TÉCNICO LISBOA

Virus Propagation On Social Networks Using Large-scale Biosensors

João Miguel Gonçalves de Sousa Andrade

Thesis to obtain the Master of Science Degree in

Communication Networks Engineering

Examination Committee

Chairperson: Prof. Paulo Jorge Pires Ferreira
Supervisor: Prof. Artur Miguel do Amaral Arsénio
Member of the Committee: Prof. Pedro Miguel Martins Encarnação

October 2013

Acknowledgments

Part of this work took form under the Harvard-Portugal Collaborative Research Award HMSP-CT/SAU-ICT/0064/2009: Improving perinatal decision-making: development of complexity-based dynamical measures and novel acquisition systems.

Abstract

The recent technological developments on mobile technologies allied with the growing computational capabilities of sensing enabled devices have given rise to mobile sensing systems that can target community level problems. These systems are capable of inferring intelligence from acquired raw sensed data, through the use of data analysis techniques. However, due to their recent advent, associated issues remain to be solved in a systematized way. Various areas can benefit from these initiatives, with public health systems having a major applicational gain. There has been interest in the use of social networks as a mean of epidemic prediction. Still, the integration between the mobile infrastructure and these initiatives, required to achieve epidemic prediction, is yet to be achieved. In this context, a system applied to epidemic prediction is proposed and evaluated.

Keywords

Pervasive computing; epidemic prediction; large-scale sensing; social network analysis

Resumo

Os recentes desenvolvimentos tecnológicos nas comunicações móveis, juntamente com a crescente capacidade computacional de dispositivos capazes de actuar como sensores, deram origem a sistemas sensoriais móveis que se podem focar nos problemas de uma comunidade. Estes sistemas são capazes de inferir inteligência de informação adquirida de sensores, através de técnicas de análise de dados. No entanto, devido ao seu recente aparecimento, ainda existem problemas que lhes estão associados e que devem ser resolvidos de uma forma sistemática. Várias áreas poderão beneficiar destas iniciativas, com os sistemas públicos de saúde a terem um maior ganho potencial. Existe interesse no uso de redes sociais como meios para a predição de epidemias. Presentemente, a integração entre infraestruturas móveis e estas iniciativas, necessária para a predição epidémica, ainda não foi alcançada. Neste contexto, um sistema aplicado à predição de epidemias é proposto e avaliado.

Palavras Chave

Computação pervasiva; predição epidémica; sensores de larga escala; análise de redes sociais

Contents

List of Figures

List of Tables

List of Acronyms

ABM Agent-based Model

AIDS Acquired Immunodeficiency Syndrome

API Abstract Programming Interface

Ajax Asynchronous Javascript and XML

CA Certification Authority

CDC Centers for Disease Control

CI Confidence Interval

CORS Cross-Origin Resource Sharing

CSS Cascading Style Sheets

D3 Data-Driven Documents

DES Discrete Event System

DOM Document Object Model

FSM Finite State Machine

ggplot2 Grammar of Graphics Plot 2

GPRS General Packet Radio Service

GPS Global Positioning System

HMAC Hash Message Authentication Code

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IDE Integrated Development Environment

IEEE Institute of Electrical and Electronics Engineers

IP Internet Protocol

JSON Javascript Object Notation

JVM Java Virtual Machine

MVVM Model-View-ViewModel

NPM Node Package Manager

OS Operating System

PBKDF2 Password-Based Key Derivation Function 2

PTTS Probabilistic Timed Transition System

PKI Public Key Infrastructure

REST Representational State Transfer

SCI Social Community Intelligence

SDK Software Development Kit

SEM Standard Error of the Mean

SHA Secure Hash Function

SIR Susceptible-Infected-Recovered

SJCL Stanford's Javascript Cryptographic Library

SPA Single Page Architecture

TCP Transmission Control Protocol

TLS Transport Layer Security

UI User Interface

URL Uniform Resource Locator

WHO World Health Organization

1

Introduction

1.1 Background Information

Distributed systems have been used as a platform to allow the interaction between groups of individuals and a set of devices. As technology advances in sensing, computation, storage and communications become widespread, ubiquitous sensing devices will become a part of global distributed sensing systems [1] [2].

Recently, the predominance of mobile phones equipped with sensors, the explosion in social networks and the deployment of sensor networks have created an enormous digital footprint that can be harnessed [3]. Furthermore, developments in sensor technology, communications and semantic processing, allow the coordination of a large network of devices and large dataset processing with intelligent data analysis [1].

The sensing of people constitutes a new application domain that broadens the traditional sensor network scope of environmental and infrastructure monitoring. People become the carriers of sensing devices and both producers and consumers of events [4]. As a consequence, the recent interest by the industry in open programming platforms and software distribution channels is accelerating the development of people-centric sensing applications and systems [4] [1].

To take advantage of these emerging networks of mobile people-centric sensing devices, researchers arrived at the concept of *Mobiscopes*, i.e. taskable mobile sensing systems that are capable of high coverage. They represent a new type of infrastructure, where mobile sensors have the potential to logical belong to more than one network, while being physically attached to their carriers [5]. By taking advantage of these systems, it will be possible to mine and run computations on enormous amounts of data from a very large number of users [1].

A people-centric sensing system imbues the individuals it serves in a symbiotic relationship with itself [6] [2]. People-centric sensing enables a different approach to sensing, learning, visualizing and data sharing, not only self-centred, but focused on the surrounding world. The traditional view on mesh sensor networks is combined with one where people, carrying sensors turn opportunistic coverage into a reality [2]. These sensors can reach into regions, static sensors cannot, proving to be especially useful for applications that occasionally require sensing [5]. By employing these systems, one can aim to revolutionize the field of context-aware computing [3].

An alternative of a world-wide coverage of static sensors to develop people-centric systems is unfeasible in terms of monetary costs, management and permissions [6] [2]. Also, it is extremely challenging in static sensing models, due to band limits and issues that arise from covering a vast area, to satisfy the required density requirements [5]. Thanks to their mobility, mobile sensors overcome spatial coverage limitations [5] [6].

Adoption issues might come up as potential users are usually unaware of the benefits that arise from technological developments. However, with the advent of smartphones, a direct impact in daily life is easier to achieve, making advantages clearer. By using opportunistic sensing, functionality can be offered in a transparent fashion [1], leaving the user agnostic of system activity and circumventing adoption obstacles that might be present in participatory sensing.

Behavioural modelling requires large amounts of accurate data [7]. These systems constitute an opportunity for intelligent analysis systems, as relevant information can be obtained from large-scale sensory data and employed in statistical models [7] [1]. Great benefits can be taken from this unconstrained human data, in opposition to the traditional carefully setup experiments [7]. With these developments it is now possible to distribute and run experiments in a world-wide population rather than in a small laboratory controlled study [1].

By leveraging the behavioural patterns related to individuals, groups and society, a new multi-disciplinary field is created: Social Community Intelligence (SCI) [3]. Real-time user contributed data is invaluable to address community-level problems and provide an universal access to information, contributing to the emergence of innovative services [3] [2] [1]. For instance, the prediction and tracking of epidemic outbreaks across populations [3]. Thus, technological benefits are shifted from a restricted group of scientists to the whole society [2].

Healthcare is a possible application, where these systems can facilitate monitoring and sharing of automatically gathered health data [2]. Epidemics are a major public health concern and it has been shown impact can be reduced by early detection of the disease activity. For instance, it has been shown that the level of influenza-like illness in regions of the US can be estimated with a reporting lag of one day, when compared to clinical methods whose results take a week to be published [3].

The advent of ubiquitous networks of mobile sensing devices constitute a paradigm shift, offering researchers challenges in network architecture, protocol design and data abstractions [5].

Results from mobile sensing networks, pervasive computing and methods of statistical analysis can be exploited, however, new unaddressed challenges arise. These challenges range from growing volumes of multi-modal sensor data, dynamic operating conditions and the increasing mobility of the sensing devices [1].

As most people will, in the near future, possess sensing-enabled phones, the main obstacle in this area is not the lack of an infrastructure. Rather, the technical barriers are related to performing privacy and resource respecting analysis, while supplying users and communities with useful feedback [1].

The main challenges in this area are as follows:

- Managing user participation in the system (participatory versus opportunistic sensing) [1];
- Managing trust in users as not to compromise the whole system [1];
- Adapting to a device's changing resource availability and sampling context [2] [1];
- Dealing with mobile sensing devices resource restrictions [1];
- Coping with sensing device mobility (e.g. when there aren't enough mobile sensors or the sensor is moving and jeopardizing sampling for a given context) [6] [2];
- Enabling devices to share sensory data while protecting user privacy [6] [1];
- Relating and optimizing diverse resources and application-relevant metrics to define data collection and dissemination methods [1] [5];
- Managing large amounts of generated data [1];
- Defining an approach to collect data capable of assessing the accuracy of algorithms that interpret sensor data [1];
- Performing robust and accurate data analysis in a dynamic real-time environment [1] [5];
- Providing the user's with useful information feedback [1];
- Sensing system scaling from a personal to a population scale [1]

Enabling user participation in the system, while minimizing mobile sensing devices resource consumption and protecting user privacy are addressed in the proposed solution. This solution considers performing robust data analysis in a dynamic environment and system scaling from a personal to a community-level, while providing useful feedback to its users.

The chosen applicational area of epidemic prediction has an inherent lack of adequate data that does not result from potentially biased simulations [8]. To counter this, similarities between computer and biological infectious agents are exploited [9].

1.2 Objectives

In an infectious disease it is necessary to detect, monitor and foresee the advent of an epidemic in a real-time environment. To operate in such a scenario the system should know who can get infected and which people have been in contact and where. Contact location, time and relationship with the subject are relevant metrics that affect the probability of disease propagation. Sensors and social networks analysis allow the integration of these concerns into personal devices, while developments in data analysis and modelling allow more accurate results regarding this data, potentially indicating community-level susceptibility to an epidemic.

This work comprises data gathering and management, intelligent analysis and privacy respecting pervasive computing applied to epidemiological disease prediction in a population. A community is the target population of the analysis. It consists of the set of sensing devices belonging to people that are users plus their associated social contact network.

Sampling is only possible when privacy requirements are met.

Participation in the system is managed in an opportunistic fashion.

The developed system is capable of exploiting large sets of multimodal sensorial data (contact and network data).

Information extraction from raw data is based on the application of data analysis strategies in a dynamic environment.

These operations occur with some degree of distribution in the mobile sensing device and data processing backend. The criteria for this is based upon privacy, communication and resource management concerns.

There are other solutions targeting these problems, but none of them has been applied to epidemic prediction, as it is the case with this solution.

Information is fed to an epidemic model-based algorithm, predicting the possibility of an epidemic and notifying users about the risk.

1.3 Main Contributions

The development of a flexible and configurable opportunistic real-time communication system capable of evaluating and predicting the outbreak of an epidemic in a community, while guaranteeing privacy, exploring social networks as data and implementing predictive algorithms.

This goal is achieved by the correlation of data that bears distinct viewpoints and resolutions, while building an effective data merge and processing algorithm that is capable of extracting higher-intelligence or information.

Its area of application is epidemic prediction, by applying data analysis methods to the large-scale network data sourced from users and their sensed contacts.

Intelligence is gathered in near real-time from a large-scale sensing network, in which only a

population sample is considered.

This technology enables the extraction of high-level information regarding epidemic outbreak, effectively predicting the possibility of a community-level epidemic, while minimizing the resource load on the sensing devices.

This work resulted in the papers: Social Web for Large-Scale Biosensors [10], Internet of Intelligent Things: Bringing Artificial Intelligence into Things and Communication Networks [11] and Epidemic Spreading Over Social Networks Using Large-scale Biosensors: A Survey [12].

1.4 Dissertation Outline

This chapter introduced the main problems faced, the objectives of this work and its contributions.

Chapter 2 describes the related work, addressing the three main areas that constitute the multidisciplinary field where the contribution of this work is inserted, namely *Pervasive Computing*, *Computational Epidemiology* and *Social Network Analysis*. A section approaching *Development* issues and requirements is also included.

The architecture for the solution is presented in Chapter 3, referring the different parts of the system while abstracting contextual details.

Chapter 4 clarifies the architectural implementation process in the rationale behind its methodology criteria and requirement-oriented decisions.

Chapter 5 provides a basis for the assessment methods, while presenting the experimental results of this solution.

In the last chapter, final remarks are made, along with a contextualised summary of the contributions. Finally, future work is proposed.

2

Related Work

2.1 Pervasive Computing

2.1.1 Range

There is a tendency to augment devices with sensing, computing and communication functionalities, connecting them together to form a network, and make use of their collective capabilities [3]. This sensing network is made possible across various ranges, including: single individuals, groups with common interests or the entire population of a city [1].

Users become a key system component, enabling a variety of new application areas such as *personal*, *social* and *community* sensing. Each of these scenarios has its own challenges on how to understand, visualize and share data with others [2].

In *personal* sensing, the focus is on monitoring the individual [2]. In these applications, information is generated for the sole consumption of the user and is generally not shared with others [1].

In *social* sensing, information is shared within the group [2]. Individuals who participate in these applications have a commonality of interests, constituting a group [1].

In *community* sensing, data is shared for the greater good. This area is inserted into SCI, where the scope of sensing encompasses the whole community. Considering data source origin, SCI has its source in three fast-growing research fields: *mobile sensor-based activity recognition*, *context inference in smart spaces* and *social network analysis*.

The key idea behind *mobile sensor-based activity recognition* is to after a series of observations acquire the mathematical model behind human activities. It takes advantage of the prevalence of

sensors that accompany users in their mobility patterns.

Context inference in smart spaces relies on already deployed infrastructures of static sensors. Static sensors allow the detection of activities, enabling space context.

Social network analysis has been studied by physicists and social scientists for a couple of decades and is a major source of information and relationships among a group of individuals.

Aggregation of data from these sources constitutes an opportunity for the extraction of intelligence in a community. Applications only become useful once they have a large enough number of individuals participating. The growing diversity in means of communication, provides a platform for a large amount of data to be available. An infrastructure capable of integrating heterogeneous data sources, combining the resulting multimodal data and extracting behavioural patterns from it, through intelligent data analysis methods is required [3].

2.1.2 Participation

The need exists to define an individual's role in the sensing system. Two modalities are considered: *participatory* and *opportunistic* [3].

In *participatory* sensing, individuals are incorporated in the decision making process over the sensed data [3]. They can decide which data to share, enjoying control over data privacy issues. In this approach, the target is restricted to a group of users willing to participate in the system [3]. As a consequence, a *participatory* sensing application should have community appeal [2]. Some systems using this modality are [13] and [14].

In *opportunistic sensing*, a system automatically takes advantage a device's resources whenever its state (e.g. location or user activity) matches the context requirements of the application [3]. *Opportunistic sensing* becomes possible by the system's ability to modify its state in response to a dynamic environment [1]. Sampling only occurs if requirements are met and it is fully automated, with individuals having no involvement in the data collection process [2]. A result of this is that the decision burden is shifted away from users and moved into the system, resulting in more resources being demanded in this decision-making process [3] [2]. This heavier resource demand should not noticeably impact the normal usage experience of the sensing devices [2]. This issue can be tackled if opportunistic sensing is considered a secondary, low-priority operation on the sensing devices [2]. Nonetheless, as devices might only be able to meet sensing requirements for short and intermittent periods, a trade-off between availability and resource management should be considered [2]. One system using this approach is [15]. Also, in [16] one application is proposed and in [2] another application is referred.

2.1.3 Context

Context affects data sensing, while sensing devices with mobility can be used in unpredictable ways [1] [16]. Context is the metadata that describes the conditions to which sensors are exposed,

affecting both data and their ability to perform sensing operations. In opportunistic sensing, context contributes to the evaluation of potential sensor candidates, indicating when sampling should be started and stopped [2].

Context is important for analysing sampled data, especially when samples might be taken under suboptimal conditions [2]. In these environments, statistical models may fail to generalize. Also, sensors may be exposed to events for a too short time period, i.e. if the user is travelling too quickly or the sensor's sampling rate is too low [1]. Possible solutions are sharing sensors of available neighbouring devices temporarily if they are best-suited to sense events [2] [1]. Devices exchange context information and data is selected from the device whose context most closely matches application requirements. A mobile sensor detects the event target using its sensors and forwards the task to its better-suited neighbours. To recover a lost event source, the area to which the source is predicted to be in is estimated and the task is forwarded to sensors in the predicted area. Another approach is to use super-sampling, where data from nearby sensors is collectively used to lower the noise in an individual reading [1]. One challenge is determining a metric for context matching that, when used in these mechanisms, provides samples with enough fidelity to respect application requirements [2].

The reliability of intelligent algorithms may decrease under the dynamic and unexpected conditions presented by mobile sensor use (e.g. different individuals execute the same activity differently). These problems can be overcome by gathering sufficient samples of the different usage scenarios, i.e. training data [1]. However, acquiring training data is costly and anticipating the different scenarios that might be encountered is not possible for all applications [1], compromising the scalability of large-scale learning models [16]. Existing solutions are based on borrowing model inputs, i.e. features, from nearby sensors and performing collaborative inference between the associated models. These models might have evolved based on different scenarios, so it is possible to discover new events that were not considered during application design [1]. Other approaches consider a combination of supervised and unsupervised learning techniques, where the learning method to apply depends on data classification stage [16].

2.1.4 Data

Data producers can be classified in terms of *modality* (e.g. mobile sensors, static sensors, web services), *internet connectivity* (e.g. constant, intermittent), *privacy sensitivity*, and *resource capabilities* (when data is processed locally). Information consumers are heterogeneous in terms of *running environments* (applications that run locally or at community-level remotely), *data needs* (high-level information or raw sensor data). This heterogeneity leads to several challenges on data management [3].

Different sensors consider the physical and virtual world with different levels of accuracy. Lack of correlation between data collected from distinct viewpoints and resolutions leads to an ineffective

data merge and processing. A sensor may sense the same event under different conditions and classify it differently, yielding inconsistent results. Also, due to environmental differences, a group of sensors in the same location might sense the same event in time and infer different results with the same algorithm. For these reasons, to be able to integrate the system, data needs to be mapped to a shared vocabulary, respecting the same metrics [3]. The resulting systems are more robust to both weaknesses of sensing modalities and defective or malicious data sources than homogeneous systems [5].

2.1.5 Architecture

A taskable sensing system, i.e. a *Mobiscope*, can take form in the following conditions: it is deployed to achieve a specific goal; already deployed devices are federated through their owners; virtual *mobiscopes* are formed by correlating gathered data [5].

These technologies are still in their beginning, leading to a lack of normalized architectures [1]. The placement of concerns on system components (e.g. remote servers, mobile sensing devices) has to be further researched [1].

As these systems have no control over human mobility patterns, the coverage of spaces, events and human interactions becomes opportunistic [2] [6]. In order to face mobility, decisions are taken in real-time [5].

Sensing devices enjoy a high degree of heterogeneity. Typically, sensed data has varying time-space resolutions and may become biased depending on the sensing context. Nonetheless, the heterogeneity in sensing data can be harnessed to increase system robustness by exploiting distinct views that may complement each other [5].

Sensing devices, designed primarily for other purposes, have resource limitations that require careful consideration as to where data processing takes place [2]. One approach is to persist data by employing local buffering capabilities [5]. However, for analysis that require large amounts of data, local storage limitations may promote the need to have data persistence on remote servers [2] [6]. Privacy issues also need to be considered as it may be inappropriate to store sensitive data in a remote untrusted system.

Connectivity issues in the system affect sensing performance. In this sensing networks, at a given time, a greater amount of data is gathered when compared to data that can be delivered. To circumvent this issues and avoid resource waste, data prioritization schemes [5], to be used when multiple nodes cover the same area, have been suggested. Opportunistic data diffusion schemes between sensing devices, with possible data aggregation, aim to improve connectivity and data quality despite data incongruence [5].

Information needed by an application may only be available by integrating data from multiple sensing modalities. As such, transmitted data must be compatible across heterogeneous networks [5].

Data analysis techniques require a systemic view, considering the sensing devices' resource constraints, communication costs to remote servers and the sampling rate required to detect and characterize interesting phenomena [2].

There is a high correlation between data accesses and user location. Because of the dynamic nature of sensor densities in both time and space, system performance depends on the mobility patterns of the sensing devices. Uniform coverage for a given is area is hard to achieve as sensors tend to visit zones in a given area in a non-uniform fashion. As such and adding the fact that interesting events might be rare, sparse data models need to be considered. For such cases data-mining techniques can be applied. Another approach is to have actuated sensing devices, i.e. sensors that are tasked to visit uncovered areas [5].

Some authors have provided a systematical architecture that can be used as a viewpoint to face this issues.

An architecture, consisting of five layers: *pervasive sensing*, *data anonymization*, *hybrid learning*, *semantic inference*, and *application* [3].

The *pervasive sensing* layer involves the gathering of data from the different data sources (mobile devices, static sensors, social web).

The *data anonymization* layer anonymizes sensed data, offering different anonymization algorithms that can be applied according to the nature of the requirements.

The *hybrid learning* layer applies data analysis algorithms to convert low-level single-modality sensing data into high-level features or micro-context. Its focus is to mine data patterns and derive behaviour and single space context, before multi-modal intelligence is extracted.

The *semantic inference* layer is needed when different micro-contexts need to be aggregated. Its objective is to match the inputted micro-contexts with an expected high-level result.

The *application* layer provides a set of accessible services that are sustained on the other layers. Applications may be installed directly on a mobile sensing device or on remote servers, communicating with the sensors.

Some authors [2] [1] propose a three stage *Sense*, *Learn* and *Share* architecture.

In the *sense* layer, sensing interaction-based mobility-enabled data is acquired from the heterogeneous sensors that are part of the system [2] [1]. The delivery of application sampling requests and the delivery of sampled data are part of this layer. A sampling request specifies at least one required sensor type and the required sampling context, i.e. the set of conditions required. Related applications may be present on the mobile sensing devices or remote server, communicating wirelessly [2].

In the *learn* layer, information extracted from raw data is analysed using statistical measures, data mining or machine-learning techniques to infer higher-level meaning [2]. Data analysis techniques and features to analyse are chosen to best fit the availability and characteristics of the sensed data and the target application [2] [1].

In the *share* layer, learned information is visualized and shared according to its application [2]. A personal application will inform its user and a community application will share aggregated information with its target group, while obfuscating their identity. Resulting information can also be used to persuade users to make positive behavioural changes [1].

These presented architectural views complement each other. Their comparison is presented in Table 2.1.

Architecture	
Share	Application
Learn	Semantic inference Hybrid learning
Sense	Data anonymization Pervasive sensing

Table 2.1: Architecture comparison

2.2 Computational Epidemiology

Computational epidemiology consists on the development and use of computer models to understand the diffusion of disease through populations with regard to space and time [17]. In epidemiology, contact tracing is the process of controlling the spread of an infectious disease by identifying individuals who were previously exposed to it [18].

In order to accurately predict and understand the propagation of diseases, the data used in these models should be representative [19]. Nonetheless, decisions have to be made with limited information. An effective prediction is difficult, especially if initial data is not expressive enough [20].

Traditional systems obtain model data either through periodic online questionnaires [21], trusted web news sources [22] or by exploiting web search queries to monitor health-seeking behaviour [23]. Social contact networks constitute a potential new data source as large-scale relevant user-related data can be acquired instantaneously and in real-time [19].

As a consequence of their capability to estimate disease propagation, these models are a powerful tool to evaluate the course of a disease in response to public health interventions [17] [24] [25] [26] [27]. The more is understood about infectious disease spreading, the more efficiently it is possible to deploy measures to counter outbreaks, such as vaccines [28] [23] [21].

The following terminology is relevant in epidemiology:

- N population size;
- β effective contact rate, i.e. rate of disease contraction;
- δ recovery rate, i.e. rate of disease recovery;

- R_0 basic reproductive number, i.e. number of susceptible individuals with no immunity that an infected individual will infect [29];
- $1/\epsilon$ average latent period, i.e. period in which infected individuals cannot transmit the disease;
- $1/\gamma$ average infectious period, i.e. period in which infected individuals can transmit the disease;

One approach for the approximation of δ, which is analogous for other rates, is determining the mean time associated to the said rate and inverting that value. For instance, one person is sick with influenza for a period that can range from three to seven days. Hence, the mean time spent as infectious is five days and the recovery rate, measured in units of $days^{-1}$ is 0.2 [30].

R_0 is usually estimated from the average number of secondary cases resulting from one primary case in a population of susceptible individuals [31] [32]. It stands for the total number of expected cases for every generation of infection in a disease. A generation is the mean time between an individual getting infected and transmitting the infection to others [31].

R_0 is highly dependant on the contact patterns that determine the disease transmission. Thus, measuring it in a location where the rate of contact is unusually high will lead to estimates that have poor generalisation. Contact rates in the population may be considerably lower, thus affecting the estimation of the value of R_0 [31]. Moreover, this parameter assumes that the population is constituted of susceptible individuals only (i.e. no a priori immunity). This is not the case for certain epidemics, such as the seasonal influenza [32].

Samples of values estimated for R_0 are present in [29]. It should be noted that $A(X)$ stands for influenza A virus subtype X.

- Pandemic influenza: $1 < R_0 \leq 2.4$
- A(H3N2) in Hong Kong (1968-69): $R_0 \approx 1.7$
- A(H1N1) in the USA (1918, second wave): $R_0 \approx 2.0$

Associated to this number, there is the transmissibility parameter or effective contact rate (usually known as β or T). Contrarily to R_0, this parameter can be extrapolated from one location to another, even in cases where the contact patterns diverge in a significant way [31]. An example is provided in [31]: in a given place with $R_0 = 2.7$, an individual may come in contact with one hundred other individuals. As such, the probability an individual will get infected from an infected contact is 2.7% or $T = 0.027$ (in an uniform contact network). If this individual moves to another location where there are 10 potential contacts and R_0 is extrapolated, it implies that on average 2.7 out of every 10 contacts or 27% of the contacts will become infected. On the other hand, if T is extrapolated, one can still verify that 2.7% of all contacts get infected [31].

$$\beta = pc \tag{2.1}$$

Equation (2.1) defines the effective contact rate of a given infection. p is the probability that a contact will result in an infection, while c is the rate at which individuals come into contact with each other, i.e. the contact rate [31]. Some models rely on a set of fixed values for c, which vary according to population characteristics, such as age and place of contact [32].

It is relevant to distinguish between epidemics and outbreaks. An epidemic results from the spread of an infection from its initial set of cases to a community level, resulting in an incidence that has population-wide impact. An outbreak is associated with cases, whose transmissibility is inherently low. In this way, the infection dies out before reaching the general population [31].

Another important concept is the *epidemic threshold*. It is calculated by the quotient between the epidemic agent's death and birth rate. It refers to the threshold above which agents can spread explosively and cause epidemics.

Endemic equilibrium is reached, when the number of infective individuals remains strictly positive for a significant amount of time. As a result, a disease remains in a population, becoming endemic [30].

It is important to note that the end of an epidemic is caused by the decline in the number of infected individuals rather than an absolute lack of susceptible subjects. Thus, at the end of an epidemic, not all individuals have recovered.

2.2.1 Model

An epidemic model is a mathematical abstraction that describes the evolution of a transmittable disease in a population.

Various parameters impact model construction. By taking them into account, while relating them appropriately, models can be defined and refined to better reflect reality and potentially go towards real-time epidemic detection rather than prediction [21].

2.2.1.A Mixing

Under homogeneous mixing, individuals belonging to the population are neighbours with every other individual, making contact at random and not mixing into smaller subgroups [33]. Here, the probability of any infected individual contacting any other susceptible individual is well approximated by the average. This is often a problematic model assumption, but it can be relaxed in complex models [34]. In this situation, the set of infected individuals has little meaning to the overall population dynamics and the relevant metric is the number of infected individuals [33]. Examples of this approach are [32], where individuals are members of social mixing groups inside which a disease is transmitted by random mixing.

In non-homogeneous mixing, the structure of the considered social network greatly influences disease proliferation as it conditions contact between individuals [17]. Examples of this approach are present in [35], [36] and [37].

2.2.1.B Spatial Distribution

Simple models assume uniform spatial distribution. More complex lattice-based models can cope with non-uniform distributions [33].

For an epidemic model to be realistic in a large population, long-distance travelling is necessary. Nonetheless, depending on the scope of the model (i.e. if the model has a reduced scope), this kind of travelling may be dismissed [32].

Epidemic models pertaining a sufficiently large area may be adequate for determining national level impact of given epidemic. However, their epidemic peak will appear later. This is due to the time it takes for the epidemic to affect surrounding areas [32].

2.2.1.C Genotypes

The genotype of the afflicted population constitutes its inherited genetic information and can determine its vulnerability against a given infectious agent and resistance towards another [24]. Recent observations suggest that the relation between ethnicity and health may be heterogeneous, and that generalising this information without considering social context may be a flawed approach [25].

The genotype of the infectious agent influences its behaviour, infectivity, resistance to public health measures and may contribute to the appearance of new sub-strains with different characteristics.

The interaction between these two variables conditions epidemic dynamics [24].

2.2.1.D Transmission

There are two directions in disease progression: *within-host progression* and *between-host transmission*. The start of *within-host progression* is triggered by *between-host transmission*.

There is a latent period between the time an individual becomes infected and the time when the capability to infect others is acquired [17]. Between-host transmission can occur in different directions: *horizontal* and *vertical*.

Horizontal disease transmission may be triggered through various forms of contact: direct contact; indirect contact (e.g. contact with a contaminated surface); droplet contact (e.g. sneezing), airborne contact (if the pathogen is resilient enough to survive in the air); fecal-oral contact (e.g. contact with contaminated food or resources) [24].

Vertical disease transmission occurs from mother to child (e.g. in the case of AIDS and Hepatitis B).

2.2.1.E Infectivity

An individual's capability to transmit disease varies over time and is viral load related. In the case of influenza, symptomatic individuals can be twice as infectious as asymptomatic ones [32].

2.2.1.F Age Structure

A rectangular age structure assumes people live to reach the average life expectancy of the population. This model is suitable for developed countries. For other countries a triangular age structure is considered more appropriate [38].

Also, infectivity can be age-dependant. For instance, children tend to infect and get infected more often than adults. Consequently, in the early stages of an epidemic, the number of cases involving children is likely to be higher [32].

2.2.1.G Epidemic Reaction

The behaviour of people is changed in response to the menacing nature of an epidemic. One future direction in computational epidemiology is to take this into account [21].

2.2.1.H Granularity

A model unit can be defined to represent a single individual or larger social units (groups, families, a small location, organisations or a country) [35] [39].

Aggregate models assume a population is partitioned into sub-populations with a predictable interaction structure within and between them. While, these models are useful for obtaining parameters, such as the total number of infections, they lack the capability to capture the complexity of human interactions that serve as a major infectious disease transmission mechanism. Also, they are incapable of providing causal explanations. The capability to provide specific details about the flow of disease spread may be required to provide insights to researchers investigating interventions against the epidemic. As the granularity of sub-populations is considered to be high, parameters such as the base reproductive number (R_0) and the contact rate are hard to observe [17]. Models under this category may group the population into a hierarchical set of mixing groups with decreasing social proximity (e.g. from a family household to a community), providing an explicit community structure. For these models, the attack rate of the epidemic, i.e. the ratio of the number of people infected with the disease over the number of exposed people, is lower than the one for models with random mixing [32].

Disaggregate models (or individual-based models) use a representation of individual agents with explicit interactions between them to model the disease spread across a social network, offering a much finer granularity [17]. If one aims to obtain an history of the disease evolution or tackle detailed intervention strategies these models represent the most suitable solution [32].

2.2.2 Mathematical Formulation

Epidemiological models can be classified depending on their mathematical formulation [24].

2.2.2.A Deterministic

These models can be used to study an epidemic analytically and constitute the most popular approach for disease modelling [32]. Under this formulation, individuals are assumed to be uniformly distributed in space and to mix at a certain rate, i.e. the effective contact rate β. They are usually based on the Susceptible-Infected-Recovered (SIR) compartmental models [24] [32], while predicting that an outbreak with an $R_0 > 1$ will originate an epidemic [31].

They are sustained on a set of differential equations and partition individuals across model-dependent compartments. Asymptotic behavior for resulting systems depends on parameter choices [24].

The reductionist linear nature of these models is limiting in relevant ways. Population dynamics, in terms of health and disease, emerge from interactions between the heterogeneous individuals that are part of these models [25]. This results in complex relations between individuals and originates complex social networks [21], that hold the mechanics that trigger the social production of health and disease [25]. Social factors interact in complex ways to influence disease risk, hence this models' decontextualized measurements for exposures may be inappropriate [25]. This notion becomes clear when taking into account the high degree of mobility people enjoy, easily travelling abroad while carrying an infectious disease with them [21].

An example of this formulation is present in [40].

2.2.2.B Stochastic

In this formulation, the probability distribution of potential outcomes in disease propagation is estimated by allowing input data to vary randomly over time. Systems may be modelled as a Discrete Event System (DES), in which system state only changes upon the occurrence of an event [17]. A Finite State Machine (FSM) variant, called a Probabilistic Timed Transition System (PTTS) may be used to represent within-host disease progression. In this approach, state transitions are probabilistic and timed and the considered states depend upon the chosen implementation [17]. The theory of stochastic processes defines the asymptotic behaviour for systems using this formulation [24].

Stochastic models usually run in discrete time and may even consider different steps in the simulation to account for periods where the mixing between individuals differs (such as night and day) [32].

The running time of these models is dependant on the number of infected individuals in the course of the simulation [32].

Examples of systems recurring to this models are [41], [42] and [17]. However, despite the fact

that many of these models have been described in the literature, few have their implementation publicly available [32].

An Agent-based Model (ABM) [25] provides information on all simulated individuals and all the significant interest factors that relate to them (i.e. infection, incapacitation, and treatment with the associated time and location), resulting in large simulations [35]. By simulating individuals, over a simulated space and time, one aims to clarify in which way macro-level insights on health and disease distribution patterns may emerge from pre-defined micro-level rules (e.g. health behaviours, individual-related interactions, and mobility in frequented environments). Conceptualization and parametrisation take a bottom-up approach, which enables the assessment of the emergence of macro-level patterns [25].

They are most appropriate when:

- Individual behaviour is complex (comprised of learning, adaptation, feedback, and reciprocity).
- Heterogeneous environments can impact behaviour and interaction, and individuals are dynamic over either space or time.
- The interaction between individuals is complex, non-linear, and can influence the behaviour of others.

In this context, individual behaviour is a function of its attributes and characteristics, environments, and interaction over time. Social processes are one example of such a set of conditions.

In every simulated unit of time, researchers update simulation parameters, resulting in stochastically-applied changes in behaviour and characteristics. Hereon, they can go into experimental simulations that fork from an initially well-defined control one. This can promote causal thinking in the analysis, thus improving the understanding on the social production of health and disease. This contrasts with deterministic and regression methods, which feed on aggregated data and whose results are a priori bound to it [25].

In the model implementation process, as the output precision from complex models is limited, it is important to strike a balance between model rigour and parsimony. When considering stochastic models, added variables are likely to increase the already high computational costs and the degree of uncertainty in simulation outcomes [25].

These approaches may not be well-suited when attempting to attain absolute population-level metrics, such as disease prevalence or incidence. Another limitation is the real possibility for these models to be parametrised by data that results from reductionist approaches, conducing to result bias. Finally, validation in systems models is often impossible to do completely. To attempt model validation, a researcher might:

- Use real data to parametrize the model.

- Construct a model from conceptual relations, whose findings can be compared to real world observations.

Either the relationships between factors in the model or the outcomes of the model can be validated, but rarely both. In the first approach, the validity of the simulations depends on the valid extraction and implementation of factors in the model, which is nearly impossible to verify, even when those factors are extracted from real data. In the second approach there is a threat resulting from the fact that while a particular model configuration might produce outcomes that predict the observed data, it is possible that there is any other number configurations that would also result in the observed data. As a consequence there is no way to ensure that the configuration that leads to the results observed in reality has been isolated [25].

2.2.2.C Network-based

Latest developments in epidemic spreading emphasize the importance of network topology in epidemic modelling [9].

Social and biological systems can be described by complex networks whose nodes represent the system actors and its links the relationships between them [9]. They may be modelled as a computational network, which is itself modelled as a graph. A graph consists of a set of points called nodes or vertices. Interconnections between these nodes are named links or edges and, in this application, they represent a form of contact or relation. The degree of a node k corresponds to the number of neighbours it has [35].

Network-based epidemic models predict that there is a probability S that an outbreak with an $R_0 > 1$ will result in an epidemic. This probability is often much lower than one, and it can differ for two networks that share the same R_0. For the cases where S is significantly less than one and R_0 is clearly above one, networks that share similar contact patterns are likely to show disparity on their results (minor outbreaks versus large epidemics) [31].

An important result in network models is the prediction of a non-zero *epidemic threshold* (λ_c). The higher a node's connectivity, the smaller the *epidemic threshold*, and consequently, the higher the probability of infection [9].

$$\frac{\beta_a}{\delta} \leq \frac{1}{\lambda_{1,A}} \tag{2.2}$$

Equation (2.2) represents the bounding of the *epidemic threshold* and defines a condition, that when not met implies the existence of an epidemic. $\lambda_{1,A}$ corresponds to the modulus of the largest eigenvalue of the adjacency matrix of the associated contact network topology, also known as the dominant eigenvalue or spectral radius of the network graph. β_a stands for the average rate of infection along a network edge and δ is the recovery rate of an infected node [35].

The strength of an epidemic can be evaluated with resort to the generalized isoperimetric constant ($\frac{1}{\lambda_{1,A}}$) of the associated social contact network, also known as Cheeger's constant [35].

The smaller the spectral radius of the network graph, the higher the isoperimetric constant and the resistance of the network against the spread of infectious agents [43].

Contextually, this constant is equivalent to the *epidemic threshold*. When the ratio between infections and cures is lower than the *epidemic threshold*, an infection dies out quickly (exponentially fast). Conversely, if it is higher an infection will die out slowly, resulting in an epidemic [35] [9].

In these models, one way to accommodate asymmetric and variable contact is by weighting the links of the contact networks. The weighted links distribute the contact rate parameter β_a over the network. The weight value and its distribution can have a significant effect on the epidemic resistance of the topology, offering the possibility to alter a network without changing its topology. This introduction gives rise to a new form of clustering, i.e. weight clusters. Such clusters can boost infectious agent spread through the network [35].

$$\beta_a = \beta_f \overline{\omega} \tag{2.3}$$

Equation (2.3) introduces a weighting parameter $\overline{\omega}$ that accounts for the average contact rate placed on the network topology. β_f stands for the full contact measure of the effective contact rate β [35].

$$\begin{gathered} \beta_f \omega_{ij} \leq 1 \\ \frac{\beta_a}{\overline{\omega}} \omega_{ij} \leq 1 \\ \omega_{ij} \leq 1 \\ \beta_a \leq \overline{\omega} \leq 1 \end{gathered} \tag{2.4}$$

The upper bound for the link weight parameter for an edge from node i to j and its impact on the average link weight $\overline{\omega}$ is presented on equation (2.4). The weights for a given link ω_{ij} take values between 0 and 1. As for matrices A and B with a_{ij} and b_{ij} respectively $\lambda_{1,A} \leq \lambda_{1,B}$ holds, the upper bound ensures the link weights do not directly alter the epidemic threshold of the network [35].

$$P(k) = k^{-\gamma} \tag{2.5}$$

In these models, specific network topologies are favoured. Scale-free networks are the family of networks whose degree distribution obeys a power-law, made explicit in equation (2.5). γ is a coefficient that in this context of analysis is subjected to $2 < \gamma \leq 3$ [35] [9] and k constitutes the number of connections per node, i.e. the average node degree.

They are constituted of nodes with few edges, having a minority of nodes with a high degree [31]. Each node has a statistically significant probability of having a very large number of edges, when

compared to the average connectivity of the network $\overline{k}$. This notion is distinct from conventional random networks, where each node has approximately $\overline{k}$ links [9]. Thus, a scale-free network has considerably more nodes of high degree than a random network [27].

This inherent large fluctuations between the number of connections in each vertex makes them appropriate to model real social networks [35] and computer virus epidemics [9]. These variations lead to the existence of local clustering, i.e. the existence of nodes with a very large number of connections [9]. However, for this topologies, even the best fit power law model may fail to predict a valid epidemic threshold for certain diseases, contrasting with what is known about them [8].

In these networks, the final size and persistence time of a given epidemic are highly sensitive to the multi-scale hierarchical structure of the considered population [28]. For instance, nodes that are in contact with a large number of other nodes are easily infected and constitute a bridge for the spreading of infections [28] [24] [27]. These well-connected nodes will enable the potential for an outbreak if there is a non-negligible probability that an infection will reach them, resulting, in the case of large scale-free networks, on the vanishment of the epidemic threshold, i.e. $\lambda_c = 0$ [9] [28]. This suggests that even weak infections can spread [28] [31] [9]. Moreover, this indicates that infections can spread regardless of their spread rate or effective contact rate ($\beta \ll 1$) [9].

In networks with bounded connectivity, epidemic prevalence is always below the epidemic threshold, resulting in the eventual death of all infections. In scale-free networks, the unbounded nature of the fluctuations in node degree lead to a state of infinite connectivity in the limit of network size. As a consequence if network size increases, an infection will last longer [9].

Network-based models may be analysed using integrated statistical and intelligent analysis methods to produce stochastic model input data, effectively representing a labelled social contact network. Modelling challenges come from the large size, irregularity and dynamism of the underlying social contact network [17].

2.3 Social Network Analysis

Recently, there has been an increase in the interest of systems approaches applied to epidemiological research. Furthermore, these appear well-suited for social epidemiology [25].

They imply that the dynamics of a system are different, qualitatively, from those of the sum of its parts. The relation between system components is considered more relevant than the attributes of its components by themselves. In the case of network approaches this means a bigger emphasis on the structural characteristics rather than on nodes characteristics. Therefore, the social ties that influence network actors have important consequences in the analysis [25].

Social network analysis involves the characterization of social networks to infer how network structures may influence the exposure risk among its nodes. It pertains the characterization of the components and subcomponents of social networks in order to understand their impact on health and disease. It constitutes the framework in which to analyse factors that have an impact

in populations. Moreover, it is a major improvement when compared with techniques base upon the assumption of independence of observations (deterministic and regression-based techniques) [25]. This assumption is not verified in network data as it is inherently relational [27] [25].

Disease, information and social support are among health-relevant factors of interest that may impact networks nodes [25]. With the appearance of the Acquired Immunodeficiency Syndrome (AIDS), social network analysis was demonstrated to be suited for infectious contact tracing [18].

Social networks for the investigation of infectious diseases are typically built with resort to personal contacts only, as these contacts are the most traceable means for disease transmission. Nonetheless, in the case of diseases which transmit not only through personal contacts, the inclusion of geographical contacts into social network analysis methods has been proven to reveal hidden contacts [18].

Social network analysis is comprised of three main branches:

- Network visualization
- Network characterization
- Stochastic and longitudinal network analysis

Network visualization focuses on diagramming network connections in order to visualize network structure and relationships.

Network characterization is directed towards understanding the roles played by individual actors, subgroups of actors, or global network structure while analysing the flow of factors of interest within the network. One example of such an analysis is the comparison of connections an actor possesses in relation to others.

Stochastic and longitudinal analysis intend to exercise inferential analysis on networks. While longitudinal analysis allow researchers to study the dynamics of temporal changes, stochastic analysis permit the construction of models targeted at network simulations [25].

2.3.1 Representation

Networks constitute a useful construct to represent data that possesses properties, whose scope goes beyond the attributes of its nodes, by providing a way to map relationships between them with resort to its edges. In this context, networks can be used to model different ideas, such as the behaviour of epidemics.

For networks whose vertexes stand for individualised social actors and whose edges represent actor relationships, random graph model representations have been the target of recent social network studies [39].

Social network data can be represented by a set of N actors and a variable that constitutes their relational tie Y_{ij}, for all ordered actor pairs (i, j), such that $i, j \leq n$ and $i \neq j$. Y_{ij} is usually dichotomous and it indicates the absence or presence of a relationship between two nodes. It can

represent disease transmission, friendship, contacts, academic collaboration or any other form of relationship.

This data can be thought of as a $n \times n$ matrix named Y, whose diagonal elements represent relations between a node and itself, whose value is typically zero. When considering binary relations between actors, this data can also be represented as a graph, whose nodes map the social actors and whose edges correspond to their relations.

The relations between actors are undirected if $Y_{ij} = Y_{ji}$ for $i, j \leq N$ [39].

2.3.2 Sampling

Sampling matters in the creation of a credible mathematical basis for statistical inference on a social network. This can be shown in the bias that arises from missing data in modelling approaches [8].

Populations of actors can be hard to find and often it is hard to obtain data representing all actors and relations. Furthermore, most inference social network models assume that the information obtained is reliable and that the necessary measurements result in no errors [39]. In practice, this is not verified as individuals and links between individuals are typically missed (i.e. they are not part of the recorded data) [27] [39].

In sampling, the unit is the node, while the unit of analysis is most commonly the dyad or relation. Under many sampling designs the set of sampled nodes determines the set of sampled relations [39].

A sampling design is:

- *Conventional* if it does not take advantage of the information collected to direct subsequent sampling of individuals.
- *Adaptive* if it employs the information collected to direct subsequent sampling, and the sampling design depends only on the observed data.

Various sampling strategies may be employed. They are supported on different approaches for network sampling and are grouped under *conventional* and *adaptative* designs [39].

Egocentric sampling is a conventional network sampling design whose steps are:

1 The selection of individuals at random with a given probability.

2 The observation of all dyads involving the selected individuals (i.e. the complete observation of the relations originating from them).

Alternative conventional designs designs consider the probability sampling of pairs and auxiliary variables [39].

Examples of *adaptative* designs are: *One-wave link-tracing*, *Multi-wave link-tracing* and *Saturated link-tracing*.

The process for *One-wave link-tracing* is comprised by:

1 The selection of individuals at random with a given probability.

2 The observation of all dyads involving the selected individuals (i.e. complete observation of the relations originating from them).

3 The identification of all individuals that have at least one relation with the initial sample, and their selection with probability 1.

4 The observation of all dyads involving these individuals.

For *Multi-wave link-tracing* or *snowball sampling*, the process described for *one-wave link-tracing* is executed k times. If k is a fixed value, then it is called *k-wave link-tracing*.

Saturated link-tracing is a variant of *k-wave link-tracing* design, in which sampling continues until wave m is reached and where $k < m$ and for which $samples_m = \emptyset$. K is the bound on the number of waves sampled. Since this design does not explicitly limit the number of waves, $k = \infty$ [39].

Nonetheless, these methods possess limitations. *Snowball sampling* finds people in proportion to their contact centrality, but equilibrium might require a number of waves that is unreasonable. In *random walk sampling*, sampling is proportional to degree and convergence can be reached quickly for small groups, but the issues associated with unfindable nodes still persist [8].

Lastly, random sampling techniques designed to improve result generalizability may not be suited to network approaches, as the resulting relational data may lack sufficient completeness and accuracy. More concretely, in a population sample these techniques may collect data about disease exposure, its health outcomes, and the number of social contacts each actor has. However, they will not gather exposures and outcome data that is actor-relational [25].

2.3.3 Limitations

In the process of analysing social networks in the context of epidemic inquiry, theory and empirical data seem to be part of two separate discussions. This is reflected in the notion of abduction, a concept that exists between induction (generating theory from data) and deduction (testing theory with data).

The different skill sets that are required and the substantial obstacles to the collection of human network data promote data deficiencies. These along with the ease at which data can be generated by simulations contribute to this issue [8].

Analysis requires network data, which may impose restrictions on result generalizability. Causal inference from currently used network analysis methods is limited, as there are considerable limitations resulting from observational models. Population dynamics are non-linear processes, i.e. a change in disease risk is not always proportional to the change in disease exposure. Furthermore,

these relations can be affected by feedback, resulting in confounding, i.e. a situation where disease can modulate exposure just as exposure can modulate disease.

The conceptual framework that supports epidemiological inquiry is not enough when taking into account both low-level social causes and macro-social causes for disease [25].

Some other factors that contribute to the chasm between the theoretical and empirical approaches are:

- The cost of acquiring the appropriate social network structure is prohibitive, especially for hard-to-reach populations.
- The way in which the relations between individuals and nodes are sampled is not standardised, leading to missing data and hindering information interpretation.
- There is no standard network model to measure against empirical results.

This mismatch results in an increase in theory-based network simulation, where the researcher controls the sampling procedure, creates a standard and measures the effect of the imposed parameters. In these approaches, the assumption of theoretical validity can be emphasized in such a way that empirical verification is missed completely. Researchers are then unable to assess if the reached answers are the valid ones, the reviving the problem of inductive reasoning, which directs epidemiological analysis.

Empirical network descriptions, both qualitative and quantitative, have the potential to find abstract characteristics of a pattern, a task for which theory and simulation are not well suited by themselves. Theoretical analysis is suited to exploring patterns, and they often do it best while decoupled from reality. Nonetheless, they should be directioned towards demonstrating mechanisms and testing observations, i.e. the deductive processes [8].

2.4 Development

2.4.1 Security

Respecting the privacy of its users is a relevant concern in a mobile sensing system [1] [3]. People are sensitive about how their data is captured and used, especially if it contains their location [1], speech [16] or sensitive images [1]. Interestingly, social network application users may take privacy as a less relevant concern [4].

Collected data may inadvertently reveal information about people. For instance, a connection between mobile sensors and observed parties may be implicit in their user's relationships [5]. Revealing personal data can risk privacy and sharing community gathered data can reveal information on community behaviours [3].

Revealing too much context can potentially compromise anonymity and location privacy. Conversely, the inability to associate data with its source can lead to the loss of context, reducing the

system's ability to generate useful information [5].

People may fear that personal information will leak from the system. Even individuals that are not sensing targets may be vulnerable to accidental privacy breaches, if they are close to a sensing device [2]. Countermeasures pausing the collection of sensor data, are unsuitable as they may cause a noticeable gap in the sensing data stream [1].

Privacy protection involves different variables, including identity (who wants data access), granularity (level of data revealed), and time (retention time of data) [3]. These issues may be tackled by the following approaches.

2.4.1.A Authentication

It deals with validating the user before the system. The sheer amount of users in mobile sensing systems might pose impediments to cryptographic authentication. Nonetheless, there is the possibility of relying in the redundancy of sensor data to validate a source anonymously [5].

2.4.1.B User Control

Control over data sharing allows users to define their participation in the system, empowering the decision making process [3]. One approach is keeping sensitive relations from being exposed, either by local filtering or by providing users with an interface to review data before it is released [5]. In [16], the user has complete control in how information is presented in the different system interfaces.

2.4.1.C Anonymization

Before data release and processing, different algorithms may be applied with the objective of not revealing user identity [3]. Some approaches can help with these problems (e.g. cryptography, data and privacy-preserving statistics) [5] [1]. Nevertheless, they may be insufficient [1].

In personal sensing, a solution is processing data locally [1] [16].

In the context of community sensing, there is the risk of leaking personal and community information.

Reconstruction attacks target innocuous-looking data and allow invasive information to be reverse-engineered [1]. A solution is for privacy to be based on group membership. Sensitive information is only shared within the groups in which users have existing trust relationships [2] [4].

2.4.1.D Trust

Ensuring both data sources are valid and that information is accurate, should be a system concern. Also, correct system use should be promoted to prevent abuses. When mining social and community behaviours, anonymous data is needed [3]. Data correctness must be verified without

violating privacy [5]. In opportunistic sensing schemes user trust may become a barrier to wide-scale adoption [2]. These issues may be addressed by providing sensing device users with a notion of anonymity through k-anonymous tasking [2].

2.4.2 Continuous Sensing

Real world systems are continuous. To model them, it is necessary to consider multiple samples in the data stream and models need to cope with sampled input [3]. Continuous sensing has applications in various sectors, particularly in personal healthcare.

Smartphones, while fairly powerful in computational terms, present limitations in supporting the needs of continuous sensing applications [4]. This view has to kept in mind when implementing these systems [16]. A requirement for continuous sensing is that the sensing device supports multitasking and background processes [1].

According to [1], only Android and Nokia Maemo phones support this capability, while iOS supports the notion of multitasking. However, due to commercial decisions, applications are not allowed to run as daemons, making it inadequate for continuous sensing [1] [16].

Some phone sensors sample at unpredictable rates, depending on the load. This may constitute an issue when consistent sampling rates are required (e.g. statistical models to interpret data) [1].

2.4.3 Resource Management

Applications that offer good fidelity and user experience without significantly altering the lifetime of the sensing devices should be offered [4]. Some sensors use a varying amount of power depending on external factors. Lack of sensor control limits the management of energy consumption [1].

A real time sensing system should supply sensor data at an high rate. However, such an approach yields high energy costs. The challenge to reduce energy consumption in expensive radio communications without significantly altering application experience exists. Data upload through the General Packet Radio Service (GPRS) can consume a large amount of energy, especially when the sensing device is far from base stations [4].

Data analysis algorithms can also draw a considerable amount of energy, if not optimized [4]. In [1] it is shown that these applications can reduce a phone's standby time from 20 to 6 hours.

Four complementary approaches to these problems are considered.

2.4.3.A Efficient Duty Cycle

The trade-off between energy consumption and sensing fidelity results in an impact in system accuracy and responsiveness [4] [16] [1]. Reducing the use of radios that consume a high amount of energy (e.g. the Global Positioning System (GPS)), is one approach to save power. Another approach is to minimize sampling while respecting system responsiveness, judged by tests with

users [4]. In case data analysis can withstand big delays, data can be collected and stored until there is the possibility to upload it [1].

2.4.3.B Summary Sending

Sending extracted features from the raw sensor data instead of raw data can be done to save power spent in transmission costs. A disadvantage of this approach is that it may not be sufficiently energy-efficient for continuous sensing [1].

2.4.3.C Computation Splitting

Resource restrictions become evident when faced with complex signal processing and data analysis algorithms [1] [16]. Application designers have to consider an accuracy and computational cost trade off [16].

One option is to reduce resource usage by leveraging remote processing infrastructures, to where different sensor data processing stages are sent whenever possible [1].

Another possibility is the use of local algorithms proposed by [16] are capable of minimizing computational costs, while compensating for imperfect sampling and the lightweight nature of the analysis process.

2.4.4 Implementation

Most smartphones in the market are open and programmable by third-party developers and offer access to a Software Development Kit (SDK), an Abstract Programming Interface (API), and other software tools [1]. With these tools, it is possible to leverage existing software, such as established data analysis libraries [1].

* FIXME Some manufacturers and operators limit the programmability of mobile phones to maintain the closed nature of their systems [4]. Also, as vendors did not anticipate the use of real-time continuous sensing to develop new applications, fine-grained sensor control is not possible [1] [16]. Finally, vendors offer different APIs, making the port of an application to multi-vendor platforms challenging and bound to specific programming languages [1].

A web technology based approach tackles issues of system universality, as it allows different types of devices to access the system, providing flexibility in terms of programming languages [6] [19]. In this approach, a publishing client on the sensing devices phone can collect samples and upload them using the web communication interface, after applying data filters and according to network availability. This client runs in parallel with other applications and is activated as needed, allowing users to configure data acquisition policies [6]. A remote server hosts the web communication interface and accepts data samples [6] [19]. An always available interface acts as a sensor representative for the mobile sensors, providing an Uniform Resource Locator (URL) that can be used by applications to access specific sensors, while accounting for the possibility of the physical sensor being disconnected. This interface provides data that is yet to be sent to

the sensor, whenever the sensor becomes reachable. A tasking server would determine the sensing devices that would be better-fitted to initiate or stop sensing, taking application requirements into consideration [6].

2.5 Other Applications

Millions of people regularly participate within online social networks. In [4] the use of phone sensors to automatically classify events in their lives is investigated. These classifications can be selectively shared using online social networks, replacing manual actions that are currently performed daily [4] [1].

In [16] a lightweight and scalable hierarchical audio classification system, designed with resource limited mobile phones in mind, while remaining capable of recognizing a broad set of events, is provided. In opposition with offline audio context recognition systems, classification is performed online at a lower computational cost, while yielding comparable results.

Conventional ways of evaluating environmental impact rely on aggregate statistical data that applies to a community [1]. In [13] a personalized environmental impact approach is developed, allowing the tracking of human actions and their impact towards urban problem exposure and contribution.

In [14] continuous physical activity data is captured and related to personal health goals in the form of user feedback [1]. These applications have been proven to be effective in impacting the way health is assessed, helping the improvement of behavioural patterns [1].

3

Architecture

The solution proposed is composed by the following architectural modules, as depicted in Figure 3.1.

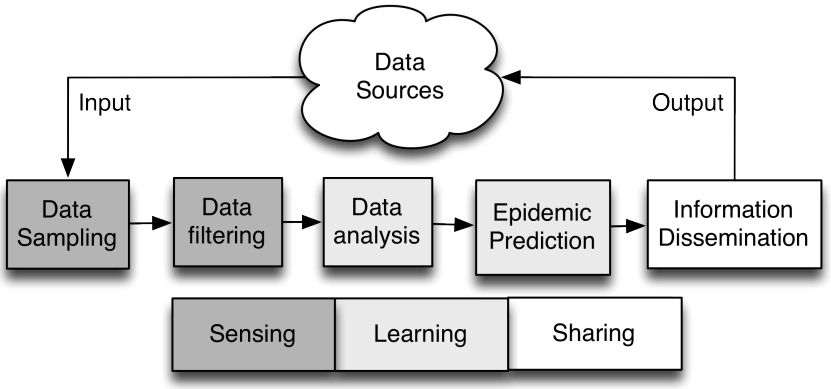

Figure 3.1: Solution Architecture

Both the *data sampling* and *data filtering* modules constitute the *Sensing* layer of the system. The aim of this layer is to obtain input. Sampling respects user privacy requirements, meaning that both a user's network and contact data are not disclosed to the system without explicit permission. Sensing, only occurs if requirements are met.

The *Learning* layer is constituted by the *data analysis* and *epidemic prediction* modules. In this layer obtained data is transformed and integrated into a model, contributing to the extraction of intelligence in the context of the applicational problem.

Information dissemination comprises the *Sharing* layer, where system output is returned to its

users.

The concepts that follow are relevant. An analysis round constitutes a end-to-end system run, i.e. the flow of data from through these modules. From when it leaves the data source to the instant in which results are returned to the users. An actor is an entity that is part of a social contact network. All sourced actors have a corresponding system mapping, but not all of them carry sensing devices, i.e. they do not take part in the system directly. This means that not all actors are users.

Another important distinction is the one between users and clients. Users are the individuals that participate in the system. Clients are the entities responsible for opportunistic data sampling.

A description regarding the purpose of each module ensues.

3.1 Data Sampling

In this module, raw data is gathered periodically from the data sources.

The delivery of application sampling requests and sampled data are part of this layer.

For each individual taking part in the system, the data gathered may belong to one of the following categories, depending on its modality.

- Network Data, i.e. relational data constituting an individual's social network.
- Contact Data, i.e. inferred meetings between individuals.

While network data does not vary significantly over little, contact data is prone to change more frequently.

3.2 Data Filtering

This module operates on the field of data anonymization and filtering.

To ensure that user identification does not reach the next modules in the chain, it is mapped to another value in an irreversible transformation. This transform has to preserve context, so that the properties required for data merging and processing are respected.

Data deemed relevant for the application is kept, while data that may lead to erroneous system behaviour is discarded.

3.3 Data Analysis

As the acquired data by the system constitutes different data modalities for different users, it has to be aggregated in a meaningful way.

The purpose of this module is to correlate data that may bear distinct viewpoints and resolutions, processing and merging data and enabling the extraction of higher-meaning or information.

Previously acquired contact and network data is merged, concluding for all users with whom they met and which social connections exist for them.

The merging process attributes different importance to the relations associated to user ties observed in the network data by measuring them against perceived user meetings. Given that infectious disease spread is primarily achieved by direct contact, contact-related data has a higher impact in this estimation.

This process consists in the weighting of network links with the information provided by contact data, as shown in Equation 2.3. Following Equation 2.4, these weights are then scaled with resort to their average.

The merged data results in a social contact network, containing all the considered individuals. This network accounts for the relationships between all actors with their associated degree of importance in epidemic spread.

As data analysis is a computationally costly process, this module should perform on the set of all aggregated data. For each analysis round the social contact network comprising all users is evaluated.

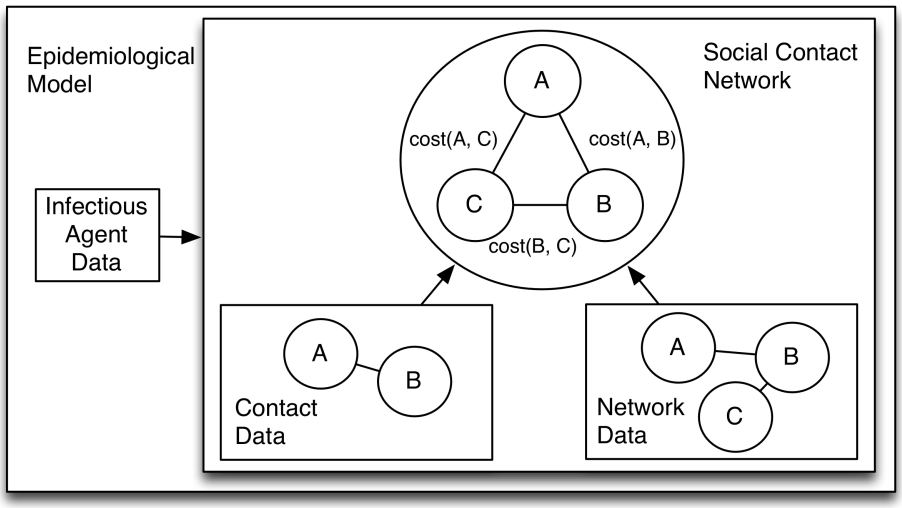

Figure 3.2: Solution Epidemic Model

This procedure is represented by Figure 3.2.

The resulting social contact network is joined with an infectious agent data, constituting an epidemic model.

The infectious agent data is predefined and fed into this module as a parameter.

User data is requested by this module when analysis is about to commence.

3.4 Epidemic Prediction

This module receives the epidemic model resulting from *data analysis*.

By employing the appropriate mathematical formulation and metrics, the evaluation of social contact network for a given epidemic agent's properties is made possible. As a result, the assessment of the dawning of an epidemic is enabled.

Social contact networks may exhibit a variety of properties. A network's node degree distribution and isoperimetric constant (or epidemic threshold) are among the most relevant in this area of application.

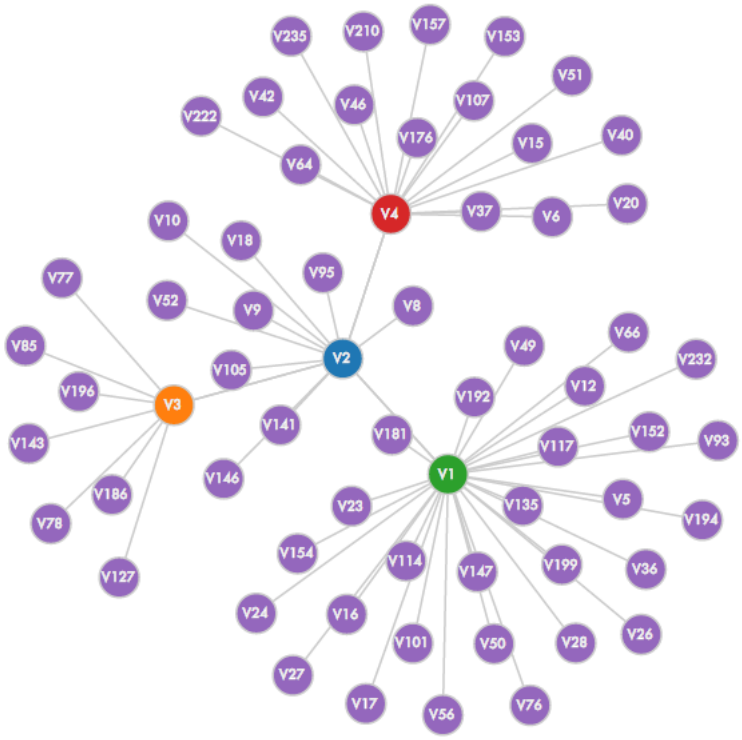

Figure 3.3: Social Contact Network Model

A network configuration is shown in Figure 3.3. Since its node degrees are arranged in an exponentially distributed way, it is a scale-free network. Colours are mapped according to node degree.

The idea behind this module is the correlation of network properties and analytical methods applied to epidemiology. More precisely, the relation between the isoperimetric constant and the epidemic threshold, as verified in Equation 2.2. By determining and comparing this metrics, the susceptibility of a network to the subsistence of disease outbreaks is assessed. Consequently, it is possible to assess a boolean measure of epidemic risk.

This module is paramount to the *learning* layer, as it is within it that intelligence is applied to

epidemic prediction.

3.5 Information Dissemination

Here, prediction metrics are transformed into notifications that are forwarded to users. This is done without breaching any of the aforementioned privacy requirements.

These notifications result from the translation of *epidemic prediction* dichotomous results into information that is interpretable by users.

After the system broadcasts this information, an analysis round is deemed concluded.

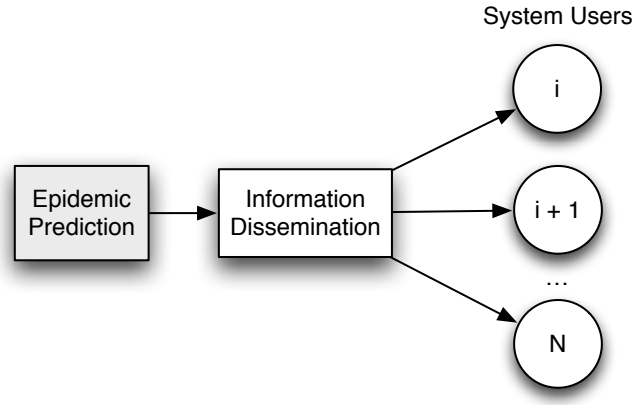

Figure 3.4: Solution Epidemic Prediction and Information Dissemination

Figure 3.4 depicts the relation between the previous module and information dissemination.

4

Implementation

4.1 Methodology

During system implementation, decisions aimed at overcoming the challenges that are inherent to the concretisation of the architecture had to be made. The rationale behind them is explained and different options presented.

A key approach in the development methodology was the exploitation of existing frameworks and libraries in the proposed solution implementation process. By taking advantage of existing validated work, the development focus is shifted, thus effectively tackling the problems at hand. Examples of this approach are materialised by harnessing the potential of works, such as:

- Facebook's Graph API [44]
- Apache Cordova [45]
- Zepto.js [46]
- Knockout [47]
- Node.js [48]
- Socket.IO [49]
- Q [50]
- Numeric.js [51]
- OpenSSL [52]

- Stanford's Javascript Cryptographic Library (SJCL) [53]

Moreover, when facing similar solutions that would cover the systems needs, preference was given to open source solutions [54].

Also, technological choices that would benefit system configuration flexibility and modularity were benefited, as these properties were deemed relevant.

Lastly, system-level platform portability was taken as a global criteria in the choice of tools to approach each of the system implementation challenges.

4.1.1 Social Network Data

Social network platforms provide the coverage needed for appropriate social network modelling.

Data has to come from social network platforms whose API offer the flexibility required by the system and are permissive enough to allow large-scale data gathering for testing purposes.

Both Facebook (due to its massive global user base), Google+ [55] (integration with other Google services), Foursquare [56] (location-aware philosophy) and Twitter [57] were considered in this decision.

Twitter was dismissed as the matching between user relations and non-virtual interpersonal relationships (more relevant to the system) was assumed to be low (allied to the fact that Twitter users have on average bigger social networks [58]).

Google+ was considered due to its ubiquousness and diversity of services, but as it is the case with Twitter, following a user does not imply any kind of real-life relation between users. Also, the Google API imposes a courtesy quota for data acquisition to access data for the bigger part of its services.

Foursquare offers an API that possesses an inherent connection between user relations and location-aware updates. However, in contrast with Facebook, it did not have any developer support service that resembled Facebook's Test User API [59].

Facebook's Graph API provides a Representational State Transfer (REST) API that enables user access to a variety of functionalities and data that are present on the Facebook platform. In this way, the social network data that is needed to feed the system is made available in a technology that is supported by the ubiquitous protocol that is the Hypertext Transfer Protocol (HTTP). This data access mechanism is similar to the other evaluated options. The striking difference is the existence of a mature API that enables the free manipulation of dummy users by its application developers, i.e. the Test User API. With resort to this API, datasets for which properties are well-known can be inserted into the system, making room for a better sustained analysis. Moreover, this API restricts the interaction between Test User data and the remainder of the regular Facebook user data. It creates, thus, an isolated and controlled environment that is adequate for system testing and data analysis. The API for interacting with Test Users is equivalent to the one used

for regular users in the terms of access to user data parameters. Consequently, the remainder of API access distinctions can be abstracted in the codebase with ease.

Furthermore, there are various studies analysing global Facebook network data [60] [61], so access to global metrics is possible. Also, Facebook users were found to be more likely to have a large number of close social connections [58], which is relevant to the problematic of this work.

The aforementioned factors contributed to the choice of Facebook and its Graph API as the social network data providers for this work.

4.1.2 Mobile Application Platform

The mobile application market is fragmented into a multitude of different platforms that imply the use of a vendor-specific API [1].

Among the major players, one can count with Google's Android [62], Apple's iOS [63], Blackberry Operating System (OS) [64] and Windows Phone [65].

In order to bridge this issue, multiple mobile OS encompassing solutions have been developed. Apache Cordova and Appcelerator's Titanium Mobile [66] are among them.

This is achieved by allowing the developer to use ubiquous software development technologies, namely the Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript. By employing these tools and by taking advantage of a uniform multi-platform Javascript API, mobile software development reaches a state where application portability becomes feasible. Through the use of a platform independent API, these solutions also expose phone sensor access to the developer [67] [68].

Apache Cordova generated applications can cover iOS, Android, Blackberry and Windows Phone, and others. Appcelerator's Titanium Mobile supports iOS, Android, among others [68].

Apache Cordova provides a wrapper for web applications. Titanium Mobile, on the other hand, requires a high amount of platform specific knowledge [68]. Moreover, while Cordova enforces no development tools, Titanium requires its users to use a specific Integrated Development Environment (IDE) for application building processes [67] [68].

By taking into account, the criteria of system portability, the choice of technology had to relay in one of the bridging options. Due to its flexible and less restraining nature the technology choice was place upon Apache Cordova. Furthermore, there's the fact that it serves more OS, allowing for a wider system spread. Another considered factor were the benefits reaped from Cordova's flexibility, i.e. mobile applications can be executed as simple web applications in a non-mobile environment. Consequences of this are added value presented when performing system-wide validation tests, where application deployment might be hindered by both timing and mobile device availability restrictions.

Due to the HTML nature of the mobile application, a technology for proper Document Object Model (DOM) manipulation in a dynamic application environment was required. Also, in light

of the Single Page Architecture (SPA), that is associated with the selected mobile application family of frameworks, an Asynchronous Javascript and XML (Ajax) capable library proved to be necessary.

For this purpose, both jQuery [69] and Zepto.js were considered. These libraries offer their users HTML DOM traversal and manipulation, and Ajax support. jQuery offers a superset of Zepto's functionality in an monolithic fashion. Zepto, on the other hand, aims to be a lightweight modular library that ultimately offers a minimalistic version of jQuery, i.e. it only exposes the features desired by the application developer. Consequently, thanks to its modularity concerns and due to associated relative performance improvements that originate from the application environment (i.e. the browser) having to load a smaller library, Zepto.js was chosen.

Once a web User Interface (UI) gets non-trivial and possesses a set of behaviours that can overlap, code maintainability becomes more expensive in terms of time consumption. Knockout.js is a Model-View-ViewModel (MVVM) framework that features declarative bindings, automatic UI refresh and dependency tracking. Declarative bindings allow the specification of the mapping between DOM elements and model values in a declarative approach in the application's HTML. The automatic UI refresh feature permits the abstraction of model value changes and consequent need for DOM updating, i.e. when a value changes, it is automatically updated to its latest value on the application's UI. Dependency tracking allows the definition of aggregated values that depend on the model, which are also automatically recomputed on model value change. The added flexibility and reduced codebase redundancy achieved by the use of this library accounts for the potential issue of investing a disproportionate amount of time in application UI changes and maintenance.

A further note on the applicational difference between Zepto.js and Knockout is that while one provides low-level DOM operations (Zepto.js), the other allows the high-level declarative definition of a user interface and its automatic mapping to model data (Knockout).

4.1.3 Programming Language

To target system correction, the initial concern of this work was to pursue development in a type safe programming language. Reasons for this were the runtime added benefits, where program state will not become undefined and lead to unexpected behaviour [70]. The programming language of choice was Scala [71]. Also, early versions of this work were targeted for the Android platform and, as such, the use of Scala was a viable decision.

From the inherent data unavailability of system's that share an application with this work (i.e. inference over a hard-to-obtain and potentially unrepresentative set of data, where the generalisation of inference results is severely limited), there came the requirement for programming language portability, which is by itself complementary to the overall system requirement.

As a Java Virtual Machine (JVM) language, Scala enjoys the capability of running in a multi-

tude of devices, while providing seamless access to existing Java libraries.

Nonetheless, the de facto portable language, when taking into account the mobile application scenario, is Javascript [67] [68]. Also, this choice of language is naturally compatible with the use of the Javascript Object Notation (JSON), which is a subset of it [72]. This data interchange format brings considerable advantages on the communication and data mapping schemes among the different parts of the system, thanks to its design goals, which aimed for it to be minimal, portable and textual [72].

Another advantage of this choice is the flexibility to move the logical blocks of the system among its different physical components. This is possible due to the high portability of the resulting codebase. This portability is boosted by the use of Browserify [73], a tool that enables the mapping of Node.js-specific libraries to general browser-friendly Javascript libraries which, in the case of this work, operate in the Apache Cordova wrapped web application that runs on the mobile sensing devices.

Since the totality of the system is implemented in Javascript, it is employed in scientific computation in the data analysis module. Benchmarks [74] show for numerical operations it outperforms languages, such as R, Octave and MATLAB, proving not be an hindrance in overall system performance.

4.1.4 System Communication

Regarding media access, both communications based on cellular technologies (e.g. GPRS) and the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards are possible, depending on the sensing device's capabilities.

The only requirement in system communication is the use of Internet Protocol (IP) based technologies.

Communications between sensing devices and their data sources are based upon a REST web-service interface, with its associated use of HTTP in the application layer, the Transmission Control Protocol (TCP) in the transport layer and IP in the network layer. This is a consequence of the need to comply to technologies compatible with the data provider's APIs. As for security, confidential and authenticated communications take place with resort to the Hypertext Transfer Protocol Secure (HTTPS) Transport Layer Security (TLS), as enforced by the used APIs. Also, the system conforms to OAuth, as it is, presently, the authentication framework used by the majority of social data sources, namely Facebook [75].

Resulting from the real-time requirements of system communication and to the web-based nature of the mobile application codebase, a solution aligning both aspects had to be chosen. For the communication between sampling devices and the data analysis module of the system, a WebSocket-based solution was deemed beneficial.

The WebSocket protocol enables two-way communication between a client in a controlled en-

vironment to a remote host that has opted-in to communicate with that client. Its security model is the origin-based security model commonly used by web browsers. This protocol is layered over TCP and it provides a mechanism for browser-based applications that require two-way communication with servers that do not rely on opening multiple HTTP connections. Naturally, delivery guarantees are covered by TCP. It was designed to coexist in HTTP environments. It enables an alternative to HTTP polling for bidirectional communications from a webpage and a remote server [76]. It is relevant to note that due to the nature of the mobile development platform that was chosen, the mobile user applications are equivalent to web applications and, as such, web pages are an inherent part of its architecture.

Given the programming language selected for the system presented in this work and its WebSocket support, the technology decision laid on Socket.IO. This technology constitutes a lightweight protocol that sits on top of HTTP. Among its aims are the goal to make real-time applications possible in every browser and mobile device, mitigating the differences between the different transport mechanisms, while supporting older user agents.

Among its features, the following should be referred [77].

- Disconnection detection through heartbeats.
- Optional acknowledgements.
- Reconnection support with buffering.
- Support for multiple sockets under the same connection, via client partitioning.

Client socket disconnection can also be detected through the use of heartbeat messages. For a Socket.IO connection to be established, an HTTP handshake is performed.If it is successful, a session id (given for the transport protocol to open connections), the number of seconds for an *heartbeat timeout* (time within which an heartbeat message is expected by the server) and the number of seconds for a *close timeout* (time after which, when a transport connection is closed, the client socket is considered disconnected if the connection is not reopened) are received by the connecting client. It should be noted that these values are shared by the client and server and, as such, both can act in accordance to them. Afterwards, the socket is deemed connected and the transport is signalled to open the connection. This parameter agreement allows for detection of connection loss in Socket.IO-based systems, which mostly benefits mobile devices scenarios or networks with debilitated connectivity.

Optional acknowledgements may be setup as callbacks to sent messages. In this way, it is possible for system state to evolve upon the execution of the callback on the receiving side of the communication.

Socket.IO operates by default in a non-volatile mode, i.e. in cases where the use of the transport connection is not possible, messages are queued. This happens until either the connection gets

re-established, and they can be sent in a batch, or the associated socket gets closed and they are dropped. If the connection is not restarted within the *close timeout* the socket of the other counterpart is presumed disconnected. From this moment, a client might reconnect its socket, which will require a new HTTP handshake.

Socket.IO's partitioning features enable the assignment of a proxy for a set of clients, that can be hidden behind an alias. Requests with a given namespace are intercepted an forwarded to the associated sockets. Thus, it is possible to filter client communication according to certain criteria. This creates the possibility to abstract the communication with the said clients, which alongside the other features makes room for seamless communication.

In the cases where the WebSocket protocol is not supported by the browser running the application, Socket.IO has several fallback transport protocols to ensure proper graceful degradation [78]. These concerns are aligned with the portability desired for this work.

Due to its coupling with HTTP, Socket.IO's communication security is based on HTTPS. TLS is the underlying protocol used in the system. By setting up a connection towards an HTTPS URL, a secure channel is setup, provided that targeted server fits the requirements.

The server needs to provide a signed X.509 certificate, which provides the means to authenticate the server (its public key), provided that is signed by a Certification Authority (CA) belonging to a trusted Public Key Infrastructure (PKI). This requirement transits from the root certificates trusted a priori by the client's user agent in an hierarchical way [79]. From the practical issues of getting acquiring a CA-signed certificate, in this work the used certificate is self-signed. This results in the loss of the server authentication guarantee. While this is not a meaningful drawback in a system prototype, proper certificates should be set up for real deployments. The system certificate was generated with OpenSSL.

Once setup, the HTTPS connection also provides two-way encryption, which allows for communication confidentiality [79]. In a deployed system, due to the sensitivity of system data, *perfect forward secrecy* should be enabled, i.e. with this guarantee it is assured that from the long-term server private key it is not possible to generate the short term session key, which is used to establish the communication channel between client and server. Thus, halting communication attempts to eavesdrop future conversation in case the server private key is compromised.

Communication with the social network platform is comprised of Ajax requests which result in HTTP GET requests to its exposed REST API. Due to the inclusion of Q, these requests are in fact asynchronous. Every request is wrapped to immediately return a promise, which will hold a value once the associated request is complete. Promises can be chained so that control flows that depend upon request completion are only executed when the needed value is available. These practices enable the creation of an asynchronous programming model, while increasing the readability of the code base in comparison to the callback-based solution that is customary in Javascript [80].

Due to the flexible nature of browser-side Ajax communication, additional security restrictions

are enabled in order to ensure system robustness. In regular client-side Javascript systems a policy of same-origin access exists. This policy results in the restriction of the requests that are allowed in the application. For instance, code running on different pages that are running concurrently, can only communicate with pages that were accessed using the same domain, port and protocol. Otherwise, a given web page would be allowed to modify sensitive data on another page freely. When cross-domain communication is desirable, mechanisms exist to do so, such as the Cross-Origin Resource Sharing (CORS). In the context of Apache Cordova, the analogue to this is domain white-listing [81].

As a result of the chosen communication technologies and the programming model they postulate, all system communications are asynchronous.

4.1.5 User Privacy

The authentication and authorization mechanism of the system clients is inherently coupled with its used data sources. Consequently, so that the required data is accessible, it is a necessary condition that users are able to authenticate on the data source platform and authorize the system to access the data it needs. By offloading these concerns to Facebook, unnecessary system resource load is averted.

In the case of Facebook, the authorization system is supported by OAuth 2.0.

To understand OAuth [82], the following terminology is relevant:

- The *resource owner* is an entity that is capable of granting access to a protected resource.
- The *resource server* hosts the protected resources and is capable of returning the resources when given valid access tokens.
- The *client* is the entity that intends to access the protected resources.
- The *authorization server* issues access tokens to the client after authenticating the resource owner with success and obtaining authorization to access the protected resources.

In this context, a user of the system is the *resource owner*, the *resource server* is the Facebook server holding the data the system requires, the system is the *client* and the *authorization server* is Facebook.

A more specific instantiation of this process would be the request of social network data by the system for one of its user. Firstly, the user logs in the mobile application with its Facebook credentials. Assuming a successful login and that it is the user's first time using the application, which implies no permissions persisted by Facebook, an authorization request for the system to be granted access to the user data is required. To take part in the system, the user will have to comply to this request (on subsequent logins, request confirmations are no longer necessary, as permission confirmations are stored by Facebook). This ensures that the user is made aware and

controls the kind of data the system has access to. After following these steps, the user is now capable of system participation.

If the would-be system user would have failed the login attempt, then taking part in the system would not be a possibility.

Also, considering the fact that system adoption can be limited due to user trust and its underlying privacy concerns, user anonymity is enforced throughout the system. In this way, if a data leak would occur, tracing data origin would require complex analysis methods.

For anonymizing user identification, a cryptographic hash algorithm is used. Encryption methods would be unsuitable as they are reversible in essence. Regular hash functions (e.g. SHA-2) aim to calculate a large amount of hashes in the shortest amount of time. This property is in synergy with ensuring data integrity. However, for storing sensitive information (e.g. passwords), the alignment of this same desire for speed with constant hardware improvements enables the construction of systems, aimed at cracking these hashes, with ever increasing computational power. One way to tackle this is the use of adaptive hashing algorithms, where the computational cost of this operation can be parametrically increased (by a work factor) to take into account the performance of current hardware [83] [84].

Well-known options are bcrypt [83] and the Password-Based Key Derivation Function 2 (PBKDF2) [85]. Bcrypt is more robust than PBKDF2 for most passphrases. Nonetheless, bcrypt is unable to use passphrases with a size over 55 characters [84]. Given the application of this methods, the transformation of Facebook user identifications (which are encoded with 64 bits [86]), the maximum pass-phrase size is well-above the limit. As such, PBKDF2 was chosen. Scrypt was another option, but the added of property of being able to resist hardware-accelerated attacks was not deemed essential for the system [84]. Another factor in the decision was the existence of a validated Javascript implementation of this algorithm, Stanford's SJCL [87]. PBKDF2 SJCL's implementation needs to be provided with a work factor and a salt. The default parametrisation uses the Hash Message Authentication Code (HMAC) Secure Hash Function (SHA) 256 as its pseudorandom function. The work factor should be re-tweaked and increased every year to keep up with Moore's law. The salt is a cryptographically-strong random value, which prevents the generated output from being equal for identical credentials. Also, it increases the entropy passed to the pseudorandom function, not depending on passphrase complexity, resulting in the infeasibility of pre-computed lookup and time-based attacks.

It should be added that this process constitutes a pseudo-anonymization transform, as the remainder of user associated data has relational ties to other users and can, thus, be subjected to data mining methods. These methods when cross-matched with public Facebook data (user identification is public) can yield the results needed to identify a user. Besides this limitations this process will be referred as user anonymization (as opposed to pseudo-anonymization).

4.1.6 Data Processing

For the creation of this system component, which comprises a network model for the underlying population network and the required analysis tools, a prototype was created rather than using an existing tool. The reason behind this choice is the high cost of learning how to efficiently customize existing tools added to the fact that very few exist that would be directly applicable to the analysis problem at hand. The added complexity associated with the much broader problem scope of other solutions it targets would hinder the development process. Examples of such solutions are EpiSimS [42] and EpiSimdemics [17].

Due to the scalability and real-time requirements of this work, the module responsible for the analysis of the sensed data needed to run on a scalable network platform. One technology that would align in the desired way with the set of the other chosen technologies is Node.js.

Node.js is built on Chrome's JavaScript runtime (V8) and aims for supporting scalable network applications. It uses an event-driven, non-blocking I/O model that enables it to be lightweight and efficient, adequate for data-intensive real-time applications that run across distributed devices [48]. Also, it is supported by the Node Package Manager (NPM), a package manager system that when used in synergy with Grunt [88] simplifies the development process of building and configuring the dependencies of a Javascript development project. Especially, in the case of works that are comprised of multiple applications that share dependencies. These factors weigthed upon and ultimately led the choice of Node.js to as the network server for the system.

For the choice of data analysis library, methods for linear algebra, and statistical and numerical analysis were required. In conformance with the technological alignment that guided previous decisions, Javascript libraries were selected. Given the relative diminute diversity in terms of mature Javascript mathematical analysis-oriented libraries, the choice was directioned towards Numeric.js.

A Javascript LAPACK [89] solution was considered, but its Node.js bindings - node-lapack [90] - proved to expose an insufficient number of functionalities, as it lacked methods for determining matrix eigenvalues.

Numeric.js is a library whose goal is to obtain the best possible performance for a Javascript program targeting numerical analysis [51]. Benchmarks results plotting the library against its alternatives validated this choice [91].

4.2 System

This section introduces the distribution of modules over system components and their physical and communication architecture.

It is followed by the discussion of system modules per se. This discussion elaborates on the techniques used to make the system objectives possible.

4.2.1 System Components

The system is composed by two distinct physical components: the mobile sensing devices (i.e. its clients) and the data processing backend (i.e. its server). The different logical modules or system concerns are distributed among them.

The data source of the system is a social network platform, i.e. Facebook.

All communications, except the ones with the social network platform (which are achieved with Ajax requests), recur to Socket.IO. It is configured in its non-volatile mode and the values for the heartbeat and contact timeout are set to the default values. Message confirmations are implemented as Socket.IO optional acknowledgements.

The exchanged Socket.IO messages trigger events on system components, resulting in state transitions. As events trigger function calls. By transitivity messages result in function calls. This process is exposed in the state machines that follow. Each component reflects its state machine, meaning that state transitions can be ascertained in its interface, i.e. client applications exhibit their state.

Services are timed for all the presented state machines. This implies that these machines have to describe the output state for a given transition starting from a previous state, but also the time in which the system is expected to do so. This results in an inherent dependency on Socket.IO timeouts.

Finally, it should be noted that message diagrams reflect the asynchronous nature of the system in their communications.

4.2.1.A Module distribution

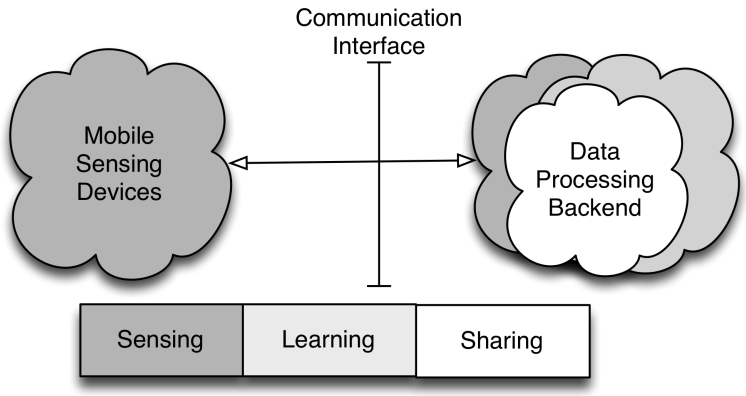

Figure 4.1: Module Distribution

An overview of the distribution of the three system layers and associated modules is presented in Figure 4.1.

The acquired raw data is originated in the *sensing* layer, located the mobile sensing devices,

and is coupled to a client. Since there are distinct data modalities they are regarded differently in terms of *data sampling*.

Social network data is acquired from social network platforms. It can, however, poorly reflect the relation between actors. For instance, two actors that have a relation in the platform may never meet or be likely to encounter each other, which is the kind of relation the model intends to emulate. Conversely, people that have no relation in a the platform, may meet on a daily basis, especially if they have the same area of residence. Thus, social network platforms are inherently incomplete in this regard.

As a response to this, contact network data could be obtained with resort to the sensing devices. In this way, individuals close to the sensing device to be detected through a short-range communication technology (e.g. Bluetooth) and used as sampling input. However, this solution would add a new level of complexity to data acquisition and its resulting data could not be representative enough to be employed in the model.

Another approach would be to gather location data with resort to GPS. Nonetheless, if it this data were to be considered in isolation it could become stale, not reflecting the real positioning of the user, and thus being useless to ascertain contacts. Location data can also be gathered from the platform. However, due to its high granularity it is potentially less accurate with regards to user contacts.

A solution to this is to gather meeting data instead of location data through the platform. Although, the granularity of this data is usually high, it has an added context. A significant number of platforms have features to associate user updates to other users, while indicating a time and location. With resort to this data, it is justifiable to presume there is a meeting between users at a certain location and time. Although there is an error associated with the assumption of time and location, in this approach the meeting-related problem is no longer present.

As a result, both network and meeting data are sourced from a social network platform.

Regarding platform data sourcing, it could either occur in the sensing devices or in a backend. It was deemed preferable for sampling to take place in the sensing devices, both for keeping user credentials in a trustworthy medium and for anonymizing their data. In a backend solution, credentials would have to be managed safely by the system itself. This approach leads to more data being transmitted, imposing a potential strain in communication and sampling constraints.

Nonetheless, it does not constitute a real sampling issue, as data is already constantly stored on the social network platform. Also, prior to any round of *data analysis*, the mobile sensing devices have to request platform data and offload it to the backend.

Consequently, *data sampling* occurs entirely on the mobile sensing devices.

The data integrity component of the *data filtering* module is achieved on the backend, as only during the aggregation process it is possible to assess which data is likely to be erroneous and thus dispensable. For instance, two users that have equivalent contact data about each other are unable

to filter out this information by themselves.

The data anonymization part takes place on the mobile sensing devices, as they constitute a privileged position where is it possible to plug the leak of user identity. If this data were to be forwarded to the network, the likelihood of it getting exposed would increase significantly.

The *epidemic prediction* module considers only infectious agent data for which there are varied and detailed information sources, e.g. the influenza model [32]. Due to its complexity, the computation associated to the *Learning* layer takes place in a backend separated from the sensing devices.

The *Sharing* layer takes place in the backend, as by placing it there, relevant information can be easily acquired as output of the *Learning* layer. It can be transmitted to all the concerned individuals, as global knowledge required for client communication is already present in the system, while preserving user privacy (only the communication channel is kept during a round). If this layer were placed in the sensing devices, each mobile device would have to know the location of the other individuals, which would lead to an extra level of complexity and the potential violation of privacy restrictions.

For these choices, the criteria of ensuring there would be a minimal load on the mobile sensing devices for resource management reasons, as they are susceptible to battery drain and data sampling restrictions, was employed. By splitting the computational requirements and placing the demand on the backend, this impact can be minimised.

4.2.1.B Mobile sensing devices

The mobile sensing devices are responsible for sampling data from the data source and for forwarding it to the data processing backend.

Figure 4.2 represents the state machine for the mobile client applications.

The client starts in the *initiated* state after the mobile device is ready. This is known through a device ready event that is sent to the application upon Apache Cordova successful initialisation.

A client may try to login to Facebook and while it is in the *awaiting login* state, it may receive the Facebook OAuth 2.0 access token, which represents a successful login. Any other reply or the lack of it, represents a failed login.

After the client is logged in, it has to associate itself with the data processing server. That is achieved with an associate message sent to the server.

While the client is in the *awaiting associate* state, it may succeed if it receives a message confirmation from the server or timeout. There are two kinds of timeouts and they both result from the Socket.IO communication library. If a heartbeat timeout occurs, as the request is queued and the protocol will attempt to re-establish the connection to resend the message, the state is maintained. If a close timeout occurs, it implies that the connection is lost and that its socket was closed. As such, a new connection will have to be setup and a new message sent.

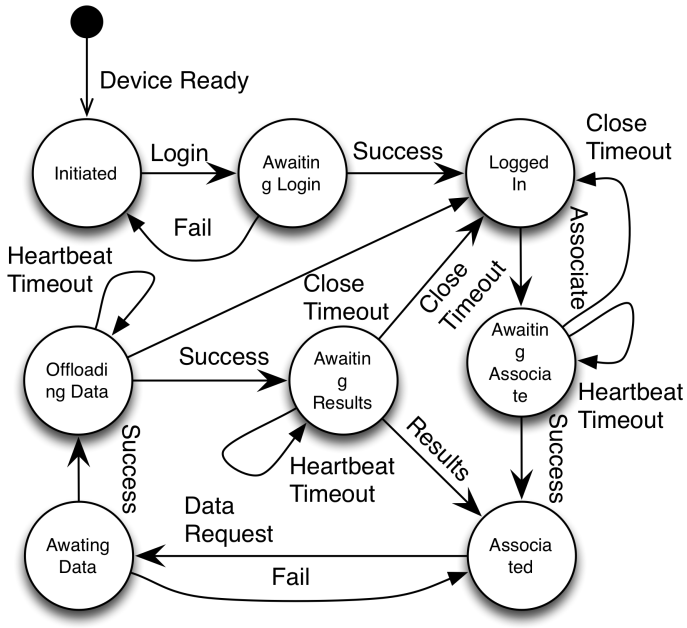

Figure 4.2: System client state machine

Upon a successful association process, the client will be *associated*. From here, it may receive a data request from the server, which will lead to sampling. A client may only hold data between server data requests. Upon this request the client attempts to re-sample its data from Facebook and drops its previous state.

Now, the client will be *awaiting data*. The data may be either successfully loaded or the request may fail. Any response or absence of it that is not a message with the expected data format is considered a failure and the client will have to wait for the next data analysis round.

If the data is loaded with success the client will move to the *offloading data* state. Request success is assured upon server confirmation. Again both Socket.IO timeouts are possible. While a heartbeat timeout will lead to connection re-establishment retrials, a close timeout leads to the loss of the client connection socket on the server side and consequently to its disassociation. As no other server state is kept, the association process will have to start again.

The client will be *awaiting results*, until these are forwarded from the server. Heartbeat timeouts will lead to attempts to re-establish the connection. If that is not possible and a close timeout occurs, the association process will have to restart for the reasons referred before.

Client association exchanges are exemplified in Figure 4.3. Once clients are authenticated in the data source (i.e. Facebook), they can associate themselves with the data processing backend.

The *login* message contains the client identification and secret. Its reply is an access token that will authenticate further Facebook requests. The *associate* message contains anonymized client identification and its acknowledgement implies that the client communication channel is now part of the associated client partition of the server. A successful message of this kind is idempotent.

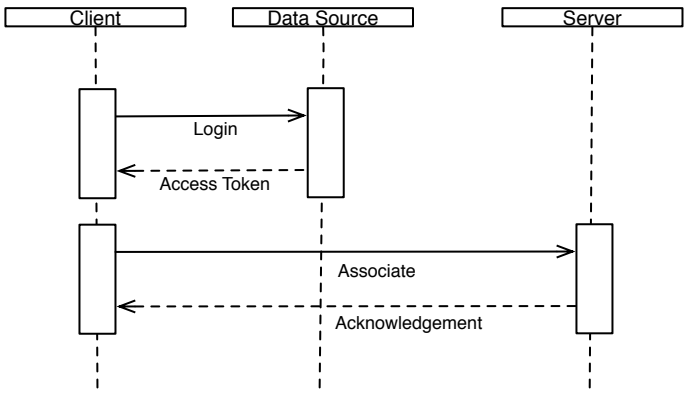

Figure 4.3: client association to the system

4.2.1.C Data processing backend

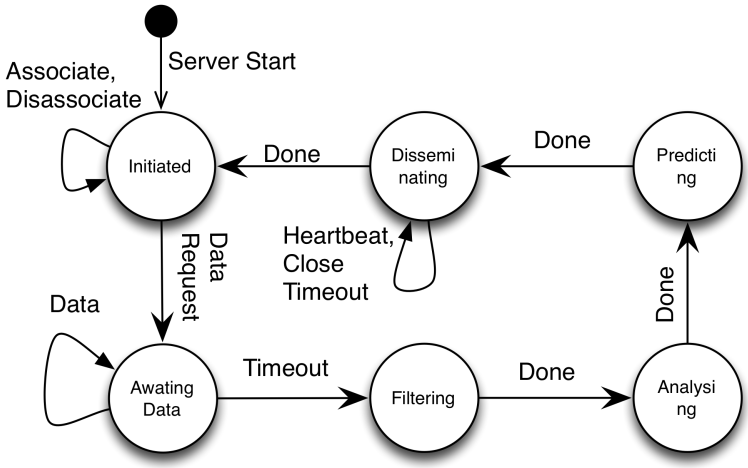

Figure 4.4: System server state machine

The data processing backend is charged with filtering and analysing the data received from the mobile sensing devices. The results obtained from this process are sent back to the client devices. Its state machine is represented in 4.4.

After the server is started, it will arrive at the *initiated* state.

From here, it may receive associate requests from clients. In the same, clients may get disconnected leading to disassociate transitions.

Association requests are not verified by the server, implying that clients for data analysis are not authenticated. This leads to a trade-off between user anonymity and system security. While the system might be vulnerable to external entities that might offload fake data or attempt to overload the server, a given client cannot be verified without having access to its credentials. Furthermore, a user should not be asked to authenticate more than once. To solve this issue an authentication

process between the clients and the server would have to be developed. Such approach would be based on the authentication of the mobile application, instead of the user itself.

Finally, given the requirement of user anonymity, authenticating a given user is unfeasible, leaving room for abuse with non-authentic data. Nonetheless, since data is sourced from a third-party, any alike system is susceptible to such vulnerabilities.

After the analysis is triggered (either intentionally or within some pre-defined periodicity), the server will broadcast a data request message to all its associated clients. This set of clients constitute the partition whose data will be used.

Afterwards, the server will arrive at the *awaiting data* state. In this state client associations are no longer possible, as the server will receive data messages from the client partition established previously. If a given client does not offload its data within a predefined time, it is excluded from the partition.

When this timeout occurs, the server will start the *filtering* data state, resulting in the removal of erroneous values from user data.

The *analysing* data state ensues, originating a merging process of data with different modalities. Infectious agent data is integrated in this state. This results, in the combination of network and contact data with resort to graph weighting techniques and its parametrisation with infectious agent data.

The *predicting* state follows. The merged data is inputted to the epidemic prediction algorithm. After it is concluded, results are generated.

The server will start the *disseminating* state, where it will forward the prediction results to the client partition. Upon heartbeat timeout, the connection will be re-established and the results resent. On close timeout, since it results in the dismissal of the only link that exists between the server and its client, the results for the analysis round will not reach the client. After this state is left, all sourced data present on the server is dropped. Both these behaviours are a consequence of client privacy requirements.

This sequence in Figure 4.5 clarifies the messages exchanges that lead to the generation of system results.

All associated mobile clients are targeted by the *data request* message, which results in a series of successive Facebook requests needed in order to commence the data analysis. This message contains no data. Afterwards, clients have access to the data that is required for sensing to be feasible.

The first one requests the user for permission for the system to use the data it requires. In this process, all the relevant user data is gathered, provided that the user has granted the client application the required Facebook permissions. The necessary extended permission is $read_stream$, which allows access to all of the user's interaction with other users in the platform [92].

The data gathered includes all users friends and their assessed contacts. Contacts are presumed

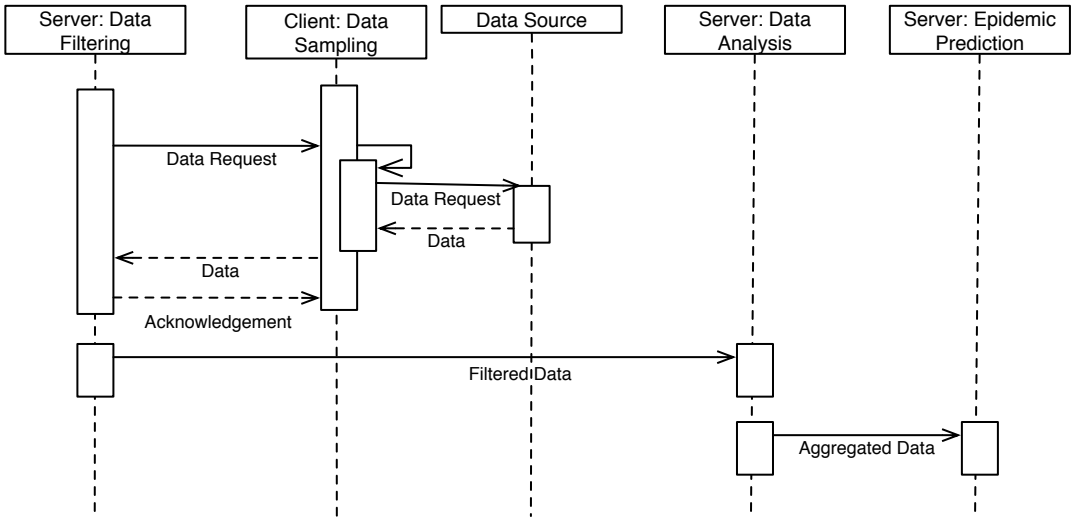

Figure 4.5: Data analysis and epidemic prediction

to have happened whenever there is a Facebook interaction, including at least another user and where a physical location is specified.

Each client replies with the data it acquired. If a given associated client (or set of clients) does not reply at this point, data analysis will be resumed, after a timeout, without considering that data. In this way, they will be excluded from the future communications that would result from the data request message. Upon receiving data with success, the server acknowledges its reception to the client.

The *filtered data* message contains the filtered data from the different sensing modalities and moves the system to the next state.

The *aggregated data* message contains the social contact contact network data, which results from the data merging processes, and the infectious agent data parameters.

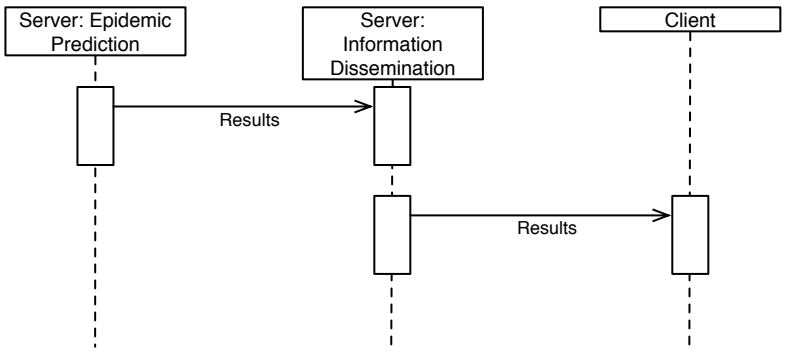

Figure 4.6: Data results reaching system users

The dissemination of system results is shown in Figure 4.6.

The *results* message contains the indication of the epidemic risk, represented in form of a dichotomous string whose value is either the presence of epidemic risk or its absence.

While the first *results* message is simply forwarded to the next module, the second one is broadcasted to every system client.

4.2.2 Modules

The implementation of system modules is described in regard to their placement in the overall solution architecture.

This section focuses on each of the architecture modules, of which the system layers (*Sense*, *Learn* and *Share*) are comprised. For each module, the different challenges related to the objectives of this work are mentioned and their completion commented upon, taking into account the chosen technologies, the assumptions made and the different options that were reviewed in the literature.

4.2.2.A Data Sampling

In the system *Sensing* layer, *data sampling* is made possible through the use of mobile sensing device applications that are coupled with each user. Due to their limited coverage, the acquired data is inherently incomplete. There are different strategies that may be employed to tackle this.

Firstly, network data sampling approaches, such as *ego-centric* sampling and the family of *link-tracing* methods can be attempted. *Link-tracing* methods can increase the network data covered by the system significantly. Alas, they come with a high computational cost [39]. For instance, *k-wave link-tracing* can roughly lead to $O(n^k)$ time for a single source actor. For *saturated link-tracing* the time complexity cannot be estimated, as $k = \infty$. Out of these methods, only *ego-centric* and *one-wave link tracing* remain reasonable in this aspect.

Secondly, inference methods may be applied to get a more accurate perspective on the acquired data. However, these methods attempt to infer missing relational data [39], which is not the case when the sampling unit that is lacking is the network node and not its links.

Thirdly, heuristic measures on the number of estimated relevant social network platform contacts could limit the required number of individuals to sample [58]. According to [61], the median value of friends in Facebook is 99. Nonetheless, these numbers are summary statistics of the global network and carry an approximation error for system users. For cases where a given user would not have the expected median number of friends, new sampling requests would have to be made, resulting in a higher sampling strain. Conversely, for cases where a user would have a number of friends higher than the expected value, some friend nodes would not get selected. This decision, even if done randomly, is subjected to errors that might impact the relevance of the gathered nodes.

The information that is most relevant for the system and the one that can impact infectious disease spreading is user contact data. The acquisition of more network data nodes will not result in more data of this kind, as the system is already theoretically capable of obtaining the totality

of the contact data existent in the social network platform.

In the light of these points, *ego-centric* sampling was considered to be the most adequate approach. While being subject to the same validation challenges as other methods, it yields objective friend connection results, it has a lower computational cost $O(n)$ per individual and all connections can be gathered from a single sampling request.

Finally, issues with missing data are mitigated by the connections observed in the acquired contact data. In this modality node relations for which there was no prior knowledge are sampled. These links also enable unobserved nodes to be detected. For instance, if a meeting involving a known node and a unknown node is detected, the system gains insight into missing node data.

Nonetheless, contact with strangers, i.e. nodes for which there is no connection in the data gathered for all modalities, is not accounted for. This constitutes a inherent limitation with the used sensing methods. To solve this issue, classes of more specialised acquisition processes would have to be developed, constituting a whole new problem area. Consequently, this work does not focus on such cases.

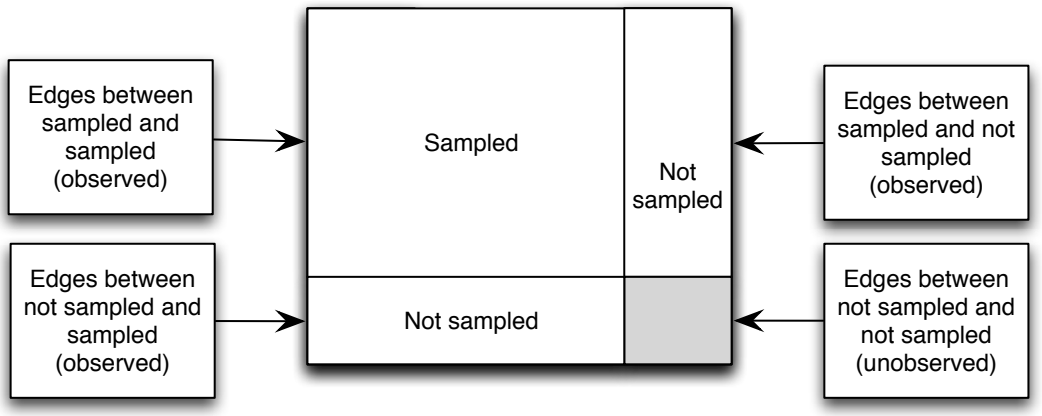

Figure 4.7: Edge data sampling for an undirected network

Figure 4.7, based on [39], illustrates the aforementioned concerns with node data sampling and the observation of edges or connections.

4.2.2.B Data Filtering

Part of the *Sensing* layer, *data filtering* has the goal of ensuring that unsuitable data does not get into the subsequent modules of the system. It functions as the safeguard before *data analysis*.

On the clients, it also functions as data anonymization module, ensuring that Facebook client identification does not reach the server in its plain text form.

Following the recommendations in [93], the parametrisation used for PBKDF2 was a work factor of 10000 and a salt size of 256 bits. SJCL's default pseudo-random function was used. The salt size should be at least as big as the size of the output of the pseudo-random function.

One drawback of this process is due to the fact that network and contact data, coming from different users, have to possess matching anonymized user identification. As such, it is necessary for different client applications to arrive at the same hash for data aggregation to be possible. This results in the need to have a common salt for all clients per analysis round.

This common salt is hardcoded in the client application code. If an identification hash is obtained and this salt is discovered, an attacker will be able to compute the anonymized identification. This has more severe implications when used in synergy with data mining methods and Facebook public user identification data, as if the identification of a single user is compromised, it is possible to determine the identification of users with whom relational ties exist.

As communication encryption is based on HTTPS, an approach to minimize this issue would be to build a step for distributing this salt to all associated clients. With this measure, for each analysis round a client would be granted the salt with which to parametrise PBKDF2. However, the server would already have the salt in its possession, so any attacker comprising it could in theory break the hashing. Furthermore, since there is no client authentication on the server, this solution is not viable, as any malicious client could get access to the round salt. Under these conditions, the present solution constitutes a compromise.

On the server, a part of this module's task is made possible through the use of programmatic checks. These verifications ensures that invalid values for expected data types are filtered. For instance, user contact data that is not array-typed is deemed invalid.

Due to the undirected nature of contact relations acquired for every user, sampling can result in duplicated contact data. The information to make this assessment is only available at this point in the server.

Users, for which there is no contact or network information, are let through as another sampled node referencing them might appear. The appearance of isolated actors is impossible in practice, as nodes are sampled through other nodes. Regardless, for the sake of correctness the implementation is able to cope with such cases.

In the final filtering stage, each actor is represented by a network node that includes a unique identifier, reflecting user anonymity. The same is verified for each of its social connections. The data for perceived contacts is also included in the resulting data.

Resulting data is transitioned to the *Learning* layer.

4.2.2.C Data Analysis

The first module in the *Learning* layer is responsible for *data analysis*. Its goal is to join the data from different modalities in a meaningful way and to output an epidemic model that can be used for the predictions of the next module in the chain.

A possible result of this aggregation process could be a social contact network with binary connections. However, it would constitute a simplistic vision of infectious disease propagation. A

more realistic network view, that can account for distinct forms of contact, can be provided by weighting network edges. Even if certain groups of individuals are topologically poorly connected, if the links between them have the highest weights, they become easily accessible by the infectious agent. Infectious agent access to pre-existing topological network clusters gets conditioned by edge weights, resulting in the creation of weight clusters [35].

This module takes advantage of these notions and encodes individual contacts in the weighting process applied to pre-existing network data. Thus, the sampled network data is extended with the notion of direct actor contact.

These methods consider the whole of the sampled network, i.e. data from all users. The possible lack of interconnections between the networks sampled for each user, arising from the non-existence of overlapping nodes in sampled user datasets, presents a challenge. This issue can be solved by detect this phenomena and by providing separate results for the users involved in such cases. With this approach, the obtained network metrics remain accurate as opposed to alternative solutions where sub-networks would be joined with a chosen edge weight. Another such solution would be for all users to be connected by default with a relatively low weight. This problematic is not approached in the current implementation, as the validation of end-to-end system execution was deemed more important than the resilience to this particular case, which would demand for further assessment complexity.

The operation process for this module is as follows. An actor name is the internal identifier used in data merging.

On the first stage the network and contact data are extracted for all actors that they reference, providing an universal mapping between an actor name and a matrix index i. This index refers to a given square matrix vector, where n is the number of the vectors and $i < n$.

On the next stage the network and contact data are transformed into adjacency matrices A and C. This is done using the mapping created earlier. As a result, both have the same dimension.

The next step introduces weights under the service of data merging. A_{ij} is a dichotomous variable from actor i to j, where 0 represents the absence of an edge and 1 its existence. As its upper bound is the number of contacts between i and j, C is restricted to $C_{ij} \in \mathbb{N}_0$. The rationale behind this is as follows. While contacts between two actors may increase over time, a social connection between them may either exist or not.

Taking matrix A, every non-zero entry is replaced by ω_{ij}, subjected to a weight bound $\beta_f \leq \omega_{ij} \leq 1$, creating the joint matrix W. W is the result of weighting matrix A with C.

$$\omega_{ij} = \beta_f^{\frac{C_n - C_{ij}}{C_n}} \tag{4.1}$$

The parametrisation for ω_{ij} is shown in Equation (4.1). β_f is the infectious agent effective contact rate. C_n is the total number of contacts for all nodes, while C_{ij} the number of contacts between nodes i and j and $C_{ij} \leq C_n$. Within the weight bound, edges that reflect at least a

contact have their value scaled by a parameter that increases with the number of contacts that edge represents, forming a relative contact measure. This is based on the assumption that more contacts lead to a higher chance of contagion. Conversely, actors that have zero contacts between them are attributed β_f, as their relative importance in infection propagation is considered lower.

The final stage scales the joint matrix. As weighting directly decreases the spectral value and epidemic threshold of A, W has to be scaled. The scaling process consists in applying the average network weight to the matrix, substituting every W_{ij} with $\frac{W_{ij}}{\overline{\omega}}$, resulting in the matrix W'.

This aggregate dataset constitutes, alongside with the infectious agent data, an epidemic model, where every sampled individual is explicitly represented. Its properties are described hereafter.

- All individuals are susceptible
- All model rates are constant
- Only horizontal transmission occurs
- There is no explicit demography
- There is no latent period for infection

It is assumed that all individuals that are part of the social contact network are equally susceptible to infection. The spread of the disease is uniform in this aspect.

Model rates are constant over time. This means that in the execution of an analysis round they remain constant.

Vector and vertical transmission are not part of the model. Only direct contagion (horizontal transmission) between individuals is possible. Vector transmission is related to indirect contact, i.e. the transmission of Malaria with mosquito bites. Vertical transmission occurs between mothers and their unborn children.

The death and birth of individuals is not taken into account, resulting in no explicit demography. This assumption holds for fast diseases like influenza. For diseases that have a slower personal rate of evolution, akin to HIV, tuberculosis and hepatitis c, this is not valid, as during this length of time the population demography does not remain unaltered [30].

An infected individual is immediately able to transfer the infectious agent to others. Thus, there is no latent period of infection.

For infectious agents that fits these assumptions, a given effective contact rate β and recovery rate δ are needed. For the influenza pandemic of 1918, these values are respectively 8.0 and 0.34 [94]. As this was the most complete analysis found, this data was used as a reference for proof of concept for data analysis.

This epidemic model is passed to the next module.

4.2.2.D Epidemic Prediction

Epidemic prediction is a module of the *Learning* layer. Its purpose is to apply the intelligent analysis algorithm to the epidemic model.

Implementation-wise, in conformance with methodology criteria and for scalability reasons, it was desired that this module would use memory parsimoniously and run quickly, while for extensibility reasons, its code should be relatively easy to read and modify.

It was necessary to find a process through which obtain viable data analysis.

Simulation-oriented strategies demand an extremely complex validation process. Furthermore they are potentially impossible to validate completely [8]. Researchers, who follow these approaches, attempt to validate their model by measuring the output of computationally expensive simulations against the public health statistics of an ongoing epidemic [20].

Alternatively, they may employ data from past epidemics, which compromises the generalisability of their results. Moreover, in the joint area of social network analysis and epidemiology there is an significant lack of data [8].

In response to the validation demands of epidemic simulation, an analytic network topology approach was selected. The main idea behind the predictive power of this module is the notion that the topological properties of a social contact network can be used to assess epidemic persistence [35] and, thus, its advent. While this approach provides a limited context, it provides accurate results for the data family in analysis.

The prediction algorithm consists in identifying the spectral radius (the maximum of the absolute of the eigenvalues of the adjacency matrix) of the social contact network.

For this purpose, Numeric.js possesses an eigenvalue calculation method, which may be invoked from a square matrix (i.e. a Javascript vector containing n vectors with size n). This feature returns all eigenvalues and eigenvectors for a symmetric matrix and it resorts to the QR algorithm [95].

However, for a single iteration step, its complexity is of the order of $O(n^3)$, which is prohibitively expensive for large matrices. Additionally, the QR algorithm computes all eigenvalues (and eventually all eigenvectors), which is rarely desirable for sparse matrices [96]. Given the reason behind this computational endeavour, the determination of the dominant eigenvalue, this is even less reasonable.

A computationally less expensive trace iteration algorithm may be employed.

$$\lambda_{dom} \approx \lim_{n \to +\infty} (Tr(A^n))^{\frac{1}{n}} \tag{4.2}$$

Equation (4.2) exposes the formulation of the said method [97], where n stands for the iteration number and Tr is the trace of the matrix A.

Nonetheless, this method is expensive. Its cost can be lessened by restricting the number of allowed iterations, provided that there is an approximation error that is suitable enough for the

area of application [98]. As such, this algorithm was implemented and used in conjunction with Numeric.js.

More sophisticated and effective iterative approaches, like the Arnoldi iteration can be tried to obtain a reduced error. Nonetheless, such would be achieved at the the expense of code re-usability and flexibility, which are deemed heavier criteria.

The obtained spectral radius and the infectious agent parameters are inserted into equation (2.2), which relates the epidemic threshold of the graph associated to the network and the rate of infection transmission and recovery.

With this process it is verified if there is a subset of actors that possess enough connections, of a relatively high weight, to constitute a weight cluster, and thus form a network bridge. If this subset is representative enough, the existence of network weight bottlenecks will be high enough for the successful spread of disease and the creation of an epidemic.

The equations yields a boolean value, which can be translated into network epidemic vulnerability, if false, or the lack of risk if true.

This value is forwarded to the next layer, the *Sharing* layer.

4.2.2.E Information Dissemination

In the *Sharing* layer, for *information dissemination*, i.e. the communication of the prediction results, two possible solutions were considered with the objective of providing users with information that is deemed useful.

- Personal notifications
- Community notifications

Personal notifications inherently imply knowledge about a given user's identity and data, which by itself compromises expected privacy requirements. Also, the impact of providing users with personal information regarding their susceptibility to an epidemic has an ephemeral value, given its probabilistic nature and the fact that this inference can evolve at any moment. Moreover, sharing such fine-grained user-oriented information in an epidemic context can potentially lead to short term consequences that could further alter the reliability of result validity, namely panic.

With this in mind, user community-level notifications were evaluated. Given that this work is targeted at epidemic prediction, the useful information at stake is the susceptibility of the community to a given epidemic. This metric can be reflected in a binary decision, i.e. either a risk exists or it does not. By leveraging this metric, notifications that target the user community become viable, as the information to spread is depersonalized and is, thus, equally useful to all of the users.

The result received from the *Epidemic prediction* module is translated into a notification that maps its value into information that is interpretable by users. More specifically, either of the "No

risk" or "Epidemic risk" strings may result from the input of the previous module.

This information is communicated to all of the users who are present in the client partition.

5

Result Assessment

5.1 Methodology

The objective of result assessment is the validation of the work produced and the determination of adequate system improvements.

For this, this section aims at validating the outcomes of this work within various of metrics. It is divided in test areas that cover different analysis aspects.

Results are gathered through the execution of a set of experiments that were defined for each area. They are stored, taking into account the expected value and standard deviation for the targeted sample.

Collected results are subjected to statistical analysis, by placing the gathered relevant summary statistics in a t-student distribution confidence interval.

In the end, they are presented in a chart format, so that their interpretation is clearer to the reader.

The general assessment methodology focuses on testing, and if possible validating, the different parts of the system individually and then progressively integrating more complexity.

Figure 5.1 illustrates the general result assessment plan.

Firstly, the system *Learning* layer is validated in terms of correction with randomly generated data. Afterwards, a system-wide analysis is made, using the same data, to evaluate its performance and scalability in terms of user base size, while running on the same local machine. These sets of experiments aim to validate the functionalities of the theoretical models.

Practical coverage modalities assess applicability of the system to a domain that is closer to the problematic. Deployment testing analyses the global architecture in a real mobile scenario

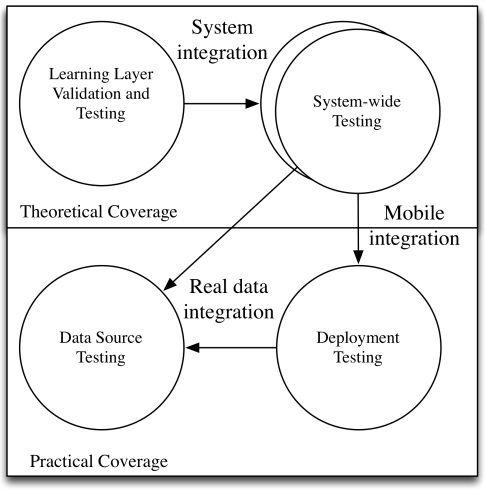

Figure 5.1: Result Assessment plan

with the data used previously. Data source testing targets the system as a whole with real data for both local and mobile scenarios. Setting up the system to cover real data has the requirement of real users. Due to the sensitive nature of the system, the cost of deploying the system on a significant scale is high. Consequently, for validating the solution with a realistic data source context, this testing is performed with resort to Facebook's Test User API [59]. The combination of these modalities results in a global evaluation of performance and scalability in a real context.

The detailed experiment methodology is described under each test area subsection.

The following tools were used in this process.

- igraph [99]
- Grammar of Graphics Plot 2 (ggplot2) [100]
- Data-Driven Documents (D3) [101]

The decision criteria were based in the need for specific functionalities and in overall system integration concerns. These will be further elaborated upon in the next sections.

5.1.1 Data Gathering

An extensive data search, for data that could be used in the study comparison, was conducted through online platforms, such as the Centers for Disease Control (CDC) and the World Health Organization (WHO) [20] [19]. Its target was the data which would account for social relations and their impact on the spread of an infectious disease. It was concluded that there is lack of accessible data in the required formats, which fits the premise in [8].

Also, data extraction from a chosen social network platform would require mass application distribution for scalability and model testing, as data access is limited by user permissions, and would ultimately not provide any means of inferring infection spread.

As a result of these factors, it was necessary to find a solution to generate data that would fit system requirements. In this endeavour, it was observed that scale-free networks provide a sufficiently adequate fit for social networks [36] [102].

Consequently, the question was shifted to the process of generating network data that would follow such a degree distribution. This data would then be parametrised with contact data.

The igraph library was found to possess a functionality for the creation of scale-free networks. Given R's [103] background as statistical analysis programming language, its bindings for this library were chosen.

The igraph network data generation process is the result of a simulation process. It uses an implementation of the Barabási-Albert model, which is a simple stochastic algorithm for building a graph with scale-free properties. Its outputs follow the scale-free law (2.5) with $\gamma = 3$, which produces networks adequate to the area of epidemiology.

One limitation of this method is that it is not possible to control the number of edges of the resulting network, restricting the execution of test scenarios depending on the variation of node connections.

The default parametrisation was chosen and changed to produce undirected graphs [104]. In this way, it was made possible to obtain methods for the production of networks that fit the desired model.

For better integration with the generated network data, the required contact data was also produced stochastically. R's uniform sampling standard library [105] was used.

An algorithm for outputting sets of contacts with a controlled size was developed. Its parameters are a set of network node reference values and the number of desired extractions. It accounts for reposition, as actors may meet each other more than once.

As an actor cannot meet itself, each set was identified after a node reference, which was excluded a priori from the sampling process that produced it. To make this exclusion possible, the set generation process was configured to exclude specific nodes by re-sampling when they were extracted.

As each contact set is dependent on the network data that provided its node references, contact sets were persisted for for each node of the produced networks.

Finally, there was the need to account for distinct contact distributions on network nodes. A step for setting up a distribution scheme was added to the contact generation process. A pool of n contacts can be uniformly distributed to m nodes, provided that $m \leq n$, $n = k \times m$ and $k, n, m \in \mathbb{N}$. Each node receives $\frac{n}{m}$ contacts. An egocentric distribution assigns the whole pool to a given node.

All generated data was exported to the JSON format, so that it could be seamlessly integrated with the remainder of the system. This JSON data is formatted to be equivalent to data extracted from the social network platform (i.e. Facebook), as that is the input format expected by the

system.

5.1.2 Result Analysis

The results from system analysis have to be contextualized in a population scope. As such, their mean and standard deviation are obtained. Since both the population mean and standard deviation are unknown, these are fit into a t-student Confidence Interval (CI). In this context, an estimated range of values where the unknown population parameter is likely to be included is given.

To determine a CI for for the population mean based on the sample mean and sample standard deviation, the score for the t distribution t^*_{n-1}, for an upper critical value, with $n-1$ degrees of freedom, where n is the sample size, has to be found.

$$p = \frac{(1-C)}{2} \tag{5.1}$$

Equation (5.1) defines the upper critical value p, where C is the desired confidence level.

The t score is usually tabled for a given p and $n-1$ [106]. In this work, it is computed with R [107].

$$SEM = \frac{s}{\sqrt{n}} \tag{5.2}$$

The estimated standard deviation for the sample mean (Standard Error of the Mean (SEM)) has to be calculated with resort to equation (5.2), where s is the sample standard deviation and n the sample size.

$$[\overline{X} - t^* SEM, \overline{X} + t^* SEM] \tag{5.3}$$

The final confidence interval can be given by (5.3), where $\overline{X}$ is the sample mean.

5.1.3 Test Platform

For the system evaluation phase of the testing and validation process, a testing platform was developed. This creation was the result of the concerns regarding system flexibility and portability. It constitutes a web application aimed at enabling smooth system configuration and testing.

The platform can be configured to launch automated tests that can gather system output artefacts, such as performance metrics and data analysis results, enabling the investigation of validation metrics.

Its communication capabilities allow the collection of test results from both the mobile clients and the data processing backend, in conformance to a test configuration. These are fully supported on the Socket.IO technology.

Through the configuration interface, all system architecture modules are configurable for added test flexibility. For instance, *epidemic prediction* parameters, such as the *recovery rate* can be set remotely from the platform's web interface.

Also, one is capable of launching and parametrising a chosen number of mobile clients. For testing purposes, client credentials can be inserted into the launched application upon load, resulting in the automation of login processes.

Data sources can be selected. Either the generated data (persisted in the filesystem) or Facebook data (reachable through the network) can be used as data samples.

A typical use case would be the launch of a certain number of clients with a given identity, social network and known contacts. After the parametrisation process is concluded, normal system behaviour ensues.

A plotting tool to visualize the metrics to assess was required. Given the integration possibilities with this platform, the use of a plotting technology that would allow interoperability with the generated data was considered an important factor. R's ggplot2 was fit for this requirement. Its agile plotting approach with a variety of metrics and configuration options constituted the added value that lead to its choice.

A more seamless testing process could be achieved by integrating a real-time data visualization technology into the testing platform. In this modality, D3 was selected thanks to its flexible nature in conceiving customisable charts. As a Javascript library, it was naturally suited for end-to-end integration with the remainder of the system, allowing the generation of automatised real-time visualizations. Consequently, during system debugging and testing, one could quickly assess the social contact networks the system had access to, along with aggregated metrics that were computed upon it. One example of a computed metric is the attribution of colour to nodes with regard to their degree and link weight, enabling a real-time assessment of a network's properties.

With these tools, the platform's communication capabilities enabled the pipelining of test outputs to both data charts and statistical analysis methods. This setup greatly contributed to the process of assessing system results.

5.2 Results

5.2.1 Learning Layer Validation and Testing

The purpose of this section is to assess the general model behaviour with scenarios that can exclude model and implementation shortcomings. Validation is done through the comparison of test outputs with the results of other authors.

The main metric to determine is the proper calculation, by the system, of the epidemic threshold λ_c for a given network. Its correct evolution according to other model parameters, such as network size and number of contacts between its actors, is also analysed.

To achieve this, the *data analysis* and *epidemic prediction* modules are used in the aggregation

of user contact and network data into a social contact network and subsequent epidemic threshold computations. The data to do so was generated randomly, as described in section 5.1.1.

Social contact networks may result from contact data with zero contacts between its actors. In these cases, the network will still possess an epidemic threshold, as it results from the relations present in network data (i.e. relations analogous to a Facebook friendship).

For the experiments in this section, 50 network samples of different sizes were generated. Their sizes S are included in $\{4, 8, 16, 32, 64, 128, 256\}$.

So that a representative number of contacts was chosen, for each of these network sizes S_i, a contact range C_i^R was calculated. It has the form $[0, S_i]$, where $\forall i \in C^R, i \in \mathbb{N}_0$.

S_i	C_i
4	$\{0, 1, 2, 3, 4\}$
8	$\{0, 2, 4, 6, 8\}$
16	$\{0, 4, 8, 12, 16\}$
32	$\{0, 8, 16, 24, 32\}$
64	$\{0, 16, 32, 48, 64\}$
128	$\{0, 32, 64, 96, 128\}$
256	$\{0, 64, 128, 192, 256\}$
S	$\{0, 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 64, 96, 128, 192, 256\}$

Table 5.1: Contacts per network size and their set union

For each of this intervals, its quartiles were extracted. These were used to generate the set of contact sample sizes C_i, which follows the expression $\{0 \cdot S_i, 0.25 \cdot S_i, 0.5 \cdot S_i, 0.75 \cdot S_i, 1 \cdot S_i\}$.

Finally, the union of all C_i sets, C_S was determined, yielding the final set of contact samples sizes. For every number of contacts in this set, 50 samples were generated.

Table 5.1 illustrates the instantiation of C_i for every S_i and the union set C_S.

As for the infectious agent data, it was assumed that the effective contact rate was $\beta_a = 0.025$ and the recovery rate $\delta = 0.2$. The recovery rate value is taken from seasonal influenza, as shown by [30]. The ratio between both values is in the same order of magnitude as in [35].

In order to acquire credible statistics, the average for all epidemic threshold approximations is found and placed in a 95% confidence interval, as exposed in 5.1.2.

In the produced charts, the average values are marked in black and the error bars, when present, represent the calculated confidence interval.

Each chart follows a notation for representing epidemic threshold approximation curves. $S_i - baseIter - per\frac{C_i}{S_i} - [w][s]$, where $Iter$ stands for the number of iterations used for the approximation and $\frac{C_i}{S_i}$ yields a relation of contact coverage for network size. For reduced ambiguity, w and s are optional properties which indicate respectively the application of the weighting and scaling processes.

The following experiments were executed with resort to the generated data and the system's *Learning* layer.

5.2.1.A Epidemic threshold approximation

The purpose of this experiment is the verification of the iterative eigenvalue method used in the *epidemic prediction* module, whose results are used to calculate the epidemic threshold, as shown in 4.2.2.D. This method yields an approximated value and it is necessary to verify that it is both numerically and theoretically correct.

The network data generated for this section, with resort to R, is used. As the aggregation of contact data is independent from these metrics, it is not considered in this experiment.

To have a basis to compare numerical results with, a comparison with R was arranged. The *epidemic prediction* approximation of the epidemic threshold is compared with the threshold results produced by R. These results are computed with the maximum of the absolute value of the eigenvalues produced by R's eigenvalue approximation methods.

Theoretical correction is ensured if the plotted values follow the epidemic threshold trend foreseen by other authors.

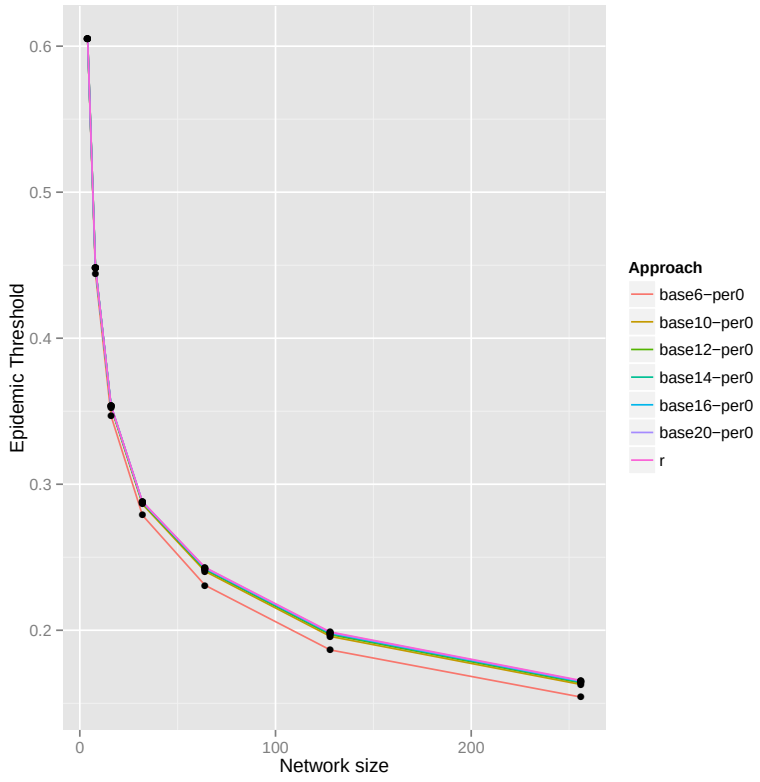

Figure 5.2: Epidemic threshold calculated by the system versus R

The chart, in Figure 5.2, plots the average epidemic threshold for different network sizes and number of iterations. As in [35] and [9], it is verified that as a scale-free network's size increases the epidemic threshold decreases exponentially, eventually leading to its vanishment ($\lambda_c = 0$).

The error between approximations and R should be low enough for the number of iterations

used. It can be minimised by increasing the number of iterations used by the algorithm. There is, however, a compromise between algorithm accuracy and computational complexity.

Only even iterations provide usable values, as on odd iterations, the trace of the matrices produced from the social contact network adjacency matrix becomes zero.

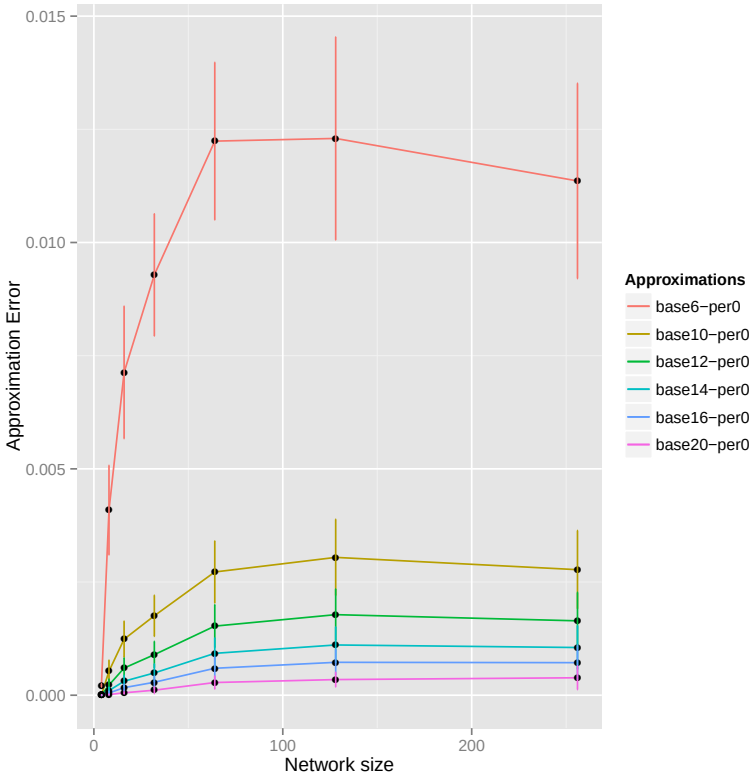

Figure 5.3: Epidemic threshold approximation error versus R

In Figure 5.3, the error between R and the algorithm with different iterations is plotted.

It is shown that the error can be greatly reduced with the number of iterations. As such, the implemented algorithm, when properly parametrised, yields results comparable to R.

This algorithm is adequate for being employed in the system. However, as system performance is a concern, a suitable number of iterations to use in the following tests needs to be chosen. It is visible that from the plotted iteration curves, the one representing 20 iterations provides the best fit. Nonetheless, the one for 16 iterations was chosen as it yields comparable results for a lower cost.

5.2.1.B Epidemic threshold with weighting and scaling

This experiment is aimed at concluding that the epidemic threshold is directly increased by the weighting of a network and that scaling is required, as verified in the work of [35] and [102].

For the same infectious agent parameters, a given weighted network is more resilient than its weightless counterpart. There is a need to lessen weighting impact, so that network and contact

data can be combined without significantly affecting the epidemic threshold. This creates the need for scaling.

To verify this principle, the epidemic threshold for networks of different sizes, produced for this section, was plotted in their un-weighted, weighted and weighted and scaled versions. The results are present in figure 5.4.

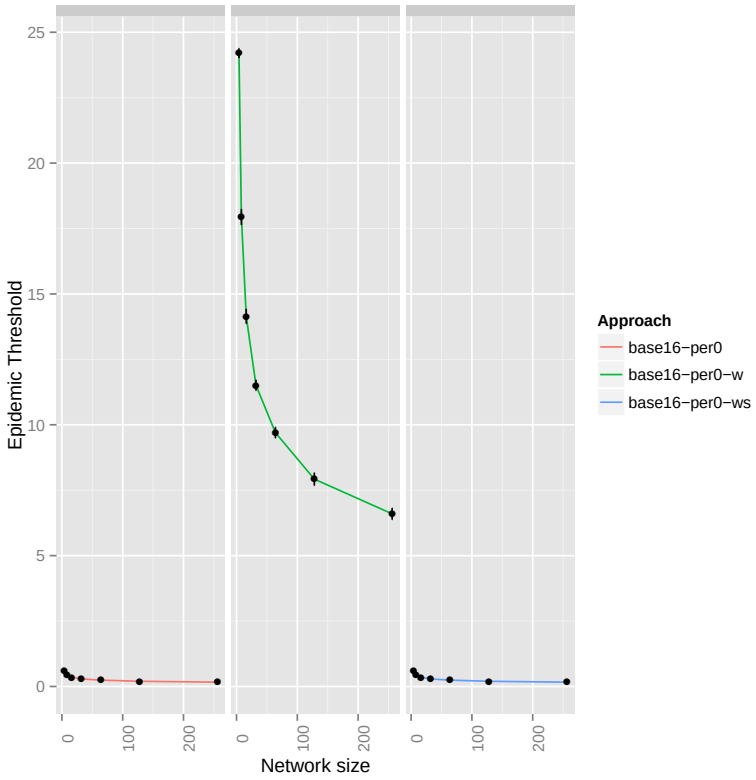

Figure 5.4: Epidemic threshold for weighted and un-weighted networks

As expected, it is visible that the weighting process clearly boosts the epidemic threshold. Scaling compensates this increase, bringing the epidemic threshold down to values on the same order of magnitude as the original network data.

5.2.1.C Epidemic threshold evolution and contact modalities

This experiment is aimed at concluding that the weighting algorithm, that is performed to merge network and contact data, correctly supports the model assumptions. This is done by assessing how different contact modalities, both in the number and distribution of contacts impact the epidemic threshold.

It is considered that more contact diversity in the network should lead to a higher chance of an epidemic.

A contact distribution, in this experiment, can be either egocentric or uniform. In a uniform distribution, contacts are spread evenly between the actors and over the network. Egocentric

distributions assign all contacts to a given actor. Contact distribution contributes to the creation and stress of network bottlenecks.

In a network configuration with bottlenecks, there are relatively few links between highly clustered sub-networks. Some individuals constitute the sole link between their cluster and the rest of network. Their presence constitutes a bridge to epidemic spread.

A small positive epidemic threshold or isoperimetric constant indicates the presence of network bottlenecks. In this way, the creation of network bottlenecks may directly impact the decrease of the isoperimetric constant of a graph.

Furthermore, the assignment of a significant number of contacts to a single node should impact the epidemic threshold decrease more significantly than the uniform distribution of the same contacts over more nodes. The number of significant contacts is a function of network size.

These premises can be analytically verified by a decreasing epidemic threshold. To make this verification possible, for every analysed network size, contact-free control network samples are included for comparison with samples with contacts. If the model implementation is correct, contact-free networks should always yield a higher epidemic threshold than networks with contacts.

Contact data is pre-generated and its attribution within a network is sequential, implying that for different analysis it does not change for a given network. In this data it is not possible for a node to have a contact with itself, meaning that network loops are not part of the data. Also, contacts are established between members of the network. If external actors were included, the epidemic threshold would decrease due to the introduction of new network nodes and consequent increase in network size.

For both Figure 5.5 and 5.6, the independent axis stands for a relative measure of contacts for network size, expressed as $\frac{C_n}{S_n}$. C_n is the amount of contacts spread on the network and S_n the network size. It should be noted that when $\frac{C_n}{S_n} = 0$, there is no contact attribution to the network. The name code for the curves is S_n, standing for their size.

Figure 5.5 depicts the epidemic threshold for networks of different network sizes with an increasing contact percentage. This percentage results from the uniform distribution of contacts.

It is visible that the uniform assignment of contacts affects the epidemic threshold. It can be concluded that, in this model, a relatively small number of contacts per individual has impact in the creation of network bottlenecks and thus direct impact in epidemic persistence.

In a more intuitive explanation, if each individual in a group encounters the same number of individuals, there is an increase in contacts between susceptible individuals in the group that is sufficient for an infectious disease to boost itself and for an epidemic to emerge.

In Figure 5.6, the impact of the variation of network size is compared with an increase in contact percentage. The contact distribution is egocentric.

It is noticeable that concentrated contacts lead to a faster decrease in the epidemic threshold, which results from the larger relative bottleneck presence of the resulting network.

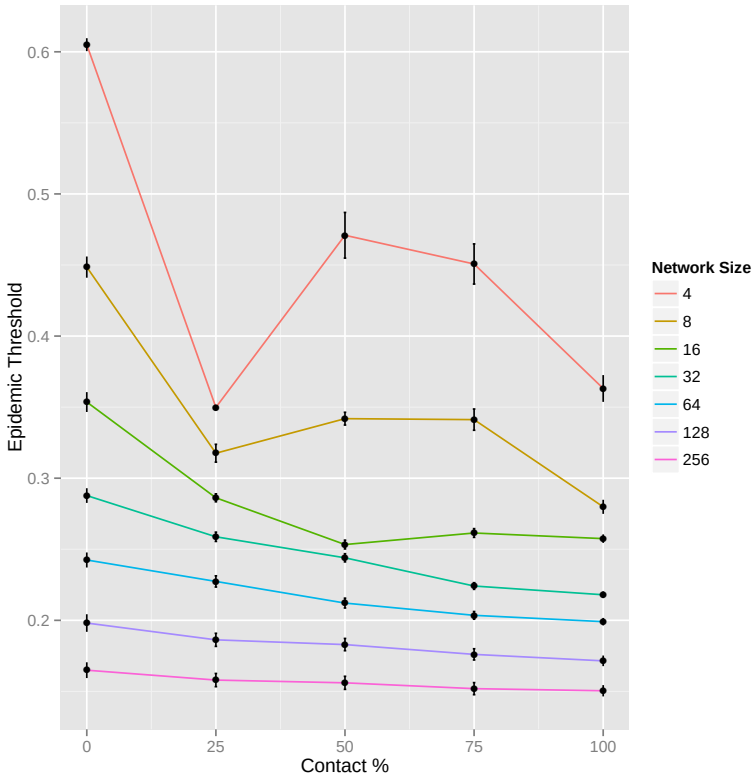

Figure 5.5: Epidemic threshold for uniform contacts as a percentage of network size

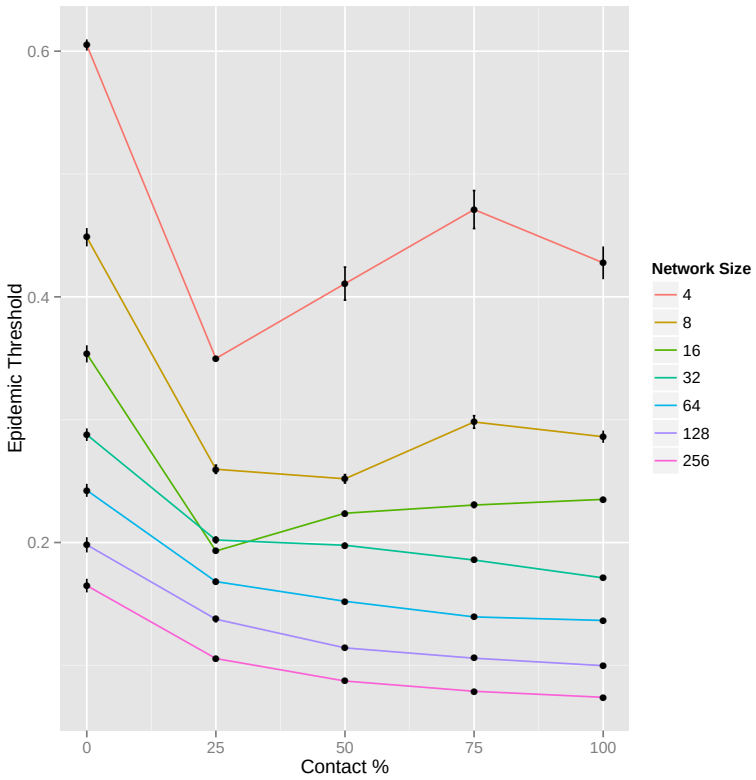

Figure 5.6: Epidemic threshold for egocentric contacts as a percentage of network size

In other words, if individuals in a group contacts a significant amount of people, a would be infectious agent would have more paths to spread and, eventually, breakthrough in the form of an epidemic.

In both scenarios, in networks of relatively smaller size, the contact introduction process greatly impacts the topology [35], effectively reshaping it into providing a threshold that would be expected from larger network sizes. This is associated with their lower size and, thus higher susceptibility to weight clusters.

It is visible that the increase in contacts results in a decrease in epidemic threshold from the initial point of zero contacts in the network. Hence, the impact of contact data in the network is a dominant factor in its susceptibility to an infection [35], in the limit, eclipsing the shape of network data. Thus, with a weighting scheme, it is possible to impact the model without being constrained by the topology of the sampled network.

Finally, it is verified that the introduction of random contacts has an impact over network data, resulting in a lower epidemic threshold [35] that decreases with network size.

By analysing this results and taking into account the conclusions of the previous model experiments, it is concluded that the implemented *Learning* layer is analytically sound.

5.2.2 System-wide Testing

This section assesses the integrated system's performance and scalability. To assess this, calculations have to be performed over different combinations of the number of users, network size and contact coverage.

The data used for this is the one generated for the previous test suite. The infectious agent parameters remain unchanged.

Both the system clients and the data processing backend run on the same machine.

As this is a system-wide testing scenario, network data is sampled for each user. Thus, for a network of a given size, not all nodes are detected by the user base.

The main performance metric to compute is the end-to-end time δ_T of system communication with its users. With resort to it, it is possible to assess scalability.

$$\delta_T = T_f - T_i \tag{5.4}$$

Equation (5.4) describe this metric. T_f is the time at which a client application receives a *results* and re-enters its *associated* state. T_i stands for the time at which it receives a *data request* message. This metric accounts for both processing and communication time.

The system was configured to deployment settings, resulting in no Socket.IO or console logging that would interfere with the measurements.

The network data in this section is sampled through the system clients. As such, each client will contribute to the perception the system has of the overall network topology. A base network

of 256 nodes was chosen for all experiments. Parts of it are sampled by each system user.

In every experiment, at least 50 timing samples were collected for each configuration of user base size and contacts for the network data. Their value was averaged and placed in a 95% CI.

For experiments to have a maximum ceiling for the automatic triggering of the system *data analysis*, an illustrative timeout value of five seconds was defined for the server.

All experiments were triggered with resort to the test platform. Users were configured and launched as local web applications and their perceived round timing values reported to the platform in real-time.

As each system round generates a single δ_T sample for the each user, sampling repetition can be scheduled within a pre-defined periodicity.

These measurements and their subsequent communication influence the timing of the results themselves. As such, the sampling period was set to a value of $512ms$, which was found to be low enough not to jeopardize the results.

5.2.2.A End-to-end time for network and user base size

In this experiment, one aims to verify which factor, either network or user base size, contributes the most to the increase of end-to-end time.

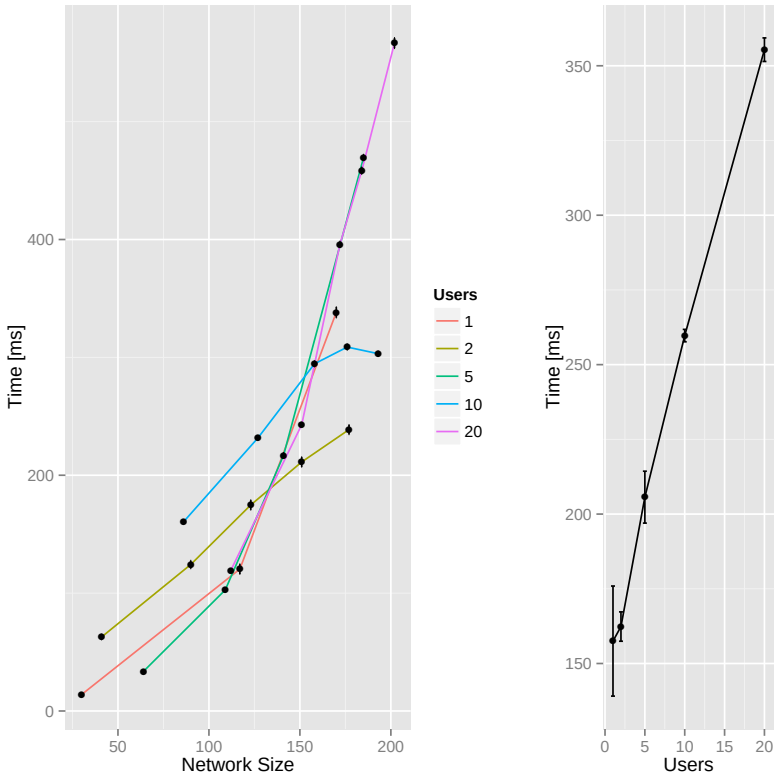

Figure 5.7: End-to-end time for network size and amount of users

As shown in the left facet of Figure 5.7, end-to-end time increases with network size.

In the right side of Figure 5.7, it is demonstrated that the size of the user base contributes to an at least linear increase in the time taken for an end-to-end response.

This is caused by the inherent communication delay when broadcasting both *data request* and *results* messages to more clients.

Also, due to the network sampling-oriented approach of this experiment, more users bring at least a linear increase in the size of the sampled network, in terms of actors, further contributing to the computational delay.

By comparing both plot facets, it is possible to verify that network size has a more relevant contribution to system response time.

It can be concluded that the *epidemic prediction* algorithm constitutes a bottleneck to system scaling, as it evolves polynomially to both network and user base size increases. Due to its polynomial nature, the algorithm is nonetheless scalable.

Improvements can be obtained from the use of more efficient numerical solutions for the determination of the dominant eigenvalue of an adjacency matrix. One example is the use of more complex divide-and-conquer eigenvalue approximation algorithms. These can be exploited, as the spectral value of a matrix is the spectral value of the smaller square matrices that compose it [108].

5.2.2.B End-to-end time and contacts

The reason for this experiments is the assessment of the timing impact of adding contacts to the network in relation to user base size. In this way, the time taken for end-to-end communication was measured for both contact-free networks and networks with an egocentric contact distribution.

In this experiment, a user implies at least the presence of an actor in the social contact networks that are used. However, this value is typically higher due to the scale-free degree distribution of the network data that was generated in 5.2.1, which impacts sampling.

The left side of Figure 5.8 represents the impact of contacts on system end-to-end time, while considering different numbers of users.

The independent axis represents the number of contacts a a fraction of the perceived network size. This ratio may become superior to one as a network may have experienced more contacts than its number of nodes.

So that a relative comparison is possible, data points where $\frac{C_n}{S_c} = 0$ represent zero contacts for that network size.

It is observed that an increase in the number of contact data relations, yields a significant impact on the increase of system response time, when compared to network sizes with no contacts. As the network is sampled by the users, the size of the network perceived by the system does not reflect its absolute size. More contacts conduce to more nodes getting detected and in an increase in size, which further conditions the overall delay.

Taking Figure 5.7 into account, it can be concluded that an increasing user base is correlated

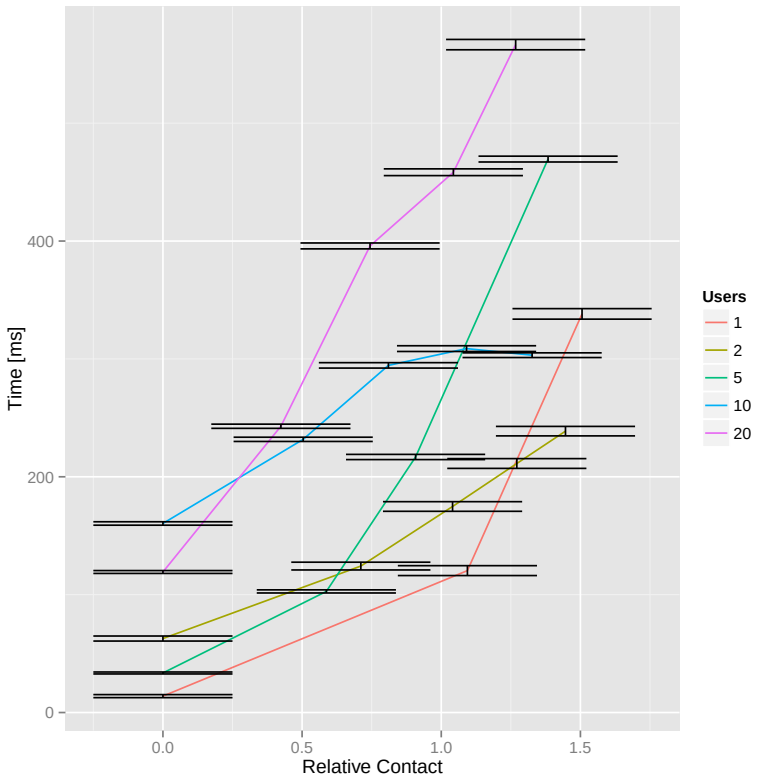

Figure 5.8: End-to-end time for contacts as a percentage of network size

with higher response times. Finally, the impact of contacts is found not to affect the overall growth trend associated with an increasing network size.

5.2.3 Deployment Testing

This section aims to validate the system in a realistic environment.

Testing has to occur in a more empirical context, so that theoretical predictions have a chance to be carried out in reality and that model assumptions can be refined [8].

For these reasons, these tests employ the use of real mobile sensing devices. For a comparative analysis, the data is identical to the one used in the previous section.

Under the same network, two Android smartphones were associated to the system. The deployment of this number of devices remains feasible to accomplish, while still being relevant to the analysis. Their models are labelled in the charts, where appropriate.

The communication to the data processing backend goes through an access point.

A minimum of 50 timing samples were collected per combination of number of devices and network and contact data merges. The sample values were averaged and placed in a 95% CI.

For an improved comparison of results, the base system configuration employed in the previous section was reused. Due to increased communication delay in this context, the timeout triggering the system *data analysis* was increased from five to ten seconds. As a result of the lower relative

computing power of the mobile devices used when compared to the conditions in the previous section, the sampling period had to be increased to $1500ms$.

It was found that the differences in device capabilities affects the timing results. Moreover, device responsiveness was affected by the periodic activity of other running background applications.

The operating system version of these devices also had impact on communications, as Android browsers for versions under 4 do not support Websockets [109]. This leaves Apache Cordova unable to communicate through its wrapped web application. To overcome this, Socket.IO's communication fallback mechanisms were set up.

The limitation on the number of users and size of data that it is feasible to source, provides a more restricted view into system performance.

Finally, in an ideal deployment environment, system communication would be supported in cellular technologies. However, this form of testing is not possible due to its associated costs.

This testing process is a validation of the integrated system addressing this thesis challenge. Nonetheless, its scope should be widened. Further testing at a larger scale should be performed in future work.

5.2.3.A Real end-to-end time for network and user base size

Taking into account the assessment materialised in system-wide testing, the purpose of this experiment is to compare this results with a real deployment environment.

As shown in Figure 5.9, with the introduction of real delays, the major source of delay remains the network size of the perceived network.

As expected network delays are superior in this assessment. They also present a much higher variability. These results are according to what would be expected from a real communication environment, where packet loss and lack of connectivity are likely to occur. Also, differences in device performance affect the precision of the results.

5.2.3.B Real end-to-end time and contacts

The purpose of this experiment is the comparison of the real integrated system results with the ones of the equivalent experiment in 5.2.2.

In Figure 5.10, it is visible that, as in system-wide testing, variable contact data impacts system response time. However, an increase in contacts is ultimately associated with an increase in network size.

The scalability of the system is constrained by the performance of its analysis algorithm. This algorithm is mostly constrained by network size, which is the factor that effectively influences end-to-end time.

The number of users also influences the delay in communications and it brings an additional cost in nodes sampled per user.

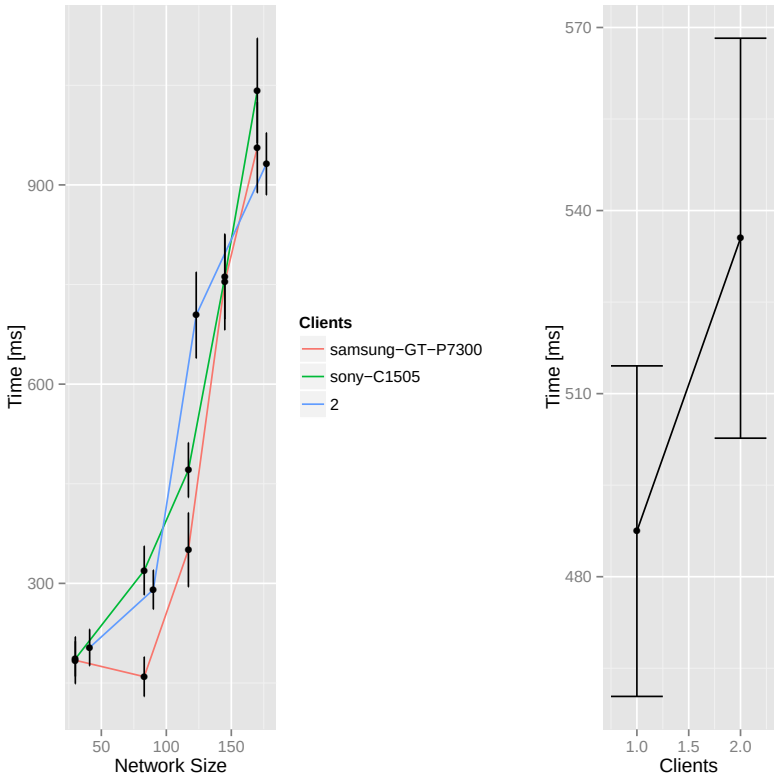

Figure 5.9: Real end-to-end time for network size and amount of users

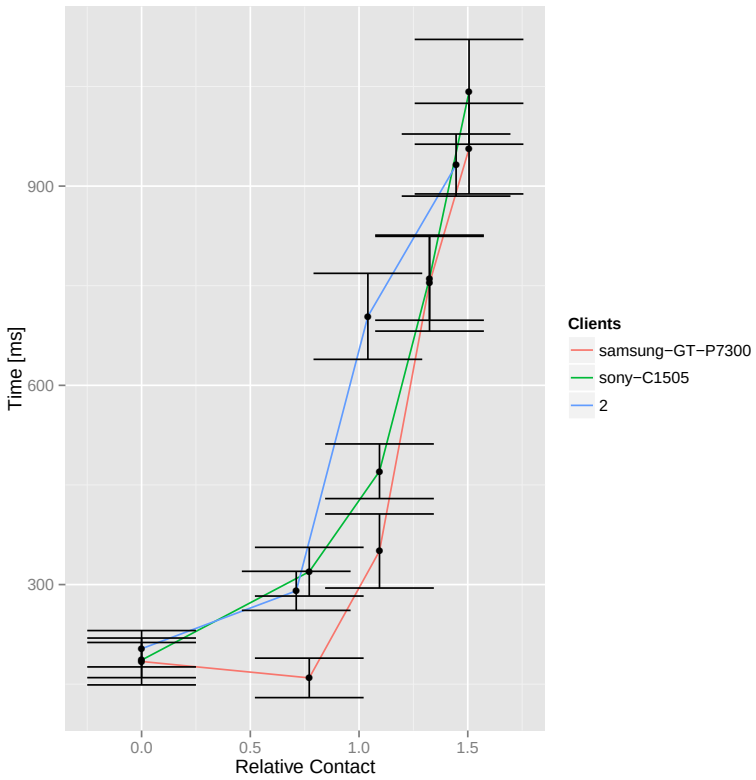

Figure 5.10: Real end-to-end time for contacts as a percentage of network size

5.2.4 Data Source Testing

The aim of this section is to test the system with real data. To acquire data from a realistic data source, network and contact data were obtained from Facebook. This is done with resort to the Facebook Test User API [59].

5.2.4.A End-to-end time with a real data source

The purpose of this experiment is to assess the impact of adding a social network platform, as the system's data source, to the end-to-end time defined in 5.2.2.

There are two scenarios to cover: local and mobile. For the local case, Facebook data accesses are achieved through a single machine in a configuration alike the one in 5.2.2. On the mobile scenario, they are done with resort to sensing devices as in 5.2.3.

A base network of 15 nodes was chosen for the experiment. 50 time samples were collected per each of the scenarios. These sample values were averaged and placed in a 95% CI. A single mobile device was used.

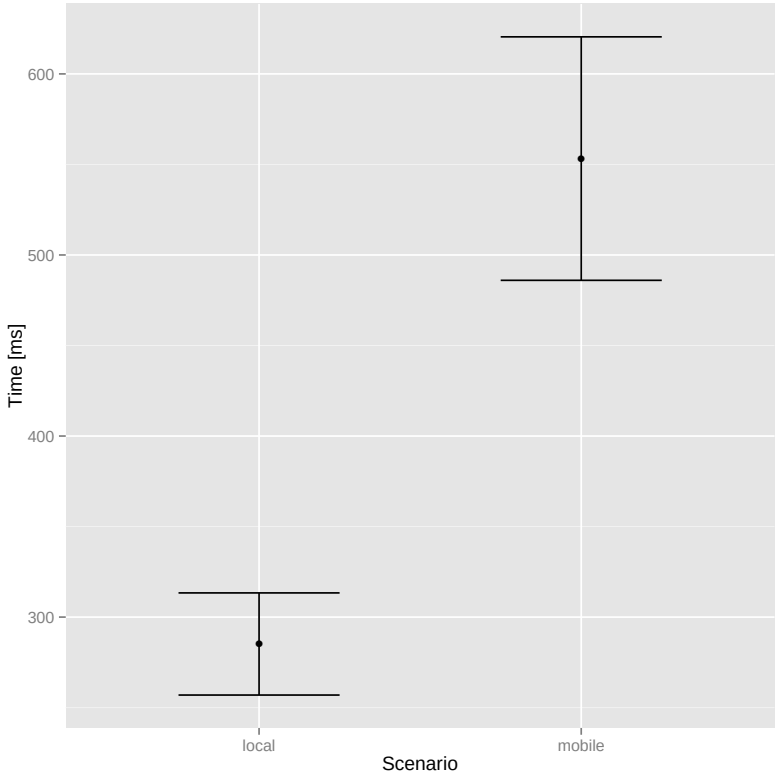

Figure 5.11: End-to-end time with a real data source

Figure 5.11 plots the end-to-end time for the two situations. It is noticeable that the overall end-to-end time is both higher and more prone to variation on mobile deployment. This is inherently related to the lower computational power of the mobile devices. Also the distributing of system clients over external devices has an additional level of communication and an associated delay.

Both situations can be qualitatively compared to the ones in 5.2.2 and 5.2.3, where end-to-end time is measured against the number of users in the system. It is visible that for both cases, there is a considerable delay contribution. The factor for this is the communication with the data source, which has its own associated cost in time.

Either caching the data requested from it or receiving data updates in the form of events could contribute to the minimization of this factor's impact.

6

Discussion and Conclusions

6.1 Conclusions

In the future, mobile sensing systems will provide scaled views of communities and individuals, contributing to solve problems affecting society [1].

Important security developments arise from these systems, where a set of sensing devices has to communicate according to security criteria in dynamic large-scale environments.

When mobility issues affecting the resolution and accuracy of these systems are finally circumvented, a new area of information extraction will bloom and its numerous applications will bring benefits to the various domains of science.

Furthermore, once technical issues are solved, this field will constitute an innovative groundbreaking technology applicable to various domains in society, namely social networking, transports, health and energy [1].

This thesis presents an innovative system aimed at addressing these technical challenges.

The proposed solution deals with privacy concerns within some limitations. Nonetheless, complete data anonymization and analysis are two areas which pose conflicting approaches. As such, the techniques employed only go as far as feasible and should be regarded as a proof of concept in the system architecture.

The resulting system solves the issue of distributed resource management distribution by placing most of the concerns on a centralized entity, liberating the mobile sensing devices of unnecessary computation. Other approaches that compromise more may be superior, but their impact has to be properly assessed and studied.

The bottleneck to system inference is in the size of the sampled network. Through the path laid

by this work, a more systematic analysis in the field of network data sampling should be raised, potentiating the impact of the currently achieved results.

Furthermore, significant performance improvements and system scalability can be unleashed by taping the potential of divide-and-conquer matrix analysis techniques and by extrapolating the validated analysis to the cloud.

The principle underlying the executed systems approach is that the whole may be different from the sum of its parts. Consequently, it is legitimate that systems models outcomes may contradict the results that are originated from other reductionist approaches [25].

Predictive system results provides one level of inference, which does not conflict with other views provided by looking at the same data with different abstractions. Quoting Frederich von Hayek, the father of complexity theory, in his Nobel acceptance speech [8]:

"...as we penetrate from the realm in which relatively simple laws prevail [the physical sciences] into the range of phenomena where organized complexity rules...often all that we shall be able to predict will be some abstract characteristic of the pattern that will appear...yet...we will still achieve predictions which can be falsified and which therefore are of empirical significance".

6.2 Contribution Summary

A new system architecture to concretise opportunistic mobile sensing systems with the applicational context of epidemic prediction was proposed, implemented and validated.

Problems related to the integration of all the requirements imposed on the solution were addressed and possible solutions to such problems proposed by previous scientific work were discussed.

The system considers the real-time processing of multi-modal data, such as social network and location data from mobile devices. User data is anonymized and it is only shared after an initial consent, while subsequent sampling occurs opportunistically. Such data is integrated with user social network data (originated from Facebook) and fed into a data merge and analysis algorithm for epidemic prediction.

The system is hence fully automated on an end-to-end perspective. Epidemic prediction is based on the analysis of the epidemic threshold of the sensed network and the infectious agent parameters, resulting in an aggregate metric for epidemic prediction. This analysis is centred on the sampled data and, as such, it is not generalisable to the whole population. Nonetheless, it sets the ground for algorithmic extensions for large-scale data merge and processing.

By delivering to users information, concerning outbreaks that might conduce to an epidemic, this system can contribute to the detection of communities that are vulnerable to a given infectious disease.

Finally, by weighting the concerns pertaining to the wide problematic spectrum of which are part the multidisciplinary area covered by this work, it was possible to arrive at a proof of concept solution that effectively implements these systems as a whole, targeting its problem domain with

accuracy.

The loss of generalisability is surmounted by the the ground set for algorithmic extensions positioned at large-scale data analysis that may target a wider problematic scope.

Ultimately, the achieved architectural execution constitutes a stepping stone for other solutions, which may be aimed at a similar community problematic, but oriented towards distinct application areas. One example of such an area is the spread of computer viruses.

6.3 Future Work

On the pursuit of improving the outcomes of this work, there are some focus area that should be mentioned.

The analysis module imbued in this system is currently only parametrisable by researchers possessing programming knowledge. As this is a tool that stems from a multidisciplinary area, it would be appropriate to offer system access and parametrisation to non-programmers. To achieve this, the creation of an user interface exposing the analytical functionality behind this system and its integration of configuration files would be adequate. This task was partially tackled in the development of the testing platform used for system validation, but as its target was not ideologically in line with this objective, it does not constitute the seamless experience it could otherwise provide.

The system explores the assumption that social networks can be modelled as scale-free networks, due to the existing lack of data. Although the model has been assessed to be fit to this assumption, more detailed and realistic social network data on the natural evolution of infectious diseases and their transmission is needed. If access to such data remains unfeasible, more forms of validation including more network topologies and other suitable network-oriented models should be attempted.

Contact data acquisition is focused on social network platforms. In line with the ground set for pervasive sensing, the perception of contacts through other means of lower location granularity and with a real-time data acquisition should be exploited. One example, would be the use of Apache Cordova phone sensor access to sample GPS data. In the current architecture, such an endeavour would require extensions to the data filtering and analysis modules, as it would have to cope with sampling errors, stale locations and the matching of global positioning coordinates to effective contacts or user meetings. Also, there is the matter of contact timing and effectively ensuring it lasts for an amount of time that is deemed significant.

Better sampling of relational data can help uncover more interaction patterns. Different approaches to this problematic should be tried, while taking into account heuristic metrics for appropriate sample size and the shortest path distance between actors. If Facebook is used as a data source, studies presented in [60] and [61] can serve as a suitable parameter source.

The sampling process offered by the system can be improved in terms of the efficiency of

communications. Client data can be sensed in real-time and cached until an analysis round is started. Presently, some social network platforms already provide support for event-based data acquisition in a publish-subscribe communication approach. If such a methodology would be employed, the redundancy of sampled data would be minimised.

Ultimately, more sensing modalities could be integrated into the sensing layer, providing a richer base of data to experiment with. For instance, the biosignal acquisition systems engineered during the Harvard-Portugal research could provide an extra dimension of information, possibly enabling epidemic models covering vertical disease transmission.

The case of disconnected users is not covered in the present work, i.e. if two system users do not possess any social contact network neighbour of any degree between them. A more distributed analysis component could help tackle the separate outputs that are required to face this issue.

System participation is opportunistic. However, so that it can occur the respect for privacy requirements is paramount. The data anonymization techniques employed are in the limit susceptible to malicious attacks. Having this in mind, a throughout system penetration testing study should be conducted, aiming to identify and subsequently mitigate system vulnerabilities.

The algorithms responsible for the calculation of system metrics can benefit from parallelisation. Divide-and-conquer matrix eigenvalue calculation techniques can be used in synergy with this measure, enabling better scaling and performance. Data merging system components with added data synchronisation and consistency complexity would emerge from this change.

Overall resource management on the system calls for a more comprehensive study on system architectural module disposition and placement. On the specific case of mobile sensing devices, more in locus resource consumption analysis should be conducted, targeting an improved and fully validated physical architecture.

The usefulness of system results can be improved and more complex metrics can be produced. When attempting more forms of analysis, often result generalisation will become a barrier. In computational epidemiology, future research should aim for a clearer synergy between theory and empiricism, providing the field with a more systemic approach to experimentation and inference [8].

Finally, cloud technologies constitute a major step in enabling large-scale data merging, analysis and result dissemination. They are the most immediate sound basis for permitting scaling to a population-level analysis and their potential should be harnessed in future work.

Bibliography

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A. T. Campbell, and D. College, "A Survey of Mobile Phone Sensing," IEEE Communications Magazine, vol. 48, no. 9, pp. 140–150, 2010.

[2] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn, "The rise of people-centric sensing," Internet Computing, IEEE, vol. 12, no. 4, pp. 12–21, 2008. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4557974

[3] D. Zhang, B. Guo, B. Li, and Z. Yu, "Extracting social and community intelligence from digital footprints: an emerging research area," in Ubiquitous Intelligence and Computing. Springer, 2010, pp. 4–18. [Online]. Available: http://www.springerlink.com/index/G85551H8L631837L.pdf

[4] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in Proceedings of the 6th ACM conference on Embedded network sensor systems. ACM, 2008, pp. 337–350. [Online]. Available: http://dl.acm.org/citation.cfm?id=1460445

[5] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich, "Mobiscopes for human spaces," Pervasive Computing, IEEE, vol. 6, no. 2, pp. 20–29, 2007. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4160602

[6] A. Kansal, M. Goraczko, and F. Zhao, "Building a sensor network of mobile phones," in Information Processing in sensor Networks, 2007, pp. 547–548. [Online]. Available: http://dl.acm.org/citation.cfm?id=1236433

[7] D. Peebles, H. Lu, N. Lane, T. Choudhury, and A. Campbell, "Community-guided learning: Exploiting mobile sensor users to model human behavior," in Proc. of 24th AAAI Conference on Artificial Intelligence, 2010. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/download/1933/2263

[8] R. Rothenberg and E. Costenbader, "Empiricism and theorizing in epidemiology and social network analysis," Interdisciplinary perspectives on infectious diseases, vol. 2011, p. DOI:10.1155/2011/157194, Jan. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2992814&tool=pmcentrez&rendertype=abstract

[9] R. Pastor-Satorras and A. Vespignani, "Epidemic Spreading in Scale-Free Networks," Physical Review Letters, vol. 86, no. 14, pp. 3200–3203, Apr. 2001. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.86.3200

[10] J. a. Andrade, A. Duarte, and A. Arsénio, "Social Web for Large-Scale Biosensors," International Journal of Web Portals, vol. 4, no. 3, pp. 1–19, 2012. [Online]. Available: http://www.igi-global.com/article/social-web-large-scale-biosensors/75199

[11] A. Arsénio, H. Serra, R. Francisco, F. Nabais, J. a. Andrade, and E. Serrano, "Internet of Intelligent Things: Bringing Artificial Intelligence into Things and Communication Networks," in Inter-cooperative Collective Intelligence: Techniques and Applications. Berlin: Springer, 2014, pp. 1–37.

[12] J. a. Andrade and A. Arsénio, "Epidemic Spreading Over Social Networks Using Large-scale Biosensors: A Survey," in Procedia Technology Volume 5, 2012, pp. 922–931.

[13] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in Proceedings of the 7th international conference on Mobile systems, applications, and services. New York, New York, USA: ACM, 2009, pp. 55–68. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1555816.1555823http://dl.acm.org/citation.cfm?id=1555816.1555823

[14] S. Consolvo, D. McDonald, T. Toscos, M. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, and Others, "Activity sensing in the wild: a field trial of ubifit garden," in Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. ACM, 2008, pp. 1797–1806. [Online]. Available: http://dl.acm.org/citation.cfm?id=1357335

[15] A. Kapadia, N. Triandopoulos, and C. Cornelius, "AnonySense: Opportunistic and privacy-preserving context collection," in Pervasive Computing, 2008, pp. 280–297. [Online]. Available: http://www.springerlink.com/index/967211Q23663002L.pdf

[16] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," in Proceedings of the 7th international conference on Mobile systems, applications, and services. ACM, 2009, pp. 165–178. [Online]. Available: http://dl.acm.org/citation.cfm?id=1555834

[17] C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, and M. V. Marathe, "EpiSimdemics: An efficient algorithm for simulating the spread of infectious disease over large realistic social networks," in International Conference for High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008., 2008, pp. 1–12.

[18] Y. Chen, C. Tseng, and C. King, "Incorporating geographical contacts into social network analysis for contact tracing in epidemiology: A study on Taiwan SARS data," in Advances in Disease Surveillance: Abstracts from the 2007 Conference of the International Society for Disease Surveillance, 2007, p. 4. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-72608-1_3

[19] L. Lopes, J. Zamite, B. Tavares, F. Couto, F. Silva, and M. Silva, "Automated social network epidemic data collector," in INForum informatics symposium, Lisboa, 2009, pp. 1–10. [Online]. Available: http://homepages.di.fc.ul.pt/~fjmc/files/workshoplopes-inforum2009.pdf

[20] P. F. Gorder, "Computational Epidemiology," Computing in Science & Engineering, vol. 12, no. 1, pp. 4–6, Jan. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5372174http://darwin.bio.uci.edu/~kabbas/research/papers/bayes1h.pdf

[21] S. V. Noort, M. Muehlen, and A. Rebelo, "Gripenet: an internet-based system to monitor influenza-like illness uniformly across Europe." vol. 12, no. 7, pp. 1–14, 2007. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17991409

[22] J. Brownstein, C. Freifeld, and Others, "HealthMap: the development of automated real-time internet surveillance for epidemic intelligence," Euro Surveill, vol. 12, no. 11, p. E071129, 2007. [Online]. Available: http://www.eurosurveillance.org/ViewArticle.aspx?ArticleId=3322

[23] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data." Nature, vol. 457, no. 7232, pp. 1012–4, Feb. 2009. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/19020500

[24] M. Kretzschmar, M. G. M. Gomes, R. a. Coutinho, and J. S. Koopman, "Unlocking pathogen genotyping information for public health by mathematical modeling." Trends in microbiology, vol. 18, no. 9, pp. 406–12, Sep. 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20638846

[25] A. M. El-Sayed, P. Scarborough, L. Seemann, and S. Galea, "Social network analysis and agent-based modeling in social epidemiology." Epidemiologic perspectives & innovations :

EP+I, vol. 9, no. 1, p. 1, Jan. 2012. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3395878&tool=pmcentrez&rendertype=abstract

[26] R. M. Christley, G. L. Pinchbeck, R. G. Bowers, D. Clancy, N. P. French, R. Bennett, and J. Turner, "Infection in social networks: using network analysis to identify high-risk individuals." American journal of epidemiology, vol. 162, no. 10, pp. 1024–31, Nov. 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16177140

[27] N. a. Christakis and J. H. Fowler, "Social Network Visualization in Epidemiology." Norsk epidemiologi = Norwegian journal of epidemiology, vol. 19, no. 1, pp. 5–16, Jan. 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3337680&tool=pmcentrez&rendertype=abstract

[28] Z.-p. Li and G. Shao, "Halting Infectious Disease Spread in Social Network," in IWCFTA'09 International Workshop on Chaos-Fractals Theories and Applications, 2009, no. November 2002. IEEE, 2009, pp. 305–308. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5362017

[29] I. M. Longini, A. Nizam, S. Xu, K. Ungchusak, W. Hanshaoworakul, D. a. T. Cummings, and M. E. Halloran, "Containing pandemic influenza at the source." Science (New York, N.Y.), vol. 309, no. 5737, pp. 1083–7, Aug. 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16079251

[30] M. Martcheva, "Introduction to Mathematical Epidemiology," 2001. [Online]. Available: http://www.math.ufl.edu/~maia/BIOMATHSEM/Lecture1.pdf

[31] L. Meyers, B. Pourbohloul, M. E. J. Newman, D. M. Skowronski, and R. C. Brunham, "Network theory and SARS: predicting outbreak diversity," Journal of theoretical biology, vol. 232, no. 1, p. 71.81, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022519304003510

[32] D. L. Chao, M. E. Halloran, V. J. Obenchain, and I. M. Longini Jr, "FluTE, a Publicly Available Stochastic Influenza Epidemic Simulation Model," PLoS Comput Biol, vol. 6, no. 1, p. e1000656, 2010. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1000656

[33] J. O. Kephart, "How topology affects population dynamics," in C Langton ed Artificial Life III Studies in the Sciences of Complexity. ADDISON-WESLEY PUBLISHING CO, 1994, pp. 447–463. [Online]. Available: http://www.mpi-sb.mpg.de/services/library/proceedings/contents/alife92.html

[34] J. H. Jones, "Notes On R0," 2007. [Online]. Available: http://www.stanford.edu/~jhj1/teachingdocs/Jones-on-R0.pdf

[35] P. Schumm, C. Scoglio, D. Gruenbacher, and T. Easton, "Epidemic spreading on weighted contact networks," in Bionetics 2007 2nd. Bio-Inspired Models of Network, Information and Computing Systems, no. 1. IEEE, 2007, pp. 201–208. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4610111

[36] L. Danon, A. P. Ford, T. House, C. P. Jewell, M. J. Keeling, G. O. Roberts, J. V. Ross, and M. C. Vernon, "Networks and the epidemiology of infectious disease." Interdisciplinary perspectives on infectious diseases, vol. 2011, p. DOI:10.1155/2011/284909, Jan. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3062985&tool=pmcentrez&rendertype=abstract

[37] G. M. Ames, D. B. George, C. P. Hampson, A. R. Kanarek, C. D. McBee, D. R. Lockwood, J. D. Achter, and C. T. Webb, "Using network properties to predict disease dynamics on human contact networks." Proceedings. Biological sciences / The Royal Society, vol. 278, no. 1724, pp. 3544–50, Dec. 2011. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3189367&tool=pmcentrez&rendertype=abstract

[38] K. Abbas, A. Mikler, A. Ramezani, and S. Menezes, "Computational epidemiology: Bayesian disease surveillance," in Proc. of the International Conference on Bioinformatics and its Applications, FL, USA, 2004, pp. 1–12. [Online]. Available: http://darwin.bio.uci.edu/~kabbas/research/papers/bayes1h.pdf

[39] M. S. Handcock and K. J. Gile, "Modeling social networks from sampled data," The Annals of Applied Statistics, vol. 4, no. 1, pp. 5–25, Mar. 2010. [Online]. Available: http://projecteuclid.org/euclid.aoas/1273584445

[40] L. Barros, R. Bassanezi, R. Oliveira, and M. Leite, "A disease evolution model with uncertain parameters," in Joint 9th IFSA World Congress and 20th NAFIPS International Conference, vol. 3, no. C. IEEE, 2001, pp. 1626–1630. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=943794

[41] K. Bisset, J. Chen, X. Feng, V. Kumar, and M. Marathe, "EpiFast: a fast algorithm for large scale realistic epidemic simulations on distributed memory systems," in Proceedings of the 23rd international conference on Supercomputing. ACM, 2009, pp. 430–439. [Online]. Available: http://dl.acm.org/citation.cfm?id=1542336

[42] S. Eubank, "Scalable, efficient epidemiological simulation," in Proceedings of the 2002 ACM symposium on Applied computing. ACM, 2002, pp. 139–145. [Online]. Available: http://dl.acm.org/citation.cfm?id=508819

[43] A. Jamakovic, R. Kooij, P. Van Mieghem, and E. van Dam, "Robustness of networks against viruses: the role of the spectral radius," 2006 Symposium on

Communications and Vehicular Technology, no. 3, pp. 35–38, Nov. 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4133799

[44] Facebook, "Graph API - Facebook Developers," 2012. [Online]. Available: https://developers.facebook.com/docs/reference/api/

[45] A. S. Foundation, "Apache Cordova," 2013. [Online]. Available: http://cordova.apache.org/

[46] T. Fuchs, "Zepto.js: the aerogel-weight jQuery-compatible JavaScript library," 2013. [Online]. Available: http://zeptojs.com/

[47] Knockoutjs.com, "Knockout : Home," 2013. [Online]. Available: http://knockoutjs.com/

[48] I. Joyent, "Node.js," 2013. [Online]. Available: http://nodejs.org/

[49] L. Labs, "Socket.IO: the cross-browser WebSocket for realtime apps," 2013. [Online]. Available: http://socket.io/

[50] K. M. Kowal, "Kriskowal/q," 2013. [Online]. Available: https://github.com/kriskowal/q

[51] S. Loisel, "Numeric Javascript," 2013. [Online]. Available: http://www.numericjs.com/

[52] T. O. Project, "OpenSSL: The Open Source toolkit for SSL/TLS," 2013. [Online]. Available: http://www.openssl.org/

[53] Stanford University, "Stanford Javascript Crypto Library," 2013. [Online]. Available: http://crypto.stanford.edu/sjcl/

[54] O. S. Initiative, "The Open Source Initiative," 2013. [Online]. Available: http://opensource.org/

[55] I. Google, "Google+ API," 2012. [Online]. Available: https://developers.google.com/+/api/

[56] I. Foursquare, "Fourquare for Developers," 2013. [Online]. Available: https://developer.foursquare.com/

[57] I. Twitter, "Twitter Developers," 2013. [Online]. Available: https://dev.twitter.com/

[58] K. Hampton and L. Goulet, Social networking sites and our lives. Washington, D.C.: Pew Internet, 2011. [Online]. Available: http://www.namingandtreating.com/wp-content/uploads/2011/07/PIP-Social-networking-sites-and-our-lives.pdf

[59] Facebook, "Test User - Facebook Developers," 2013. [Online]. Available: https://developers.facebook.com/docs/test$\delimiter"026E30F$_users/

[60] L. Backstrom, P. Boldi, and M. Rosa, "Four degrees of separation," arXiv e-prints, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2380723

[61] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the facebook social graph," arXiv e-print, pp. 1–17, 2011. [Online]. Available: http://arxiv.org/abs/1111.4503

[62] I. Google, "Android Developers," 2013. [Online]. Available: http://developer.android.com/index.html

[63] I. Apple, "IOS Developer Program - Apple Developer," 2013. [Online]. Available: https://developer.apple.com/programs/ios/

[64] B. Limited, "Blackberry Developer," 2013. [Online]. Available: http://developer.blackberry.com/

[65] M. Corporation, "Windows Phone Dev Center," 2013. [Online]. Available: http://developer.windowsphone.com/en-us

[66] A. Inc., "Titanium Mobile Application Development," 2013. [Online]. Available: http://www.appcelerator.com/platform/titanium-platform/

[67] I. Dalmasso and S. Datta, "Survey, comparison and evaluation of cross platform mobile application development tools," in 9th International Wireless Communications and Mobile Computing Conference (IWCMC), 2013, pp. 323–328. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6583580

[68] H. Heitkötter, S. Hanschke, and T. Majchrzak, "Evaluating cross-platform development approaches for mobile applications," in Web Information Systems and Technologies, 2013, pp. 299–311. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-36608-6_8

[69] T. jQuery Foundation, "JQuery," 2013. [Online]. Available: http://jquery.com/

[70] A. Wright and M. Felleisen, "A syntactic approach to type soundness," Information and computation, vol. 115, no. 1, pp. 38–94, 1994. [Online]. Available: http://port70.net/~nsz/articles/classic/wright_felleisen_type_soundness_1992.pdf

[71] EPFL, "Introducing Scala," 2012. [Online]. Available: http://www.scala-lang.org/

[72] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)." Fremont, California, U.S.: IETF, 2006, pp. 1–10. [Online]. Available: http://tools.ietf.org/html/rfc4627.txt

[73] J. Halliday, "Substack/node-browserify," 2013. [Online]. Available: https://github.com/substack/node-browserify

[74] A. Edelman, "The Julia Language," 2013. [Online]. Available: http://julialang.org/

[75] Facebook, "Login Dialog - Facebook Developers," 2013. [Online]. Available: https://developers.facebook.com/docs/reference/dialogs/oauth/

[76] I. Fette and A. Melnikov, "The WebSocket Protocol." Fremont, California, U.S.: IETF, 2011, pp. 1–71. [Online]. Available: http://tools.ietf.org/html/rfc6455):

[77] G. Rauch, "LearnBoost/socket.io-spec," 2013. [Online]. Available: https://github.com/LearnBoost/socket.io-spec

[78] L. Labs, "Socket.IO: the cross-browser WebSocket for realtime apps," 2013. [Online]. Available: http://socket.io/#browser-support

[79] C. Allen and T. Dierks, "The TLS Protocol Version 1.0." Fremont, California, U.S.: IETF, 1999, pp. 1–80. [Online]. Available: http://xml2rfc.tools.ietf.org/html/rfc2246

[80] K. M. Kowal, "Q," 2013. [Online]. Available: http://documentup.com/kriskowal/q/

[81] A. S. Foundation, "Apache Cordova API Documentation," 2013. [Online]. Available: http://cordova.apache.org/docs/en/2.9.0/guide$\delimiter"026E30F$_whitelist$\delimiter"026E30F$_index.md.html

[82] D. Hardt, "The OAuth 2.0 Authorization Framework." Fremont, California, U.S.: IETF, 2012, pp. 1–77. [Online]. Available: http://tools.ietf.org/html/rfc6749.html

[83] N. Provos and D. Mazieres, "A Future-Adaptable Password Scheme," in USENIX Annual Technical Conference, ..., 1999, pp. 1–12. [Online]. Available: http://static.usenix.org/event/usenix99/provos/provos.pdf

[84] C. Percival, "Stronger key derivation via sequential memory-hard functions," Self-published, pp. 1–16, 2009. [Online]. Available: http://www.unixhowto.de/docs/87_scrypt.pdf

[85] B. Kaliski, "PKCS #5: Password-Based Cryptography Specification Version 2.0." Fremont, California, U.S.: IETF, 2000. [Online]. Available: http://tools.ietf.org/html/rfc2898%23section-5.2

[86] Facebook, "64 Bit.- Facebook Developers," 2013. [Online]. Available: https://developers.facebook.com/blog/post/45/

[87] E. Stark, M. Hamburg, and D. Boneh, "Symmetric Cryptography in Javascript," in 2009 Annual Computer Security Applications Conference. Ieee, Dec. 2009, pp. 373–381. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5380691

[88] B. Alman, "Grunt: The JavaScript Task Runner," 2013. [Online]. Available: http://gruntjs.com/

[89] Netlib, "LAPACK - Linear Algebra PACKage," 2013. [Online]. Available: http://www.netlib.org/lapack/

[90] C. Umbel, "NaturalNode/node-lapack," 2013. [Online]. Available: https://github.com/NaturalNode/node-lapack

[91] S. Loisel, "Numeric Javascript: Benchmarks," 2013. [Online]. Available: http://www.numericjs.com/benchmark.html

[92] Facebook, "Extended Permissions - Facebook Developers," 2013. [Online]. Available: https://developers.facebook.com/docs/reference/login/extended-permissions/

[93] O. Foundation, "Password Storage Cheat Sheet - OWASP," 2013. [Online]. Available: https://www.owasp.org/index.php/Password$\delimiter"026E30F$_Storage$\delimiter"026E30F$_Cheat$\delimiter"026E30F$_Sheet

[94] G. Chowell, C. E. Ammon, N. W. Hengartner, and J. M. Hyman, "Transmission dynamics of the great influenza pandemic of 1918 in Geneva, Switzerland: Assessing the effects of hypothetical interventions." Journal of theoretical biology, vol. 241, no. 2, pp. 193–204, Jul. 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16387331

[95] S. Loisel, "Numeric/src/numeric.js at master · sloisel/numeric," 2013. [Online]. Available: https://github.com/sloisel/numeric/blob/master/src/numeric.js#L1478

[96] B. Parlett, "The QR algorithm," in Computing in Science & Engineering. College Park, MD: American Institute of Physics. Society, IEEE Computer, 2000, pp. 51–76. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=814656

[97] E. Gonzalez, "Determination of the dominant eigenvalue using the trace method," IEEE Multidisciplinary Engineering Education Magzine, vol. 1, no. 1, pp. 1–2, 2006. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Determination+of+the+Dominant+Eigenvalue+Using+the+Trace+Method#0

[98] Z. Jelonek, "Solving polynomial equations," Mathematica Aeterna, vol. 2, no. 8, pp. 651–667, 2012. [Online]. Available: http://demmath.mini.pw.edu.pl/archive/dm45_4/4.pdf

[99] T. igraph Project, "The igraph library for complex network research," p. 2013. [Online]. Available: http://igraph.sourceforge.net/documentation.html

[100] H. Wickham, "Ggplot2," 2013. [Online]. Available: http://ggplot2.org/

[101] M. Bostock, "D3.js - Data-Driven Documents," p. 2013. [Online]. Available: http://d3js.org/

[102] G. YAN, T. ZHOU, J. WANG, Z.-Q. FU, and B.-H. WANG, "Epidemic spread in weighted scale-free networks," Chinese Physics Letters, vol. 22, no. 2, pp. 510–513, 2005. [Online]. Available: http://iopscience.iop.org/0256-307X/22/2/068

[103] I. f. S. Wien and M. of WU, "The R Project for Statistical Computing," 2013. [Online]. Available: http://www.r-project.org/

[104] Igraph, "R: Generate scale-free graphs according to the Barabasi-Albert model," 2013. [Online]. Available: http://igraph.sourceforge.net/doc/R/ba.game.html

[105] M. Maechler, "R: Random Samples and Permutations," 2013. [Online]. Available: http://stat.ethz.ch/R-manual/R-devel/library/base/html/sample.html

[106] I. Wolfram Research, "Student's t-Distribution - from Wolfram MathWorld," 2013. [Online]. Available: http://mathworld.wolfram.com/Studentst-Distribution.html

[107] M. Maechler, "R: The Student t Distribution," 2013. [Online]. Available: http://stat.ethz.ch/R-manual/R-patched/library/stats/html/TDist.html

[108] Y. Nakatsukasa and N. Higham, "Stable and Efficient Spectral Divide and Conquer Algorithms for the Symmetric Eigenvalue Decomposition and the SVD," SIAM Journal on Scientific Computing, vol. 35, no. 3, pp. A1325–A1349, 2013. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/120876605

[109] Fyrd, "Can I use... Support tables for HTML5, CSS3, etc," 2013. [Online]. Available: http://caniuse.com/#feat=websockets