18.9.7

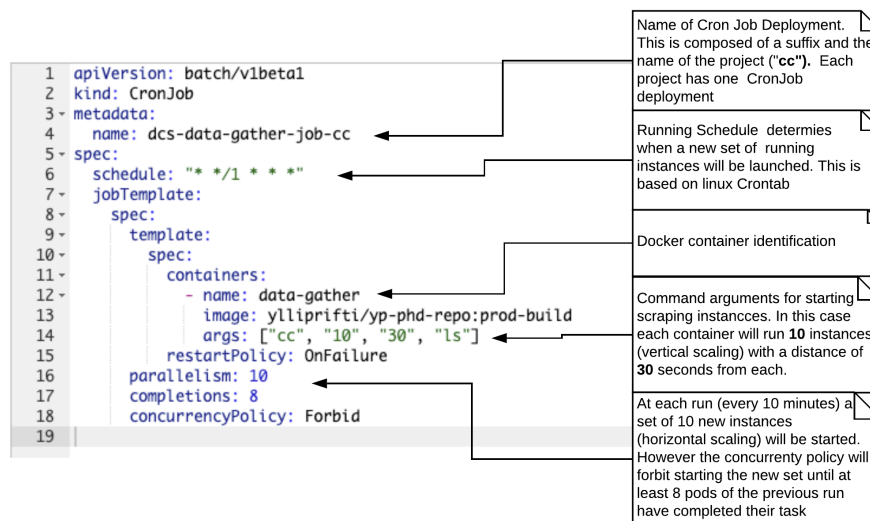


Figure A.1: Kubernetes CronJob deployment of a Scraping Project

- **Ubuntu Docker Containers**
- *Project Configuration Loader*

The following python code is the implementation for loading the project configuration into an in memory instance of the ProjectConfiguration entity.

Listing A.1: Python - Project Configuration Loader

```

1 from py.core.entity
    .ProjectConfiguration import ProjectConfiguration, ProjectCollection
2 from py.core.interfaces
    .IProjectConfigurationLoader import IProjectConfigurationLoader
3
4 from interface import implements
5
6 import os
7
8 EXTRACT_EXT = '.extract'
9 DISCOVERY_EXT = '.discovery'
10 ARCHIVE_EXT = '.archive'

```

```

11
12
13 class
    ProjectConfigurationLoader(implements(IProjectConfigurationLoader)):
14
15     def __init__(self, projects_directory: str):
16         self.projects_directory = projects_directory
17
18     def load_project_configuration
    (self, project_name: str) -> ProjectConfiguration:
19         """
20
21         loads the project folder and file structure into ProjectConfiguration
22         :param project_name: name of the project to load
23         :return: an ProjectConfiguration
24         instance holding project collections and query pathss
25         """
26         project_directory = list(filter
    (lambda x: x == project_name, os.listdir(self.projects_directory)))
27
28         if len(project_directory) == 0:
29             return None
30
31         project_configuration = ProjectConfiguration(project_directory[0])
32         full_project_path = "{}/{}"
33         ".format(self.projects_directory, project_configuration.project_name)
34         project_configuration.working_directory = full_project_path
35
36         extract_folders = [d for d in os.listdir(full_project_path)
37
38             if os.path.isdir("{}{}".format(full_project_path, d))]

```



```
35
36     for folder in extract_folders:
37         collection = ProjectCollection.ProjectCollection(folder)
38
39         all_files = os.listdir("{}{}".format(full_project_path, folder))
40         extract_file
41         = list(filter(lambda x: x.endswith(EXTRACT_EXT), all_files))
42         discovery_files
43         = list(filter(lambda x: x.endswith(DISCOVERY_EXT), all_files))
44         archive_files
45         = list(filter(lambda x: x.endswith(ARCHIVE_EXT), all_files))
46
47         if len(extract_file) > 0:
48             collection.extractor = extract_file[0]
49
50         if len(discovery_files) > 0:
51             collection.discovery = discovery_files
52
53         if len(archive_files) > 0:
54             collection.archive = archive_files[0]
55
56         project_configuration.collections.append(collection)
57
58     return project_configuration
```

Code snipped from the query manager

Listing A.2: Python - OXPath Query Manager snippet

```
1
2 from py.core.interfaces.QueryManager import QueryManager
3 from interface import implements
4
5 class OxpathQueryManager(implements(QueryManager)):
```

```

6
7     def __init__(self, project_configuration: ProjectConfiguration):
8         self.__project_configuration = project_configuration
9
10    def
11
12        get_extraction_query_path(self, url: str, collection: str) -> str:
13            """
14                Used to generate the next runnable XPath
15                extraction query. Uses the project configuration and the XPath
16                Extraction Query provided
17                . Clones the file and injects the current :param url provided.
18                :param url: The url to inject in the extraction query.
19                :param collection: The working project
20                data-collection where the current set of XPath queries are found
21                :return:
22                The path to the updated Extraction query that is ready for execution
23            """
24            logging.info("=> :: OxpathQueryManager
25                .get_extraction_query_path :: -> Searching for collection: {} in "
26                    "project configuration")
27
28            collection_item = [x for x in self.__project_configuration
29                .collections if x.collection_name == collection]
30
31            if collection_item is None or len(collection_item) != 1:
32                logging.warning("=> :: OxpathQueryManager
33                    :: -> Collection not found in project configuration")
34
35            return None
36
37            extractor_folder = "{}/{}
38            ".format(self.__project_configuration.working_directory, collection)
39
40            logging.info("=> :: OxpathQueryManager

```

```

.get_extraction_query_path :: -> Extraction Folder: {0}"
29         .format(extractor_folder))
30
31     extractor_file
32     = "{}/{}".format(extractor_folder, collection_item[0].extractor)
33
34     logging.info("==> :: OxpathQueryManager
.get_extraction_query_path :: -> Extractor file: {0}"
35         .format(extractor_file))
36
37     temp_file = OxpathQueryManager
38     __clone_file(extractor_folder, extractor_file, url)
39
40     return temp_file

```

Locker implementation

In addition to the *continues campaigns*, discrete campaigns can be run. In the discrete case extraction campaign are identified by a number. It is expected that each URL in the queue to be processed (i.e. selected by the extractor) once for the running campaign. That means that independently of the queue sorting rules, lower priority URLs will jump to the top of the queue once all other URLs have been processed for that campaign. Once all URLs have been processed the campaign identified will move next by adding one to the campaign identifier. These logic is reflected by two separate implementations of the DistributedLocker interface even though most of the implementation is common and incorporated in the AbstractDistributedLocker:

Listing A.3: Code snippet of DistributedLocker implementation

```

1
2 from py.core.interfaces.IDistributedLocker import IDistributedLocker
3 from interface import implements
4 logging = LoggingUtils.get_logger()
5
6 class DistributedLocker(AbstractDistributedLocker):

```

```
7
8     def __init__(self, project_store
9         : ProjectMongoStore.ProjectMongoStore, collection_name: str):
10         super().__init__(project_store, collection_name)
11
12     def next(self, batch_size: int = 1):
13         return super().next(batch_size)
14
15     def next_archive(self):
16         return super().next_archive()
17
18     def current_campaign(self) -> int:
19         pass
20
21     def move_next_campaign(self, item_id) -> int:
22         pass
23
24 class AbstractDistributedLocker(implements(IDistributedLocker)):
25     def next(self, batch_size: int = 1) -> str:
26         if batch_size == 1:
27             return self.__next__()
28
29         items = list()
30         for _ in range(0, batch_size):
31             items.append(self.__next__())
32
33         return items
34
35     def __next__(self):
36
37         match_query = {"$match": {"$and": [{"$or": [{"attributes.deleted":
```

```
37     {"$exists": False}}, {"attributes.deleted": {"$eq": False}}}],
38
39     {"$or": [{"attributes.archived": {"$exists": False}},
40             {"attributes.archived": {"$eq": False}}
41             ]}}]}}}
42
43     project_query
44 = {"$project": {"_id": 1, "url": 1, "last_extraction": 1}}
45
46     sort_query = {"$sort": {"last_extraction": 1}}
47
48     limit_query = {"$limit": 1}
49
50     pipeline = [match_query, project_query, sort_query, limit_query]
51
52     next_item = list(self.__project_store.run_aggr(pipeline))
53
54     if next_item is None or len(next_item) == 0:
55         return None
56
57     selected_item_id = next_item[0]["_id"]
58
59     try:
60         nr_items_updated = self.__mark_next_item(selected_item_id)
61         if nr_items_updated > 0:
62             return next_item[0]['url']
63         return None
64     except Exception as e:
65         logging.error("Error: {}".format(e))
66         return None
```

The following is code snipped from the implementation of the OXPath runner interface.

Listing A.4: Runner Interface

```

1
2
3 class OxpathRunner(implements(Runner)):
4     def __init__(self, config: OxpathConfig.OxpathRunnerConfigurator):
5         self.config = config
6         self.__scheduler = sched.scheduler(time.time, time.sleep)
7
8     def run_default(self):
9         print(self.config.oxpath_binary)
10        return sb.call(['java', '-jar', self.config.oxpath_binary])
11
12    def run_oxpath_raw(self, *oxpath_params: str):
13
14        logging.info('==> ::->:: Starting Oxpath Run ')
15
16        params_array = ['java', '-jar', self.config.oxpath_binary]
17        for item in oxpath_params:
18            params_array.append(item)
19
20
21        logging.info('==> ::->:: Callable pipeline: {}'.format(params_array))
22
23        callable_pipe =
24        sb.Popen(params_array, stdin=sb.PIPE, stdout=sb.PIPE, stderr=sb.PIPE)
25
26        try:
27            outs, errs = callable_pipe.communicate(timeout=3600)
28            return outs, errs
29
30        except sb.TimeoutExpired:

```

```
logging.warning('==> ::->:: TIMEOUT Failed to complete pipeline ')
29
30     callable_pipe.kill()
31     return callable_pipe.communicate()
32
33 def extraction_runner(self, url: str, collection_name: str,
34                       project_configuration
35 : ProjectConfig.ProjectConfiguration, clear: bool = True):
36
37     logging.info('==> :: Extraction Runner: {}, project
38 : {}'.format(collection_name, project_configuration.project_name))
39
40     query_manager
41 = QueryManager.OxpathQueryManager(project_configuration)
42
43     _query_path
44 = query_manager.get_extraction_query_path(url, collection_name)
45
46     logging.info('==> :: Query Path: {}'.format(_query_path))
47
48     output, error = self.run_oxpath_raw
49 ('-q', _query_path, '-f', 'JSON', '-mval', '-jsonarr', video_buffer)
50
51     if clear:
52
53         logging.info('==> :: Clearing query: {}'.format(_query_path))
54         QueryManager.OxpathQueryManager.clear(_query_path)
55
56     logging.info('==> ::
57 Query Complete with output: {} and error: {}'.format(output, error))
58
59     return {"output": output, "error": error}
```

Storage retry mechanism code snipped:

Listing A.5: Storage operations - dealing with availability

```
1 from enum import Enum
2 import time,sys
3 import py.core.utils.LoggingUtils as log_utils
4 logger = log_utils.LoggingUtils.get_logger()
5
6
7 class DelayMethod(Enum):
8     """
9     Enumeration to define Delay Methods
10    """
11    LINEAR = 1
12    FIBONACCI = 2
13    POWER_LAW = 3
14    NO_RETRY = 4
15
16
17 class DelayRetry(object):
18     def __init__(self, method: DelayMethod, MAX_RETRY: int):
19         self.retry_function = self.fibonacci
20         self.max = MAX_RETRY
21         self.nr_retries = 0
22         if method == DelayMethod.LINEAR:
23             self.retry_function = self.linear
24         elif method == DelayMethod.FIBONACCI:
25             self.retry_function == self.fibonacci
26         elif method == DelayMethod.POWER_LAW:
27             self.retry_function = self.power_law
28
29
30     def retry(self, func, args):
```



```
31     """
32     retry - executes all [function references
    ] or [functions references by name] as string in the func array.
33     In case of exception will retry execution
    for a number of times based on retry_function defined at object
34     instantiation.
35
36     This method is useful when dealing with unreliable
    sources. For example when calling an API endpoint that can
37     become unavailable for short periods
    of time, or when calling a distributed database that is not always
38     available
39
40     :param func: Function reference or function name
41     :param args: Array of parameters for each function in func
42     :return:     the return of the last func called
43     """
44     self.nr_retries = 0
45     logger.info("Retry
call initiation: Functions: {}, Parameters: {}".format(func, args))
46     while True:
47         try:
48             for idx in range(0, len(func)):
49                 if idx == 0:
50                     temp_result = func[idx>(*args[idx])
51                 else:
52                     # logger.info(*args[idx])
53                     try:
54
55                         method = temp_result.__getattr__ (func[idx])
56                         if method is None:
```

```

        logger.error("Method was None on Index: {}".format(idx))
57         return None
58     except:
59
60         logger.error("Method not found on Index: {}".format(idx))
61         return None
62         temp_result = method(*args[idx])
63         return temp_result
64     except:
65         ex_type, ex_value, ex_traceback = sys.exc_info()
66         logger.warning("Exception Type: {}. Exception Value:
67 {}, Exception Traceback: {}".format(ex_type, ex_value, ex_traceback))
68         self.nr_retries += 1
69         time_to_sleep = self.next(self.nr_retries)
70         logger.warning
71 ("Retry Tentative: Retries {}, Time to Sleep: {}, Functions: {},
72 Parameters: {}".format(self.nr_retries, time_to_sleep, func, args))
73         if time_to_sleep >= 0:
74             time.sleep(time_to_sleep)
75         else:
76             logger.warning("Retry end and return None")
77             return None
78
79 def next(self, current: int) -> int:
80     value = self.retry_function(current)
81     return value if value < self.max else -1
82
83 def no_retry(self, current: int):
84     return -1
85
86 def fibonacci(self, current: int) -> int:
87     if current == 0 or current == 1:

```

```
84         return current
85     return self.fibonacci(current - 1) + self.fibonacci(current - 2)
86
87     def linear(self, current: int) -> int:
88         return current
89
90     def power_law(self, current: int) -> int:
91         if current <= 1:
92             return 1
93         return current * (current - 1)
94
95     def retries(self):
96         return self.nr_retries
97
98     def __enter__(self):
99         return self
100
101     def __exit__(self, exc_type, exc_val, exc_tb):
102         pass
```

Storage Interface

Listing A.6: Storage operations interface

```
1 from interface import Interface
2
3 class StorageConnector(Interface):
4
5     @property
6     def client(self):
7         pass
8
9     @property
10    def database(self):
```

```
11     pass
12
13     @property
14     def collection(self):
15         pass
16
17     def get_db(self, db_name: str):
18         pass
19
20     def get_col(self, col_name: str):
21         pass
22
23     def run_aggr(self, pipeline: list):
24         pass
25
26     def run_find(self, query, projection=None, sort=None, limit=-1):
27         pass
28
29     def run_update(self, filter_qry, payload):
30         pass
31
32     def run_insert(self, payload):
33         pass
34
35     def run_update_one(self, filter_qry, payload):
36         pass
37
38     def run_insert_one(self, payload):
39         pass
40
```

A.1 Data collection code snippets

Listing A.7: XPath generated discovery query - rover.com

```

1 doc('https://www.rover.com/search/?...')
2
3     //input[@id="location-input-sidebar"]/{@N12}/{presenter /}
4     /(//a[@aria-label="Next page"]/{nextclick/})*
5     //div[contains(@class, 'SearchResultsWrapper__Results
6     ')]//a[contains(@class, "NameRow__NameLink")]:<links>
7     [
8         .:<link=qualify-url(@href)>
9         :<discovery_payload> [
10            .[ //a[contains
11            (@class, "NameRow__NameLink")]:<name=normalize-space(.)> ]
12            [ //...//span[contains
13            (@class, 'InfoColumn__Title')]:<bio=normalize-space(.)> ]
14            [ //...//span[contains(
15            @class, 'InfoColumn__Location')]:<location=normalize-space(.)> ]
16            [? //...//div[contains(@class,
17            'PriceAndFavoriteColumn__Price-')]:<price=normalize-space(.)> ]
18            [? //...//img[contains(@class,
19            'ImageColumn__DesktopImage')]:<profile_img=qualify-url(@src)> ]
20            [? //...//div[contains
21            (@class, 'InfoPills__ReviewsWrapper')]/span[contains(@class
22            , 'CalloutBadge__Badge')]:<nr_reviews=normalize-space(.)> ]
23
24            [? //...//div[contains(@class, 'StarRating__SvgWrapper
25            ')]:<rating=normalize-space(@aria-label)> ]
26
27            [? //.../
28            //...//div[contains(@class, 'HorizontalLayout__HorizontalWrapper

```

```
')]/a[contains(@class, 'InfoPills__CalloutBadgeAnchor  
')]//span[contains(@class, 'InfoPills__StyledCalloutBadge  
' )]:<repeat_pet_owner=normalize-space(.)> ]  
16         ]  
17     ]
```


Appendix B

Additional images and illustrations


```

{
  _id : https://boards.4chan.org/pol/thread/355970429
  value : {
    url : https://boards.4chan.org/pol/thread/355970429
    attributes : {
      archived :  true
      deleted :  false
      hidden :  false
      incomplete :  false
      inactive :  false
    }
    data : {
      op : {
        image : //i.4cdn.org/pol/1641909781132.png
        image_name : File: Schermata 2022-01-11 alle(...).png (2.83 MB, 2880x1348)
        subject : value
        author_name : Anonymous
        authorid : 3FoX9Gox
        date_time : 01/11/22(Tue)14:03:01
        post_number : 355970429
        message : Pack of 30ish arabs assaulted two italian girls on new year's eve in the central square of Milan , in front of the Police.https://www.ilfattoquotidiano.it/2022/01/07/milano-un-video-mostra-unaltra-aggressione-in-piazza-duomo-nella-notte-di-capodanno/6448789/
        direct_replies : [ 12 items ]
        flag : Italy
      }
      replies : [ 18 items ]
      0 : {
        reply : [ 1 item ]
        0 : {
          message : [ 1 item ]
          author_name : [ 1 item ]
          authorid : [ 1 item ]
          date_time : [ 1 item ]
          post_number : [ 1 item ]
        }
      }
    }
  }
}

```

Figure B.1: 4chan data structure

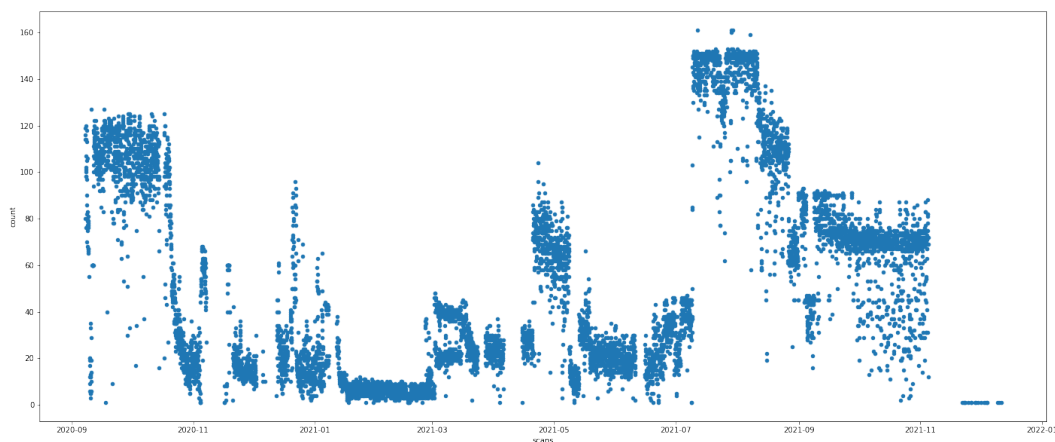


Figure B.2: Number of discovery instances by time

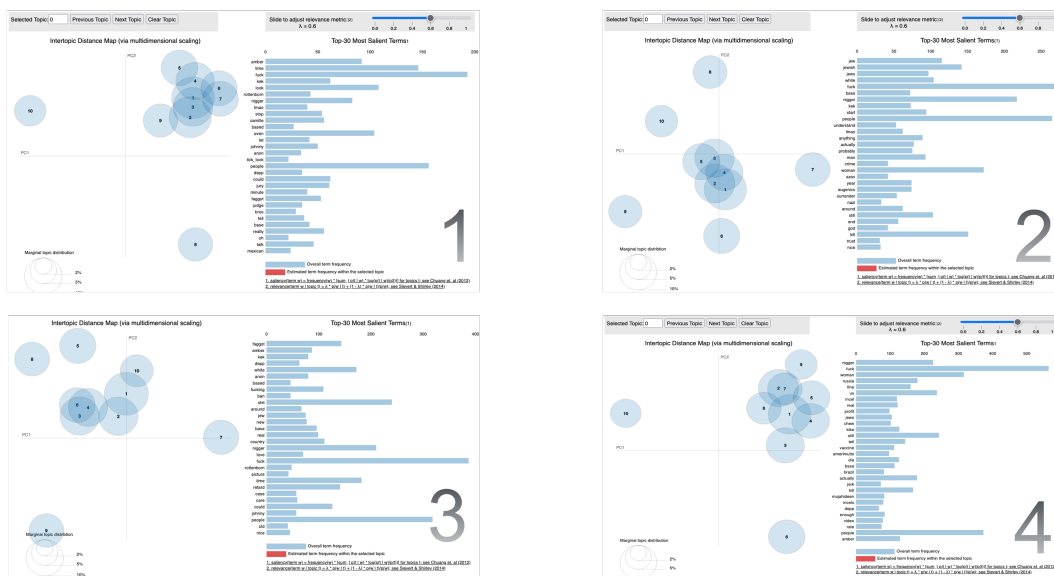
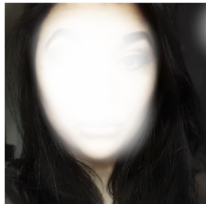



Figure B.3: Live board and t_{-1}, t_{-2}, t_{-3} boards topic models




Figure B.4: Live board and t_{-1}, t_{-2}, t_{-3} boards sentiment


Calendar Recently Updated




1. Zylia 

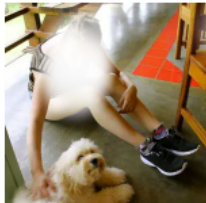
Grown up with German Shepherds, no dog is too much
London, HA9


1 repeat pet owner 7 reviews 

 "Zyla had a lovely time with Tish and her little puppy! I was sent regular updates and pictures of Zyla on her stay. Zyla was taken on a wonderful long walk and..." [\(more\)](#)


from **£30** per night 


Calendar Recently Updated




2. Sophie 


Fun, reliable and flexible pet care
London, NW6


13 reviews 

 "Sophie helped us out at the last minute watching our dog Whisky while we went to a wedding for the weekend. Whisky was very well looked after- we got lots of h..." [\(more\)](#)


from **£45** per night 


Calendar Recently Updated




3. Natasha 

Northern Irish dog lover!
London, SE1

2 repeat pet owners 7 reviews 

 "Natasha was great; I received regular updates and Otto was happy and well cared for. I have booked with Natasha again already for a future trip!" [\(more\)](#)

from **£35** per night 

with-german-shepherds-no-dog-is-too-much/?service_type=overnight-boarding&start_date=19%2F09%2F2022&end_date=25%2F09%2F

Figure B.5: Rover.com search result example

Appendix C

Code repositories

The main code repository for this research project can be found here: <https://bitbucket.org/yprifti/data-gather-open/>

- Folder *data-gather-open/jupyter* contains the Python Jupyter notebooks
- Folder *data-gather-open/integrations/api/app* contains third-party API integrations for complex discovery queries that depend on additional data and algorithms
- Folder *data-gather-open/k8s/dcs-k8s/k8s-deployments* contain the Kubernetes cluster configurations, deployments, and helm charts for deploying and running data collection campaigns
- Folder *data-gather-open/mongo* contain MongoDB queries for data exploration, map-reduce and data cleansing
- Folder *data-gather-open/projects* contains data campaign web extraction XPath and DR-Web-Engine queries
- Folder *data-gather-open/py* contains the main python code for the OSNNs-ScrA web data extraction distributed engine
- File *data-gather-open/py/main.py* contains the CLI for running data extraction campaigns

As part of this project, a running docker container was created. The container is a single running instance of the extraction engine. The repository for the docker container can be found here: <https://bitbucket.org/yprifti/data-gather-docker>.

- There are three docker files in this repository. The File *Dockerfile* contains the working copy. The file *Dockerfile-Pub* contains a slim version containing only the runnable engine. The file *Dockerfile-Dev* contains a debuggable version of the engine, able to lunch X-Server into the host machine; hence allowing to observe the execution of the extraction query.
- The file *builder* offers a CLI for building the docker image.
- The file *runner* offers a CLI for running local instances of the container
- The files' *id_rsa_docker** contain the RSA public and private keys for access to the code repository. These are used within the container to allow code access. All instances will pull the latest version of the code before execution, allowing hot rolling and propagation of code changes into the running distributed engine.
- A version of the public build container can be found on the Docker Image Repository: <https://hub.docker.com/repository/docker/ylliprifti/yp-phd-pub>

Bibliography

- [1] Ylli Prifti, Iacopo Pozzana, and Alessandro Provetti. Live monitoring 4chan discussion threads. In *7th Int'l Conference on Computational Social Science*, 2021.
- [2] Y. Prifti I. Pozzana and A. Provetti. On-line page scraping reveals evidence of moderation in 4chan/pol/ anonymous discussion threads. In *Proc. of 3rd European Symposium on Societal Challenges in Computational Social Science*. ETH Press, 2019.
- [3] Y. Prifti P. De Meo, I. Pozzana and A. Provetti. The dynamics of recommendation in high-trust personal care services. In *5th Int'l Conference on Computational Social Science (IC2S2)*, 2019.
- [4] Y. Prifti P. De Meo, I. Pozzana and A. Provetti. Finding gender bias in web-based, high-trust interactions. In *Proc. of 2nd European Symposium on Societal Challenges in Computational Social Science, GeWISS reports*, 2018.
- [5] Y. Prifti P. De Meo, I. Pozzana and A. Provetti. Gender bias in web-based, high-trust interactions. In *5th Int'l Conference on Computational Social Science (IC2S2)*, 2019.
- [6] Alessandro Provetti Iacopo Pozzana, Ylli Prifti and Anders Seyersted Sandbu. Mapping the norwegian 4chan: How conspiracy theories travel the language barriers. In *7th Int'l Conference on Computational Social Science (IC2S2)*, 2021.
- [7] Ylli Prifti. 4chan /pol board as a temporary evolution of live threads and posts., July 2021.
- [8] Paschalis Lagias, George D. Magoulas, Ylli Prifti, and Alessandro Provetti. Predicting seriousness of injury in a traffic accident: A new imbalanced dataset

- and benchmark. In Lazaros Iliadis, Chrisina Jayne, Anastasios Tefas, and Elias Pimenidis, editors, *Engineering Applications of Neural Networks - 23rd International Conference, EAAAI/EANN 2022, Chersonissos, Crete, Greece, June 17-20, 2022, Proceedings*, volume 1600 of *Communications in Computer and Information Science*, pages 412–423. Springer, 2022.
- [9] Andrea Ballatore, A. Pang, Iacopo Pozzana, Ylli Prifti, and Alessandro Proveti. Geo-referencing as a connector between user reviews and urban environment quality. In *5th Int'l Conference on Computational Social Science*, 2019.
- [10] Trudy Govier. *Social trust and human communities*. McGill-Queen's Press-MQUP, 1997.
- [11] Castelfranchi Cristiano, Rino Falcone, and Lorini Emiliano. A Non-reductionist Approach to Trust. In Jennifer Golbeck, editor, *Computing with Social Trust*, pages 45–72. Springer London, London, 2009.
- [12] Alejandro Portes. Social capital: Its origins and applications in modern sociology. *Annual review of sociology*, 24(1):1–24, 1998.
- [13] Pierre Bourdieu. Le capital social: notes provisoires. *Actes de la recherche en sciences sociales*, 31(1):2–3, 1980.
- [14] Robert Leonardi, Raffaella Y Nanetti, and Robert D Putnam. *Making democracy work: Civic traditions in modern Italy*. Princeton university press Princeton, NJ, 2001.
- [15] Denise M Rousseau, Sim B Sitkin, Ronald S Burt, and Colin Camerer. Not so different after all: A cross-discipline view of trust. *Academy of management review*, 23(3):393–404, 1998.
- [16] Diego Gambetta. *Trust: Making and Breaking Cooperative Relations*. Blackwell, 1988.
- [17] Stephen Paul Marsh. Formalising trust as a computational concept. Technical report, University of Stirling, 1994.

- [18] Stephen Marsh and Pamela Briggs. Examining Trust, Forgiveness and Regret as Computational Concepts. In *Computing with Social Trust, Human-Computer Interaction Series*, page 9. Empty, 2009.
- [19] Rino Falcone and Cristiano Castelfranchi. Social trust: A cognitive approach. In *Trust and deception in virtual societies*, pages 55–90. Springer, 2001.
- [20] Wanita Sherchan, Surya Nepal, and Cecile Paris. A survey of trust in social networks. *ACM Computing Surveys (CSUR)*, 45(4):1–33, 2013.
- [21] Raph Levien. Attack-Resistant Trust Metrics. In *Computing with Social Trust, Human-Computer Interaction Series*, page 121. Empty, 2009.
- [22] Cai-Nicolas Ziegler. On Propagating Interpersonal Trust in Social Networks. In Jennifer Golbeck, editor, *Computing with Social Trust*, pages 133–168. Springer London, London, 2009.
- [23] Jennifer Golbeck. *Computing with social trust*. Springer Science & Business Media, 2008.
- [24] Jennifer Golbeck. *Introduction to Computing with Social Trust*. Springer, London, UK, 2009.
- [25] Jennifer Golbeck and Ugur Kuter. The ripple effect: change in trust and its impact over a social network. In *Computing with Social Trust*, pages 169–181. Springer, 2009.
- [26] Yu Zhang, Huajun Chen, and Zhaohui Wu. A social network-based trust model for the semantic web. In *International Conference on Autonomic and Trusted Computing*, pages 183–192. Springer, 2006.
- [27] Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, 11 1999.

- [28] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651, 2003.
- [29] Jennifer Golbeck, James A Hendler, et al. Filmtrust: movie recommendations using trust in web-based social networks. In *CCNC*, volume 2006, pages 282–286. Citeseer, 2006.
- [30] Yousra Asim, Ahmad Kamran Malik, Basit Raza, and Ahmad Raza Shahid. A trust model for analysis of trust, influence and their relationship in social network communities. *Telematics and Informatics*, 36:94–116, 2019.
- [31] John R Douceur. The Sybil Attack. In Druschel Peter and Kaashoek, Frans and Rowstron Antony, editor, *Peer-to-Peer Systems*, pages 251–260, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
- [32] Majed AlRubaian, Muhammad Al-Qurishi, Sk Md Mizanur Rahman, and Atif Alamri. A novel prevention mechanism for Sybil attack in online social network. In *2015 2nd World Symposium on Web Applications and Networking (WSWAN)*, pages 1–6. IEEE, 2015.
- [33] Deborah Bunker. Who do you trust? the digital destruction of shared situational awareness and the covid-19 infodemic. *International Journal of Information Management*, 55:102201, 2020.
- [34] Cristiano Codagnone and Bertin Martens. Scoping the sharing economy: Origins, definitions, impact and regulatory issues. *SSRN Electronic Journal*, 2016.
- [35] Shankar Iyer, Justin Cheng, Nick Brown, and Xiuhua Wang. When does trust in online social groups grow? In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 283–293, 2020.
- [36] Carol Moser, Paul Resnick, and Sarita Schoenebeck. Community commerce: Facilitating trust in mom-to-mom sale groups on facebook. In *Proceedings*

of the 2017 CHI Conference on Human Factors in Computing Systems, pages 4344–4357, 2017.

- [37] Shaozhong Zhang and Haidong Zhong. Mining users trust from e-commerce reviews based on sentiment similarity analysis. *IEEE Access*, 7:13523–13535, 2019.
- [38] Sonja Grabner-Kräuter and Ewald A Kaluscha. Empirical research in on-line trust: a review and critical assessment. *International journal of human-computer studies*, 58(6):783–812, 2003.
- [39] D Harrison McKnight, Vivek Choudhury, and Charles Kacmar. Developing and validating trust measures for e-commerce: An integrative typology. *Information systems research*, 13(3):334–359, 2002.
- [40] Markus Schläpfer, Lei Dong, Kevin O’Keeffe, Paolo Santi, Michael Szell, Hadrien Salat, Samuel Ankesaria, Mohammad Vazifeh, Carlo Ratti, and Geoffrey B West. The universal visitation law of human mobility. *Nature*, 593(7860):522–527, 2021.
- [41] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000, 2010.
- [42] Evelien Otte and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of information Science*, 28(6):441–453, 2002.
- [43] Pasquale De Meo, Katarzyna Musial-Gabrys, Domenico Rosaci, Giuseppe ML Sarne, and Lora Aroyo. Using centrality measures to predict helpfulness-based reputation in trust networks. *ACM Transactions on Internet Technology (TOIT)*, 17(1):1–20, 2017.
- [44] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology*, 2(1):113–120, 1972.
- [45] Ruchi Mittal and MPS Bhatia. Characterizing the properties of the multiplex social networks. In *Proceedings of the INDIACOM*, page 5, 2018.

- [46] Davide Vega, Roc Meseguer, and Felix Freitag. Analysis of the social effort in multiplex participatory networks. In *International Conference on Grid Economics and Business Models*, pages 67–79. Springer, 2014.
- [47] Desislava Hristova, Anastasios Noulas, Chloë Brown, Mirco Musolesi, and Cecilia Mascolo. A multilayer approach to multiplexity and link prediction in online geo-social networks. *EPJ Data Science*, 5(1):24, 2016.
- [48] Minttu Laukkanen and Nina Tura. The potential of sharing economy business models for sustainable value creation. *Journal of Cleaner Production*, 253:120004, 2020.
- [49] Aurélien Acquier, Thibault Daudigeos, and Jonatan Pinkse. Promises and paradoxes of the sharing economy: An organizing framework. *Technological Forecasting and Social Change*, 125:1–10, 2017.
- [50] Sandeep Kayastha. Defining service and non-service exchanges. *Service Science*, 3(4):313–324, 2011.
- [51] Sarah Netter, Esben Rahbek Gjerdrum Pedersen, and Florian Lüdeke-Freund. Sharing economy revisited: Towards a new framework for understanding sharing models. *Journal of Cleaner Production*, 221:224–233, 2019.
- [52] John Stark. Coherent vision, strategy, plan, resources, metrics. *Product Lifecycle Management: 21st Century Paradigm for Product Realisation*, pages 129–133, 2005.
- [53] T I M BERNERS-LEE, JAMES HENDLER, and O R A LASSILA. THE SEMANTIC WEB. *Scientific American*, 284(5):34–43, 2001.
- [54] Davide Marengo, Danny Azucar, Claudio Longobardi, and Michele Settanni. Mining Facebook data for Quality of Life assessment. *Behaviour & Information Technology*, 0(0):1–11, 2020.
- [55] Stine Lomborg and Anja Bechmann. Using APIs for data collection on social media. *The Information Society*, 30(4):256–265, 2014.

- [56] Anne Oeldorf-Hirsch and Darren Gergle. “Who Knows What”: Audience Targeting for Question Asking on Facebook. *Proc. ACM Hum.-Comput. Interact.*, 4(GROUP), 1 2020.
- [57] Mai Monica and Leung, Carson K. and Choi Justin M C. and Kwan Long Kei Ronnie. Big Data Analytics of Twitter Data and Its Application for Physician Assistants. In Alhadjj Reda and Moshirpour, Mohammad and Far Behrouz, editor, *Data Management and Analysis: Case Studies in Education, Healthcare and Beyond*, pages 17–32. Springer International Publishing, Cham, 2020.
- [58] Jürgen Pfeffer, Katja Mayer, and Fred Morstatter. Tampering with twitter’s sample api. *EPJ Data Science*, 7(1):50, 2018.
- [59] M Gjoka, M Kurant, C T Butts, and A Markopoulou. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, 2010.
- [60] Salvatore A Catanese, Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Crawling Facebook for Social Network Analysis Purposes. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics, WIMS ’11*, New York, NY, USA, 2011. Association for Computing Machinery.
- [61] A Trifa, A H Sbaï, and W L Chaari. Evaluate a Personalized Multi Agent System through Social Networks: Web Scraping. In *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 18–20, 2017.
- [62] De S Sirisuriya et al. A comparative study on web scraping. *Empty*, 2015.
- [63] Rabiyatou Diouf, Edouard Ngor Sarr, Ousmane Sall, Babiga Birregah, Mamadou Bousso, and Seny Ndiaye Mbaye. Web Scraping: State-of-the-Art and Areas of Application. In *2019 IEEE International Conference on Big Data (Big Data)*, 2019 IEEE International Conference on Big Data (Big Data), pages 6040–6042, Los Angeles, United States, 12 2019. IEEE.

- [64] Michael K. Bergman. White paper: The deep web: Surfacing hidden value. *The Journal of Electronic Publishing*, 7(1), August 2001.
- [65] Tim Furche, Georg Gottlob, Giovanni Grasso, Christian Schallhart, and Andrew Sellers. Oxpath: A language for scalable data extraction, automation, and crawling on the deep web. *The VLDB Journal*, 22(1):47–72, 2013.
- [66] Ram Sharan Chaulagain, Santosh Pandey, Sadhu Ram Basnet, and Subarna Shakya. Cloud based web scraping for big data applications. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 138–143. IEEE, 2017.
- [67] Aldo Hernandez-Suarez, Gabriel Sanchez-Perez, Karina Toscano-Medina, Victor Martinez-Hernandez, Victor Sanchez, and Héctor Perez-Meana. A web scraping methodology for bypassing twitter api restrictions. *arXiv preprint arXiv:1803.09875*, 2018.
- [68] J. Clement. Facebook users worldwide 2019, Jan 2020.
- [69] Christopher Olston, Marc Najork, et al. Web crawling. *Foundations and Trends® in Information Retrieval*, 4(3):175–246, 2010.
- [70] Eloisa Vargiu and Mirko Urru. Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artif. Intell. Research*, 2(1):44–54, 2013.
- [71] Carolyn J Hirsch, Jack L Hirsch, and Carolyn J Hirsch. *SQL, the Structured Query Language*. Tab Books, 1988.
- [72] Ellen Spertus and Lynn Andrea Stein. Squeal: a structured query language for the web. *Computer Networks*, 33(1-6):95–103, 2000.
- [73] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.
- [74] Ora Lassila, Ralph R Swick, et al. Resource description framework (rdf) model and syntax specification. ””, 1998.

- [75] Brian McBride. Four steps towards the widespread adoption of a semantic web. In *International Semantic Web Conference*, pages 419–422. Springer, 2002.
- [76] Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan, and Firman Firdaus. Comparison of web scraping techniques : Regular expression, html dom and xpath. In *2018 International Conference on Industrial Enterprise and System Engineering (ICoIESE 2018)*. Atlantis Press, 2019/03.
- [77] James Clark, Steve DeRose, et al. Xml path language (xpath), 1999.
- [78] Andrzej Ehrenfeucht and Paul Zeiger. Complexity measures for regular expressions. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, STOC '74*, page 75–79, New York, NY, USA, 1974. Association for Computing Machinery.
- [79] Andrew Jon Sellers, Tim Furche, Georg Gottlob, Giovanni Grasso, and Christian Schallhart. OXPath. In *Proceedings of the 20th international conference companion on World wide web - WWW '11*. ACM Press, 2011.
- [80] Tim Furche, Georg Gottlob, Giovanni Grasso, Xiaonan Guo, Giorgio Orsi, Christian Schallhart, and Cheng Wang. Diadem: thousands of websites to a single database. *Proceedings of the VLDB Endowment*, 7(14):1845–1856, 2014.
- [81] Jinsong Guo, Valter Crescenzi, Tim Furche, Giovanni Grasso, and Georg Gottlob. Red: Redundancy-driven data extraction from result pages? In *The World Wide Web Conference*, pages 605–615, 2019.
- [82] Ruslan R Fayzrakhmanov, Emanuel Sallinger, Ben Spencer, Tim Furche, and Georg Gottlob. Browserless web data extraction: challenges and opportunities. In *Proceedings of the 2018 World Wide Web Conference*, pages 1095–1104, 2018.
- [83] Olaf Hartig and Jorge Pérez. Semantics and complexity of graphql. In *Proceedings of the 2018 World Wide Web Conference*, pages 1155–1164, 2018.

- [84] KB Sundhara Kumar, S Mohanavalli, et al. A performance comparison of document oriented nosql databases. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pages 1–6. IEEE, 2017.
- [85] Niteshwar Datt Bhardwaj. Comparative study of couchdb and mongodb–nosql document oriented databases. *International Journal of Computer Applications*, 136(3):24–26, 2016.
- [86] David Bernstein. Containers and cloud: From lxc to docker to kubernetes. *IEEE Cloud Computing*, 1(3):81–84, 2014.
- [87] Zhiheng Zhong and Rajkumar Buyya. A cost-efficient container orchestration strategy in kubernetes-based cloud computing infrastructures with heterogeneous resources. *ACM Transactions on Internet Technology (TOIT)*, 20(2):1–24, 2020.
- [88] Emilija Jokubauskaitė and Stijn Peeters. Generally curious: Thematically distinct datasets of general threads on 4chan/pol. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 863–867, 2020.
- [89] Thomas Colley and Martin Moore. The challenges of studying 4chan and the alt-right. *New Media & Society*, 0(0):1461444820948803, 0.
- [90] Antonis Pappasavva, Savvas Zannettou, Emiliano De Cristofaro, Gianluca Stringhini, and Jeremy Blackburn. Raiders of the lost kek: 3.5 years of augmented 4chan posts from the politically incorrect board. *CoRR*, abs/2001.07487, 2020.
- [91] Gabriel Emile Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. A longitudinal measurement study of 4chan’s politically incorrect forum and its effect on the web. *CoRR*, abs/1610.03452, 2016.
- [92] Jay Shah and Dushyant Dubaria. Building modern clouds: using docker, kubernetes & google cloud platform. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0184–0189. IEEE, 2019.

- [93] Gabriel Emile Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. Kek, cucks, and god emperor trump: A measurement study of 4chan's politically incorrect forum and its effects on the web. In *Eleventh International AAI Conference on Web and Social Media*, 2017.
- [94] Antonis Papasavva, Savvas Zannettou, Emiliano De Cristofaro, Gianluca Stringhini, and Jeremy Blackburn. Raiders of the lost kek: 3.5 years of augmented 4chan posts from the politically incorrect board. In *Proceedings of the International AAI Conference on Web and Social Media*, volume 14, pages 885–894, 2020.
- [95] Chat Room. Levenshtein distance. *algorithms*, 12(14):32, 2019.
- [96] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.
- [97] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.
- [98] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [99] Guido Barbian. Trust centrality in online social networks. In *2011 European Intelligence and Security Informatics Conference*, pages 372–377. IEEE, 2011.
- [100] David A Askay. Silence in the crowd: The spiral of silence contributing to the positive bias of opinions in an online review system. *New Media & Society*, 17(11):1811–1829, 2015.
- [101] Justin J Miller. Graph database applications and concepts with neo4j. In *Proceedings of the southern association for information systems conference, Atlanta, GA, USA*, volume 2324, 2013.
- [102] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

- [103] Cristiano Castelfranchi and Rino Falcone. Trust is much more than subjective probability: Mental components and sources of trust. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 10–pp. IEEE, 2000.
- [104] Sefik Ilkin Serengil and Alper Ozpinar. Hyperextended lightface: A facial attribute analysis framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4. IEEE, 2021.
- [105] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.
- [106] Philip Sedgwick. Pearson’s correlation coefficient. *Bmj*, 345, 2012.
- [107] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [108] Mathieu Blondel, Masakazu Ishihata, Akinori Fujino, and Naonori Ueda. Polynomial networks and factorization machines: New insights and efficient training algorithms. In *International Conference on Machine Learning*, pages 850–858. PMLR, 2016.