Alma Mater Studiorum
Università di Bologna

Master's Degree Course in Computer Science

A Framework for Measuring and Improving Social Inclusion with Network Science

Master Thesis

Supervisor
Saverio Giallorenzo

Co-Supervisor
Angelo Trotta

Author
Emanuele Sinagra

Academic Year 2023-2024

Abstract

Promoting social inclusion in childhood is a crucial step in creating a more equitable and inclusive society. Socially included students improve their self-esteem, self-confidence, academic outcomes, mental health, social relationships, openness to diverse cultures, sense of social responsibility and justice.

This study describes our framework for measuring and improving the social inclusion of study participants using network science techniques. We describe each framework component starting from theory, such as definitions, measures and improving procedure and ending with a practical application, such as measuring a school class and improving social inclusion in the most isolated subjects. We built a device prototype for data collection activities by recording social interactions in frequency and duration. After defining a case study and on-field data collection, we shape the network analysed to identify the most interesting students and network structure properties. In the case study, we identify the students who are most isolated, most popular, and most frequently bridges for students' interactions. Considering how much the network is heterophile or homophile by interaction degree, we connect the most isolated with the most popular student through a bridge student, thus increasing the social inclusion of the least included student.

Contents

List of Figures

Chapter 1

Introduction

1.1 Context

Diversity, Equity, and Inclusion (DEI) are words becoming very popular in many fields, such as the workplace, academia, and medicine. We know that diversity, considered a cultural advantage, is now an integral part of modern and industrialized societies [35]. While equity guarantees equal opportunities in the application context, we focus our study on social inclusion, which improving is the desired result [29, 22, 23, 15]. Inclusion aims to create a united and cohesive environment where individuals are perfectly integrated [28]. The European Union has studied the effect of inclusion and exclusion in society and individuals for several years and promotes action to improve social inclusion [26, 18, 12, 10, 9, 8, 4].

1.2 Problem and Motivation

The problem of lack of inclusion in primary schools can have long-lasting effects on children's physical, social, and psychological development. When children do not feel included in their school environment, it can lead to problems such as low self-esteem, anxiety, and depression [25]. Additionally, exclusion from social activities and peer groups can exacerbate the issue, resulting in poor academic performance and limited opportunities for success later in life [28, 36]. It is, therefore, crucial to address this problem early on, providing an inclusive and supportive environment where all children feel welcome and valued [19].
In thesis aims to enhance social inclusion by presenting a comprehensive framework for measuring and enhancing social relations.
This framework consists of both theoretical and practical components.

The theoretical part delves into social theories and human behaviour, providing a basis for developing social relations measuring methods using network analysis measures to determine the level of inclusion and solution quality. Here we provide a procedure for improving social relations in the most isolated student pulling him into a new social relationship. On the other hand, the practical part explains how the data pipeline can be managed to achieve the desired results.

Here we also present our Social relation algorithm used in the data pipeline to construct the social relations network is discussed.

To apply and validate our framework in a real-world scenario, we develop and implement a device prototype for data collection. Then, in one primary school classroom scenario, we apply the procedure for improving social relations and report the final results.

Chapter 2

Conceptual framework

This framework aims to improve people's social relations by changing their social interaction habits. We use sociology with a mathematical approach and engineering to achieve this. Sociology explains how face-to-face interactions are defined and occur; Network theory explains how elements are linked to each other by sharing a specific property and how to improve social inclusion; and Engineering records social interactions.

2.1 Theory

This section provides social, network and measures definitions, which are the pillars of our framework for recording and measuring social relations. Inclusion improvement procedures and requirements for recording devices are also supplied.

2.1.1 Social Relations as the Sum of Interactions

The typical way to describe people's social relations is by using a network composed of people linked to each other by social relations. Defining social relations is a crucial step in our study; providentially, the social relation structure is already described in Sztompka's hierarchical of social behaviours, called "sociological hierarchy" (Figure 2.1) [7], which provides a hierarchical social structure where each level has unique properties. In Sztompka's model, social relationships are the sum of the social interactions between individuals over a period of time [38, ch.22]. Recordings of subjects' social interactions are needed to calculate their social relationships and infer their level of social inclusion; an aspect of social relationships.
Social interactions can be recorded in various ways [21, 16, 27], and the common

	Physical movement	Meaning	Directed towards others	Await response	Unique/rare interaction	Interactions	Accidental, not planned, but repeated interaction	Regular	Interactions described by law, custom, or tradition	A scheme of social interactions
Behavior	Yes									
Action	Yes	Maybe								
Social behavior	Yes	No	Yes							
Social action	No	Yes	Yes	No						
Social contact	Yes	Yes	Yes	Yes	Yes					
Social interaction	Yes	Yes	Yes	Yes	Yes	Yes				
Repeated interaction	Yes	Yes	Yes	Yes	Yes	Yes	Yes			
Regular interaction	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		
Regulated interaction	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
Social relation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Figure 2.1: Sociological hierarchy.

methods are the following:

* Observation: This method is based on observation of social interactions and recording behaviours or events. Observational data can be collected through video, audio, or written notes. It can provide rich and detailed information about social interactions, but it can be time-consuming and may be influenced by observer bias.

* Self-report: This method uses a standard questionnaire or survey to ask individuals about their social interaction experiences, attitudes, or behaviours. It is efficient and easy to administer, but it may be subject to social desirability bias and may not capture the complexity and nuance of social interactions.

* Social network analysis: This method analyses the network properties of a social network to understand social interaction patterns and processes. Social network analysis can provide insights into the social context of interactions, the roles and relationships of individuals, and the diffusion of information or influence. It requires specialized knowledge and software to conduct and interpret the analysis and may be subject to technology limit bias and not optimized software settings.

Identifying face-to-face social interaction requires the definition of social interactions, face-to-face, and social distances.

Social interaction is the process of reciprocal influence exercised by individuals over one another during social encounters. It involves verbal and nonverbal communication and can occur in various forms, such as face-to-face conversations and group discussions [20].
Face-to-face interaction is a social interaction carried out without any mediating technology [17]. It is defined as the mutual influence of individuals' direct physical presence with their body language and verbal language [1, p.15][3, p.50]. Face-to-face interaction allows people to communicate more directly and can improve mental health by reducing various mental illnesses, most commonly depression and anxiety [31, p.15].
Social interaction distances were first studied and defined by anthropologist Edward T. Hall in the 1960s. Hall's coined the term 'proxemics', which theories how people use space during social interactions in different cultures [2]. His research identified the four types of social interaction distances: intimate, personal, distance, and public distance, which are still widely used today.

These four types of social interaction distances (Figure 2.2) are described as follows:

1. Intimate distance: This is the closest distance between two individuals, typically ranging from skin-to-skin contact to about 46 cm apart. Intimate distance is typically reserved for close family members, romantic partners, and very close friends.

2. Personal distance: Personal distance ranges from about 46 cm to 1.2 meters from the individual. It is the distance at which people have personal conversations with family and friends or during business transactions.

3. Social distance: Social distance ranges from 1.21 to 3.7 meters from the individual. It is the distance at which people interact in social situations, such as parties, public places, or work.

4. Public distance: This farthest distance, from 3.7 to 7.6 meters or more. It is the distance at which people interact in public, such as in a lecture hall, a theatre, or a public park. At the close phase of public distance (3.7 — 7.6 meters), the types of nonverbal meanings that can be perceived vary dramatically.

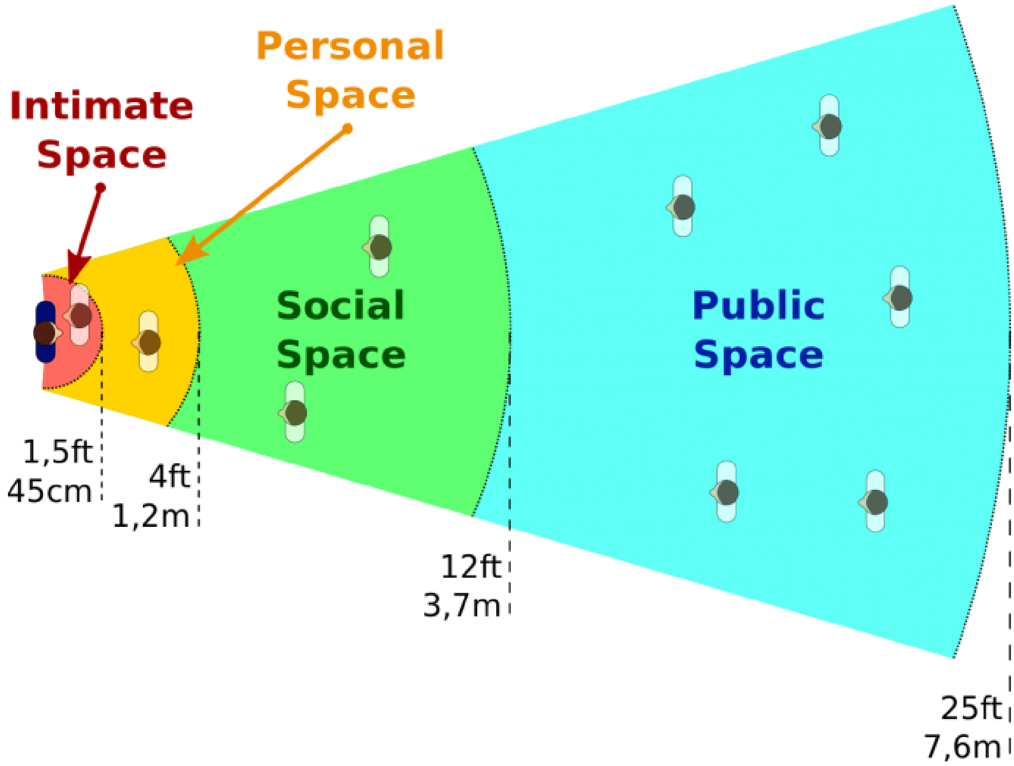

Figure 2.2: Personal Spaces in proxemics.

For our purposes, valid face-to-face social interactions (Figure 2.3) are within the 121 cm personal distance between the two subjects' fields of view (Figure 2.4). To determine the subject fields of view, we consider the direction of the upper body as the view's central point using the binocular range of 120 degrees, granting a comfortable interaction. [32].

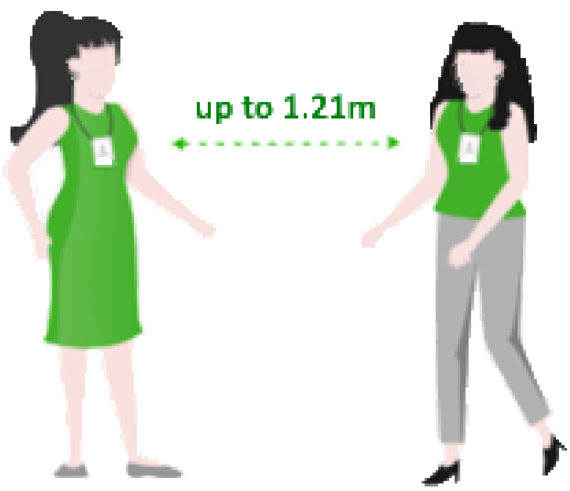

Figure 2.3: An example of a valid face-to-face interaction.

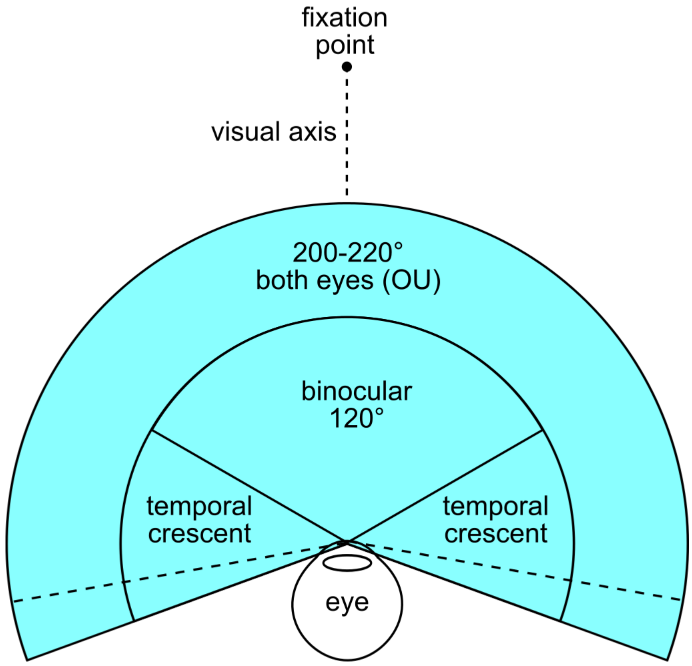

Figure 2.4: Horizontal fields of view both eyes.

The typical measurements of "Behavior Analysis" are made in frequency and duration [37]. However, there are no reference values in the literature, such as the minimum duration of social interaction, because they strongly depend on the study context. Based on the Sztompka model, social interaction is the sum of social actions in a social contact lasting over time, which can be described as follows:

1. Contact: the first face-to-face subjects seconds' having contact. It can evolve into an interaction or end in a brief contact;

2. Interaction: if the subjects continue a contact with each other continuously for a minimum amount of time.

A minimum amount of time in social interaction must be set by measuring behaviour in the study context. In this framework, we use devices to automate the observation method for recording face-to-face interaction, avoiding human bias due to observation and self-evaluation methods but remaining subject to false positives (FP), such as accidental positioning of people recorded as interaction, and false negatives (FN), such as unrecorded interactions lost due to the device settings and limits.

2.1.2 Network Analysis to Measure Social Inclusion

Since we want to study the properties of the social network nature and the characteristics of its elements and its relations, we use network analysis to identify the network nature, such as the element tends to have relations with its similar elements in degree and critical students, such as the most isolated, the most popular, the most frequent bridge in relations. Here, we define the subjects' interaction graph [24, ch.6], their properties, how to measure [24, ch.7] their inclusion and how to improve it.

Graph

The definition of the graph G is G = (V, E) with |V| vertices and |E| edges. Vertices are elements (students), and edges are interactions weighted by frequency and cumulative time of interactions. Vertices have a unique numeric label associated with a subject, and edges are weighted undirected for their social interaction's nature which involves two interacting subjects.
The weights for i-th interaction E_i are t_i for the cumulative time consumed in interactions and c_i for the count of the number of interactions.

Weighted degree centrality

The first simple measure is the weighted degree centrality, a variant of the degree centrality measuring the node's importance in a network [24, ch.7.1.1].
The degree centrality is a measure used in network analysis to quantify the importance of a node in a network based on the number of edges it has with other nodes. It is calculated as the number of undirected edges connected to the node. Nodes with a higher degree of centrality are considered more important or influential in the network because they have more connections and can reach more nodes directly. The degree centrality of a vertex v, for a given graph G, is defined as:

$$C_D(v) = deg(i)\, where\; deg(v) = \sum_{j=i}^{n} E_{ij}$$

The degree centrality is a simple and widely used measure of centrality, but it does not take into account the quality or strength of the connections, and the size and density of the network can influence it. The weighted degree centrality also considers the weights associated with those edges considering the strength of connections between nodes, making it a good choice for analyzing networks where the strength of connections is an important factor. It is commonly used in social

network analysis, where the weights on edges could represent the frequency or intensity of communication between individuals.
The weighted degree centrality of a vertex v, for a given graph G, is defined as:

$$C_{WD}(v) = deg_W(i)\, where\; deg_t(v) = \sum_{j=i}^{n} E_{ij}, p_i$$

where p_i *is* t_i *for weight as cumulative time or* c_i *for the interactions count.*

The inclusion troubles can be detected by ranking the node's degree centrality ascending or setting a cut-off. A good inclusion has the highest values in deg_t and deg_c, with low values otherwise. We prefer longer, fewer interactions than frequent shortest interactions, where the Dunbar number sets the upper bound to 150 as the maximum limit of significant relationships that an individual can maintain [6].

Eigenvector centrality

The second helpful measure is eigenvector centrality measuring the node's influence in a network [24, ch.7.1.2].
In eigenvector centrality, a node's centrality is determined by its connected nodes' centrality by summing the degree centralities of its neighbours. It is based on the concept of eigenvectors and can be calculated using an iterative algorithm known as the power method. This algorithm starts with an initial estimate of the centrality of each node, repeatedly updating these estimates based on the centrality of their neighbours until a stable solution is reached.

The eigenvector centrality of a vertex v, for a given graph G, is defined as:

$$C_E(v) = eig(v)\, where\; eig(i) = (\sum_{j\, \in\, neighbours(i)} deg(j)) - deg(i))$$

The weighted eigenvector centrality using c as weight, is defined as:

$$C_E(v) = eig(v)\, where\; eig(i) = (\sum_{j\, \in\, neighbours(i)} deg_c(j)) - deg_c(i))$$

The eigenvector centrality has applications in fields such as social network analysis, biology, and economics, where identifying key players in a network can provide valuable insights. In our application, the most important students have the highest eigenvalues used for evaluating how to connect them to other students who need more inclusion.

Betweenness centrality

The third measure is Betweenness centrality measuring a node's importance in a network based on the shortest paths that pass through that node [24, ch.7.1.7].

It is calculated as the number of shortest paths between all pairs of nodes in the network that pass through a particular node, divided by the total number of shortest paths between all pairs of nodes in the network. The resulting value is between 0 and 1, with higher values indicating greater betweenness centrality.

The betweenness centrality x_i is given by the formula:

$$x_i = (\sum_{sd} n^i_{sd}) \text{ for all shortest paths.}$$

where s is the source node and d is the destination node.

Betweenness centrality has many applications in various fields, including social network analysis, transportation network analysis, and power grid analysis. It is often used to identify key nodes in a network that are critical to the flow of information, resources, or power and can help to guide network planning and optimization efforts.
The nodes with the highest value have a more central role in the network; In our application, this measure is used to connect the lower node's degree centrality to the higher node's eigenvector centrality passing from a node with higher betweenness centrality. For instance, we want to connect an isolated student with another well-included student by choosing a student bridge that acts as a walking point for all the shortest paths more often because it can open new shortest links.

Assortativity coefficient

The assortativity coefficient is a network measure that describes the tendency of nodes in a network to form connections with other nodes with similar properties or characteristics.
It measures the degree of homophily or similarity among the nodes in a network and can be calculated using the Pearson correlation coefficient between the degrees of connected nodes [5]. It is defined as the difference between the observed degree correlation and the expected degree correlation in a network, divided by the maximum possible difference.

The assortativity coefficient formula is the following:

$$r = \frac{\sum_{ij}(A_{ij} - \frac{d_i d_j}{2m}) x_i x_j}{\sum_{ij}(d_i \delta_{ij} - \frac{d_i d_j}{2m}) x_i x_j}$$

where d_x is the degree of node x, m is the number of edges, and

$$\delta_{kl} = \begin{cases} 1 & if\ k = l \\ 0 & otherwise \end{cases}$$

The assortativity coefficient ranges from -1 to 1, with positive values indicating assortative mixing, negative values indicating disassortative mixing, and values close to zero indicating random mixing.
The assortativity coefficient finds applications in various fields, including social network analysis or biological networks helping to reveal patterns of functional similarity or evolutionary behaviour.
A social network tends to maintain its natural characteristics; In our application, the assortativity coefficient is used to verify if the selected destination node with higher eigenvector centrality is a choice in line with the network's nature.

K-core

The k-core is a network analysis technique to identify the most densely connected subgraphs (or "cores") of a network. In a k-core, every node is connected to at least k other nodes within the subgraph [24, ch.7.2.2].
The k-core is useful to reveal a network's most cohesive and tightly-knit subgraphs. These subgraphs can provide insights into the organization and structure of the network, as well as the relationships between nodes.
To construct a k-core, the nodes with the lowest degree are removed from the network one at a time until all remaining nodes have a degree of at least k. The resulting subgraph is the k-core of the network.
It is used to identify important nodes in the network. Nodes with a high degree in the k-core are often critical to the structure and function of the network. Removing these nodes can lead to the fragmentation or collapse of the network.
The k-core metric is a useful tool for understanding the structure and function of networks. It can be applied to many network types, including social, biological, and technological networks.
Our study set k to a high value to find and remove the best reachable nodes. The remaining unconnected nodes in the network are potentially the nodes with inclusion trouble. This technique can be used to take preventive action for the nodes (students) connected with other nodes by a few entry points.

2.1.3 The Procedure to Improve Social Inclusion

This procedure aims to increase the inclusion of the most isolated student, whom we have seen essential to a person's life prospects. After identifying critical students, we create an indirect link between the most isolated and the most popular students through an exogenous intervention based on finding a bridge student who can connect the two persons. Hence, we force the new link creation, in line with the network's nature, hoping to trigger an endogenous process that improves the social inclusion of the most isolated student, creating new connections for them.

Before starting, we measures network properties, such as the number of nodes and edges, average node's edge, node characteristics, network diameter, and average path length, to better understand the network's nature.

The procedure to measure and improve social relations is the following:

1. Choose the most important network's characteristic and calculate the corresponding network assortativity coefficient to identify the network's nature;

2. Select the most isolated student using the weight degree centrality ranking by descending the duration and the number of interactions where the first choice is the interaction of the shortest duration and, in case of ties, the quantity of the interactions;

3. Select the most popular student, not already connected with the most isolated, using the eigenvector centrality ranked in descending order;

4. Identify the shortest paths from the most isolated student to the most popular, selecting the shortest with the sum of the passing node's betweenness centrality with the highest value, then modify the network to create the new connection from the most isolated to the most popular one.

5. Inspect the k-core isolated student comparison in the original and modified network to quantify the quality of the solution;

6. Compare the original and modified network's assortativity coefficient to verify if the changes respect the nature of the network and quantify the solution quality and how much has changed.

In step 2, an alternative way to detect potentially isolated students is to remove the highest k-core nodes from the network and rank the remaining node by degree;

the most isolated students in the sub-network are potential candidates for social inclusion improvements.
In step 3, to reduce the impact by respecting the nature of the network, to minimize the strength of intervention, we connect the most isolated student with a popular student with minor degree centrality by maximizing the different neighbour students between the two nodes.
This procedure can be repeated several times on isolated students regarding the short duration of the interactions or too many interactions that exceed the number of Dumbar.

2.1.4 Data Collection Device

The device aims to record the social interaction in frequency and duration between identifiable subjects, which we achieve using a development board and radio waves, based on the assumption that the more powerful is the revealed signal and the shortest is the distance between two devices [30].
Chapter 2.1.1 highlights our objective of identifying face-to-face social interactions as a field of view contacts up to a distance of 1.21m for a minimum amount of time, depending on the context. Several options are available to identify frontal position accurately, such as using a directional antenna to hone in on the exact location or transforming an omnidirectional antenna into a directional one through a reflective box. In Chapter 2.1.3, which describes our device, we relied on the latter technique, utilizing a box that reflects waves on all sides except the front to pinpoint the frontal people positioning.
Here, we define our empirical approach for identifying the signal strength threshold on valid interactions. Below we describe this method based on test scenarios and its validation to find signal strength thresholds, which strongly depend on device properties. These scenarios are static or dynamic, differ in subjects' positions and must be recorded by the device for final considerations. The tests are performed by executing scenarios statically to evaluate the device's functionality and determine a first signal strength threshold. On the other hand, dynamic validation, performed by executing the same scenarios dynamically, is used to fine-tune the signal strength obtained from the static ones. By utilizing this approach, we can determine the scenarios' optimal signal strength threshold.
The scenarios are the following:

1. Scenario A: Two frontal subjects;

2. Scenario B: A frontal subject and another lateral frontal that looks the first one obliquely at 45°;

3. Scenario C: Two subjects back-to-back;

4. Scenario D: Two subjects in a row;

5. Scenario E: Two subjects side-by-side;

6. Scenario F: One subject facing forward and the other sideways.

Based on the face-to-face interaction definition, the valid social interaction scenarios are A and B (Figures 2.5 and 2.6); invalid otherwise.
The static scenarios must be performed by placing the two devices at a distance of 1.21m on a table and registering timestamp and signal strength values. On the other hand, the dynamic scenario must be designed to reproduce all the scenarios sequentially within a timeline.
The final consideration depends on device technology; however, this approach and all recorded information are valuable to identify a reasonable threshold.
For instance, in the case of Bluetooth technology we examined, the values collected through static and dynamic tests fluctuate due to several factors related to the technology employed. Therefore, it is advisable to use the average of static and dynamic values of the only recorded high spikes lower values (above scenario values average) in scenarios A and B as the best choice for determining the final signal threshold.

2.2 Practice

This section provides all the necessary practical clarifications and steps to apply our theoretical framework. Here, we use the Theoretical part to define the entire process, from data collection to the final result we aim for in our study. At this stage, we refrain from describing a specific device. Still, we will provide a detailed account of the device we utilized in our case study later, which is outside the conceptual framework.

2.2.1 Data Pipeline

Establishing a well-structured data pipeline allows us to effectively manage and outline each stage of data processing, from the initial gathering of raw data to its

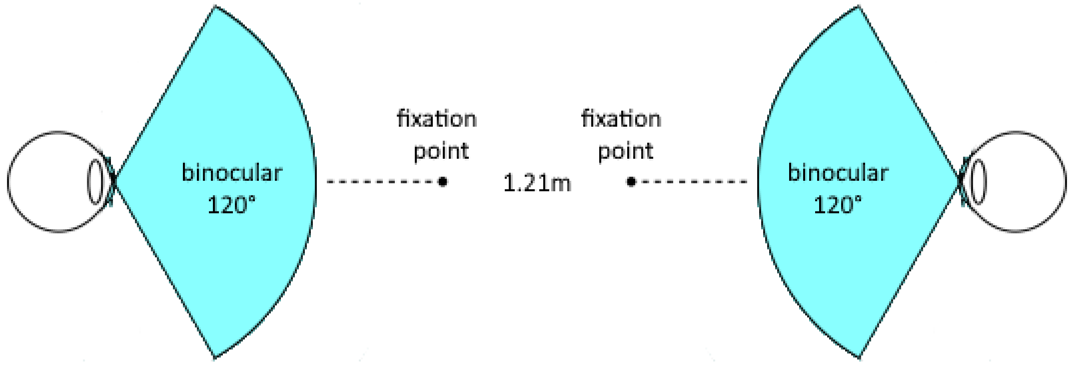

Figure 2.5: Scenario A.

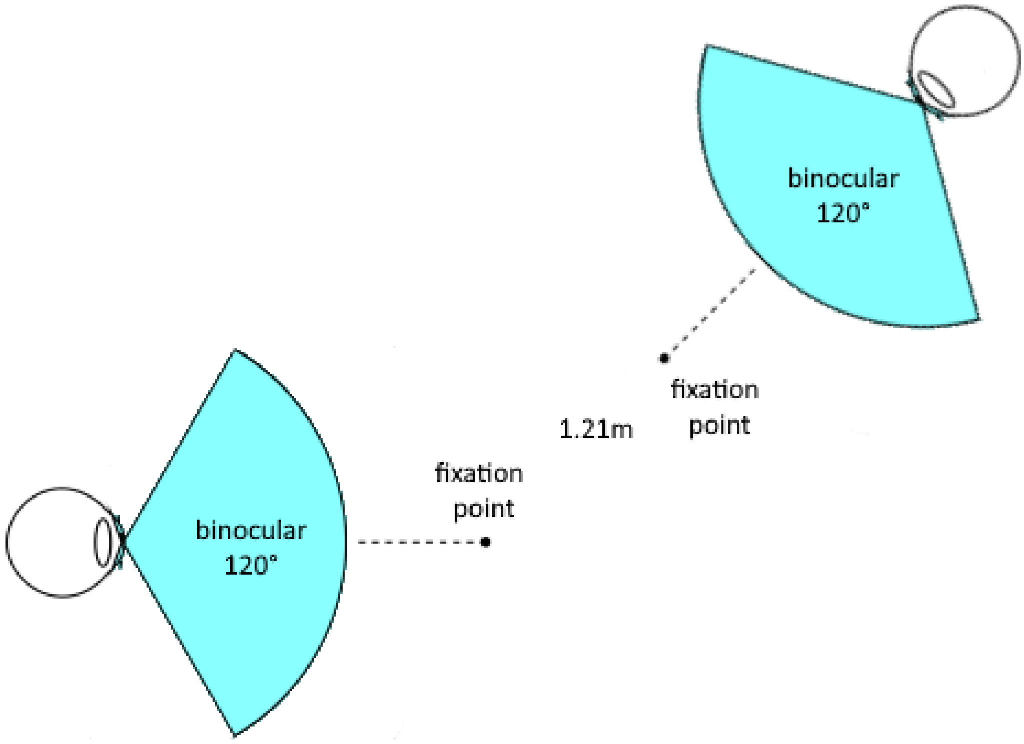

Figure 2.6: Scenario B.

conversion into valuable information used in reporting and decision-making [14]. The data pipeline is tailored to a specific application's needs and requirements and varies slightly according to the context. Figure 2.7 depicts the custom-designed data pipeline for our study, showcasing the various input, processing, and output stages involved.

Figure 2.7: Data Pipeline.

Data collection

During this phase, IoT devices are crucial in collecting extensive raw data, including device identification, timestamp, and radio wave signal strength. The information gathered from these devices ensures a vast data pool for further analysis and decision-making. This data helps to understand people's behaviour patterns and preferences.
The recording activity can be done in the two following ways:

1. On-site: logging raw data inside the device;

2. Off-site: logging data externally, such as a server.

Data integration

The dataset created during this phase is crucial in the subsequent analysis steps. Here, we get data from various sources, which we collect and combine, ensuring uniformity across the board. The preferred approach can be influenced by the data collection method used. For instance, if data is acquired on-site, we can concatenate the files to create a single file that contains all the necessary details. On the other hand, if the data is collected off-site, the server is responsible for organizing it and preparing it for analysis in the following stages.

Data preprocessing

During this critical phase, the combined dataset is meticulously analyzed and transformed into a complex network of social relations described in chapter 2.1.2. The processing of the data leads to the creation of a comprehensive and fully realized network, which not only captures the essence of the social relations within the dataset but will be used in the following stages for revealing patterns and connections that were previously unseen. It is essential for understanding the network's social structure and element relationships.
The conversion process is performed with the following steps:

1. Data binarization: converting the signal strength values into 0 or 1 for contact detecting using the threshold identified during the application tests described in chapter 2.1.4;

2. Social relation computation: applying definitions described in chapter 2.1.1 we calculate people's social relations algorithmically;
   This computation can be done in the two following ways:

   (a) Online: unknowing all data before algorithm execution;

   (b) Offline: knowing and using all recorded data for result calculation.

   In Chapter 2.2.3, we propose our online algorithm.

3. Network construction: building the network in the proper format for data visualization, such as Gephi format.

Data visualization

After building the network, we transform complex datasets into easy-to-understand visualizations to get a first insight that helps us understand patterns, trends and relationships within the data so that we begin to understand the structures of social relations. Data visualization allows us to see the big picture of the social interaction network, providing insight into how people are connected. This information can be used to identify key players in the network, understand social trends and monitor changes.

Data analysis

By applying the mathematical approach outlined in chapter 2.1.2, we can comprehensively examine the network's structure and dynamics. This approach encompasses measures of centrality, clustering, and community detection, all of which shed valuable light on the most significant nodes within the network, the level of interconnectivity between closely-related nodes, and clusters. These analytical tools, in conjunction with steps 1-4 of the procedure described in 2.1.3, allow us to achieve our goal of measuring and improving the inclusion level of people.

Data evaluation

After applying the inclusion improvement procedure, we can assess the effectiveness of the process. A crucial factor in evaluating the results is comparing the k-core ranking of the most isolated node against the network changes that occur. It is vital to ensure that these changes align with the nature of the network, as described in the assortativity coefficient metric in steps 5 and 6 of the chapter 2.1.3 procedure. By evaluating these factors, we can determine the success of the inclusion improvement process and make any necessary adjustments, such as selecting a different effective path.

Data reporting

In conclusion, we can communicate our findings in a more conversational and understandable style with the help of network visualization, illustrations and tables. Such aids can simplify and streamline the process of conveying our results to the intended audience. Subsequently, the relevant staff will absorb and interpret the information before implementing the optimal solution based on our gathered insights.

2.2.2 Signal Strength for Contact Detecting

In the field of telecommunications, signal strength is a critical measure, particularly in radio frequency engineering. It is calculated by assessing the transmitter's power output as received by a reference antenna stationed at a distance from the transmitting antenna [33]. The signal strength parameters vary depending on whether the transmission is conducted through high or low-powered systems. For low-power systems, which are more compatible with our proxemics measuring, the signal strength is usually expressed in decibel-milliwatts (dBm), a widely standardized unit of measurement.
In telecommunications, typically, there is a node that acts as a broadcaster and others as receivers. However, in our study, we aim to use multipurpose devices that perform the two activities in parallel.

`BROADCAST:` cyclically sending a signal over time;

`SCAN:` cyclically reading the received signals.

In practice, our data collection goal is to have one device transmit its signal to all other units while also recording the signals it receives from others at predetermined time intervals. The approach of selecting a time interval is crucially dependent on the specific study that is being conducted and serves to determine the smallest unit of measurement in terms of time. Additionally, it has a significant influence on the volume of data collected and the device's energy consumption. In Chapter 3, we will specify and apply a standardized time interval of one second to our device. Contact detection is reached by converting the intensities of the recorded signals into binary values.

2.2.3 Social Relations Computation Online Algorithm

Here we propose our online algorithm to transform the time-sequential contacts between two devices into a social relationship that has as its property the frequency and the cumulative time of the interactions. In our development, we use the social relation definition described in Chapter 2.1.1 as the sum of the interactions identified as prolonged contact over time without recording quick contacts that do not evolve into interactions within a time limit defined based on the case study.

In summary, social interactions are identified through protracted contacts that meet specific criteria, including face-to-face positioning and maximum distance. The device and signal strength ensure these criteria are met, yet we still need to determine when contacts transition into interactions and how to calculate their duration. Our algorithm aims to track the establishment and course of these interactions by utilizing global variables.

First, we establish our interest in frequency detecting the presence of contact by defining the TIME_INTERVAL variable. Another crucial aspect is determining the duration after which a contact transforms into a social interaction with the START_TIMEFRAME variable. In this time frame, multiple contacts must occur within a specific time limit defined by the START_TIMEOUT variable to develop into an interaction. Once established, it will end when the time limit specified by the END_TIMEOUT variable is surpassed.

Before describing the algorithm more technically, we summarize the definition of the algorithm's global setting variables used for detecting interaction over time.

`TIME_INTERVAL` specifies up the reading frequency;

`START_TIMEFRAME` sets a minimum timeframe to transform the contact into an interaction;

`START_TIMEOUT` sets how long two subjects can stop contact without breaking its duration; once this limit is exceeded, the contact end without becoming interaction;

`END_TIMEOUT` define up a minimum time for the receiver to detect a contact to continue the interaction; once this limit is exceeded, the interaction ends.

The following pseudocode algorithm for logging social interaction uses start and last local variables to track the starting time and the last time the contact was seen. The algorithm performs cyclic operations within a specific interval, reading the binarized contact with the device. On affirmative detection, the algorithm updates the last contact detection time and sets the start variable to the current time if it is a new contact. On the other hand, if the contact is not confirmed, the algorithm calculates the duration by subtracting the current time from the start variable and the timeout time by subtracting the current time from the last variable. If the timeout exceeds the window for establishing an interaction, the contact is deleted, and the algorithm moves on to the next detection. Alternatively, if the interaction has taken place and the contact detection time limit is exceeded, the algorithm cancels it and add interaction data to the relation.

```
start = last = None

while True:
   contact = read binarized device contact

   if contact is True:
        last = NOW
      if start is None:
         start = NOW
   else if start is not None:
      duration = last - start
      timeout = NOW - last

      if duration < START_TIMEFRAME and timeout > START_TIMEOUT:
         start = last = None
         continue

      if timeout > END_TIMEOUT:
         start = last = None
         update relation

   wait TIME_INTERVAL
```

Here, the algorithm application example using the following parameters settings:

`TIME_INTERVAL` : 1 second

`START_TIMEFRAME` : 5 seconds

`START_TIMEOUT` : 2 seconds

`END_TIMEOUT` : 5 seconds

Figure 2.8 shows a first contact that occurs on second five, which sets a time frame of 5 seconds (set by START_TIMEFRAME) indicated by the vertical red line; during this period, further contacts must take place with a maximum cadence of two seconds (set by START_TIMEOUT). Upon reaching the 5 seconds by renewing the contacts, it turns into interaction; otherwise, it is discarded. Then in Figure 2.9 and 2.10 show renewed contact and the transformation of the contact into interaction. Furthermore, once the interaction has been established, it is extended until another contact occurs within 5 seconds (END_TIMEOUT), which renews it. Figure 2.11 and 2.12 show renewed interaction and the final result.

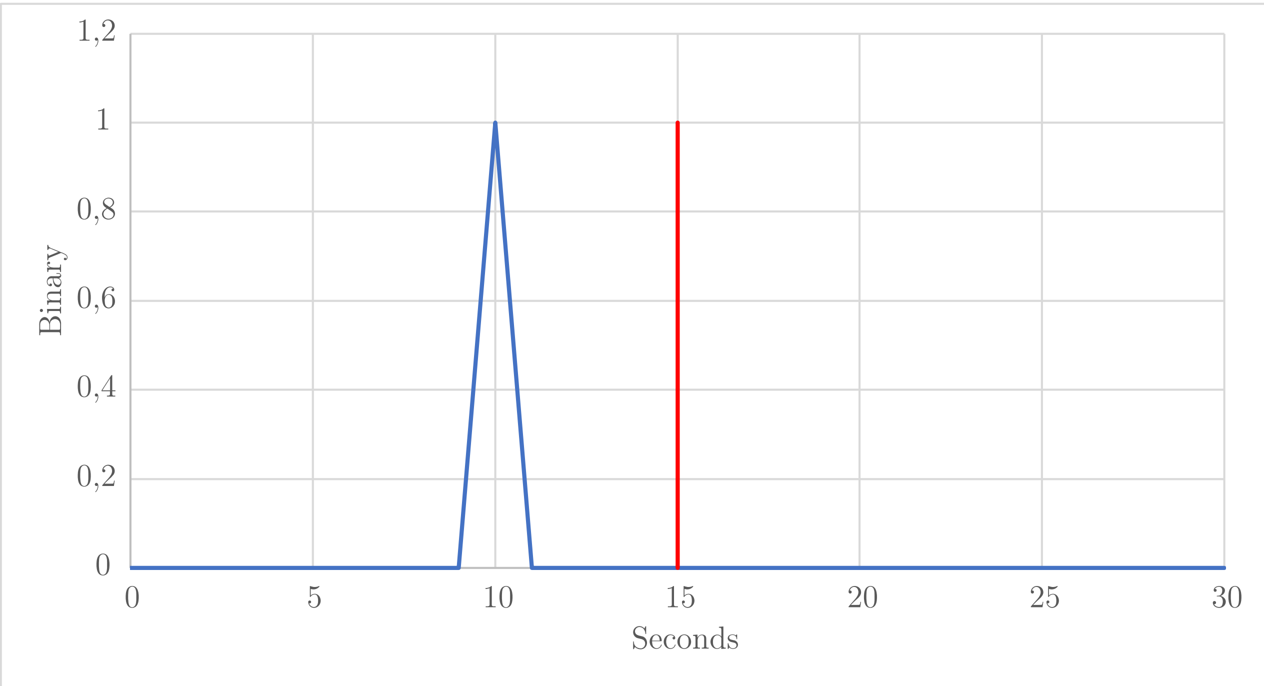

Figure 2.8: First Contact.

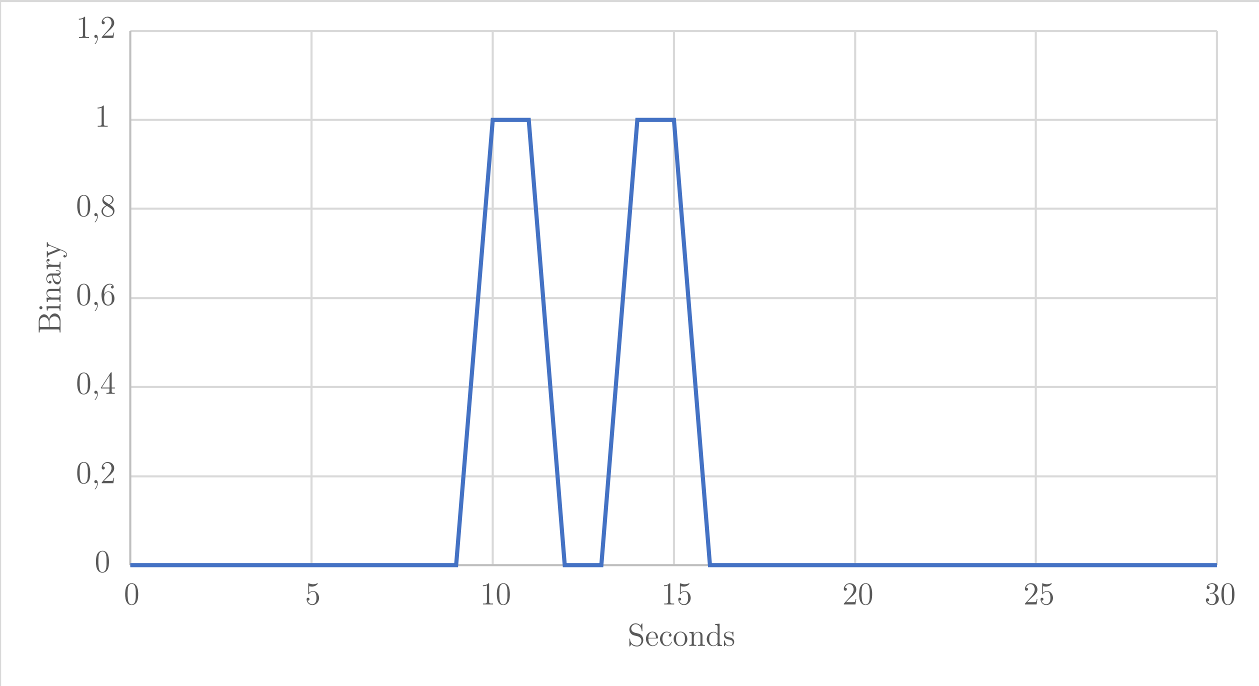

Figure 2.9: Renewed Contact.

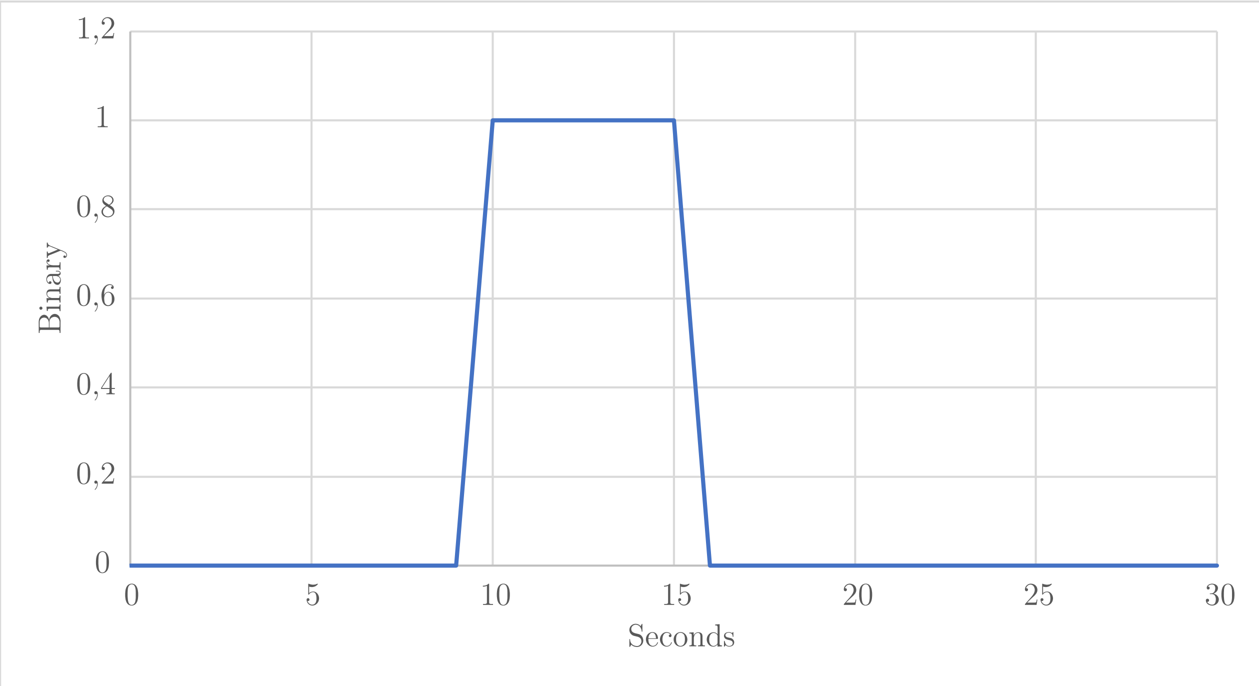

Figure 2.10: Result of contacts transformed into interaction.

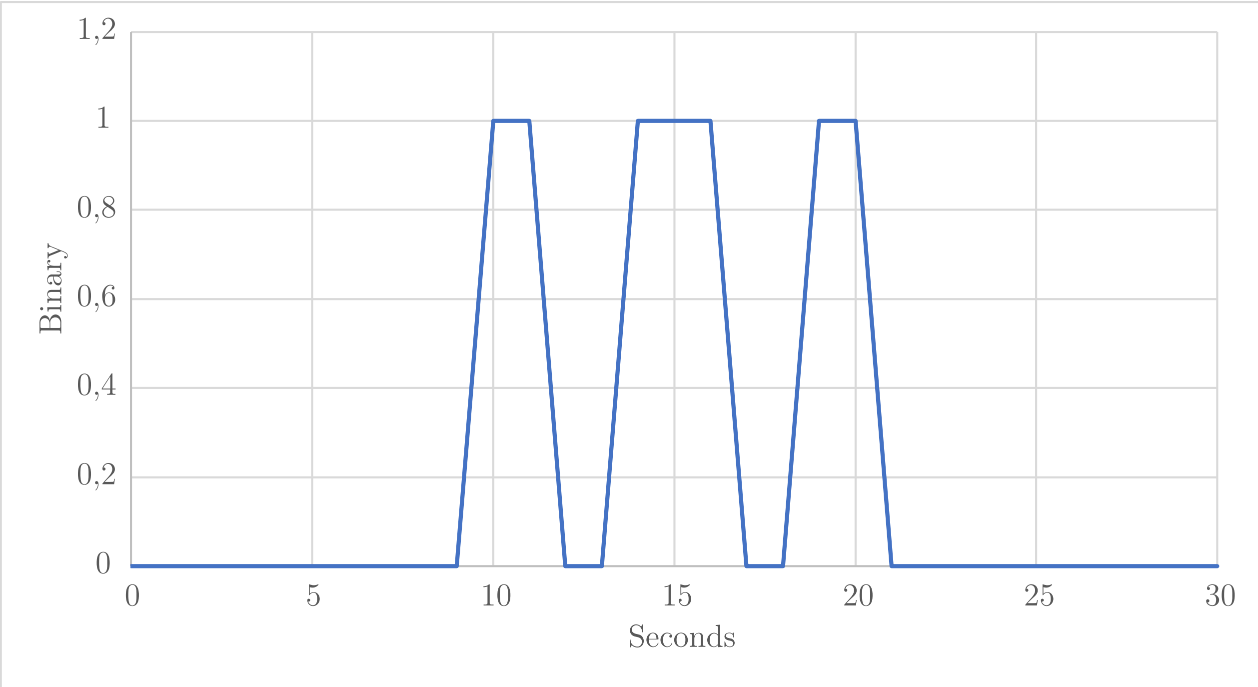

Figure 2.11: Renewed Interaction.

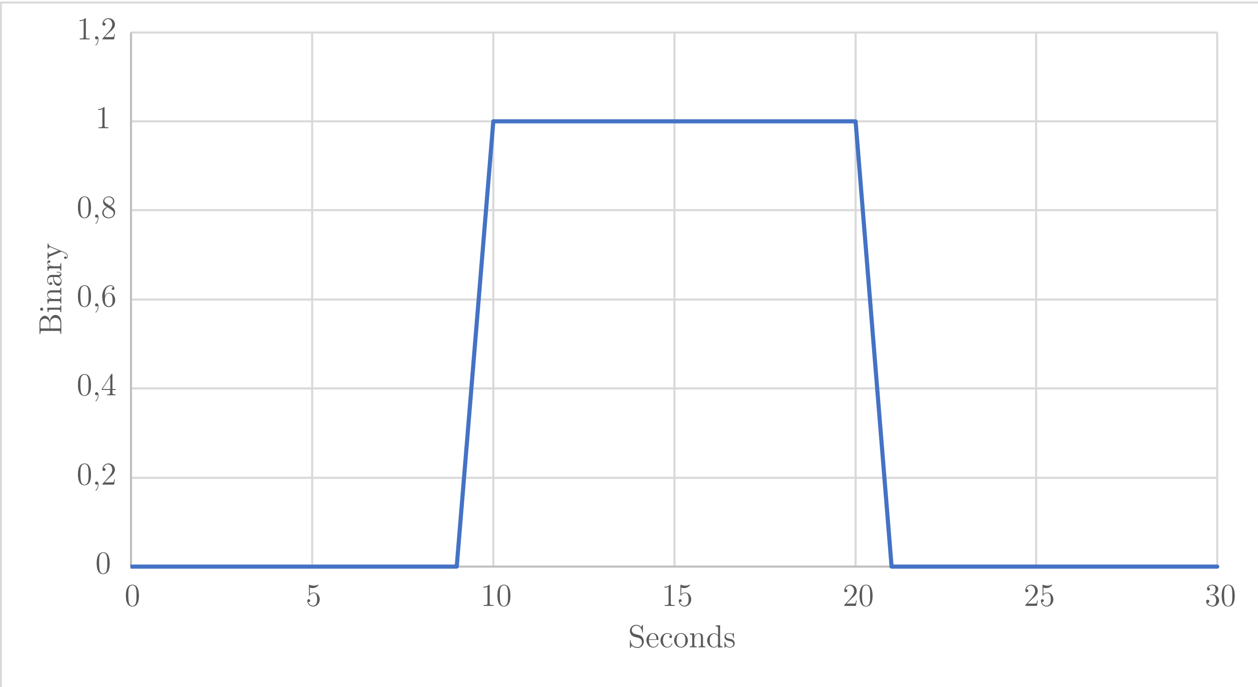

Figure 2.12: Interaction Final Result.

Chapter 3

Device Prototyping

In this Chapter, we explain how we developed the prototype contact tracing device between people for our study purpose, which we will then use in a real case described in the next Chapter. The first phase of device development involves identifying the project requirements and finding a proper solution. This phase includes selecting the appropriate hardware and its characteristics; this meets with the Practical implementation of the Theory described in Chapter 2.2, such as identifying face-to-face contact through a reflective device case, signal strength monitoring via Bluetooth and data storage management, and also introducing the device powering not yet discussed. Once the hardware and specifications are established, we move on to the next phase, which involves the development of the source code. To ensure that the device meets our requirements from a software perspective, we perform rigorous testing and software calibration. These tests allow us to fine-tune the device and ensure it works efficiently and effectively. Ultimately, the ultimate goal is to create a reliable and accurate contact tracing device.

3.1 Requirements and Solution

In order to create a practical device prototype, it is necessary to establish several critical requirements that the prototype must fulfil. This initial phase determines our envisioned device's hardware components and software functionalities. Once we have a clear understanding of these requirements, we move forward with selecting a suitable development board that can be used to implement the prototype. Additionally, we recognized the need to create a protective case for the various components of the device and make Bluetooth directional. Finally, we choose the most appropriate language and programming environment, talk about using

Bluetooth for contact tracing, how we store data in the device's memory, and our choice of the power supply to ensure the prototype's functionality for a day of experimentation.

3.1.1 Requirements

This section will outline the necessary device requirements for efficient contact tracing recording. Radio waves are the primary technology required for successful contact tracing, which allows for accurate proximity estimation. Additionally, devices must possess memory capabilities to store contact information. In addition to these requirements, we have set further constraints to ensure that the device is easily programmable, capable of being charged, and has low-cost consumption for initial prototyping purposes. By adhering to these device requirements and constraints, we can develop a first contact tracing solution that is both efficient and cost-effective.

Here, we summarize all discussed device requirements:

- Radio waves technology: enabling the device's proximity communication with each other to track contact events;
- Memory capabilities: to store social contact records;
- Easy programmable: to program with a high-level language with memory management;
- Easy to charge: easy adding one battery to power the device;
- Low cost: to make studying possible with a limited budget;
- Low consumption: to ensure at least one day's device operation on a limited battery.

3.1.2 Hardware

Here, we detail our implementation for our contact tracing device, focusing specifically on how we met the identified requirements in terms of hardware and its functionality. After evaluating various radio wave technologies for proximity calculation, we ultimately decided to utilize Bluetooth Low Energy (BLE) technology which we will explain better later, because it can cover the distances we identified in Chapter 2, from the personal distance and beyond. We conducted thorough market research to find the ideal development board with Bluetooth LE technology that satisfied our requirements. Our research led us to select the ESP32 development board, created by Expressif System, which offered the necessary features such as Bluetooth LE, integrated flash memory, ease of programmability, and low cost and power consumption.

Here we summarize how ESP32 covers our requirements:

1. Radio waves technology: supports Wi-Fi (802.11 b/g/n) and Bluetooth (Bluetooth Low Energy and Classic Bluetooth) connectivity, making it suitable for various IoT applications;
2. Memory capabilities: includes up to 4 MB of flash memory, providing enough space for firmware, data storage, and program execution;
3. Easy programmable: can be programmed using a variety of programming languages and environments, including Arduino and MicroPython, making it accessible to a wide range of developers;
4. Easy to charge: provides a USB connector for device programming and powering;
5. Low cost: offers a cost-effective solution for those on a budget who require similar capabilities to other high-end microcontrollers such as Raspberry, STM32, and Particle. Compared to its predecessor, the ESP8266, the ESP32 is more powerful and has a larger memory size while maintaining simplicity and low power consumption;
6. Low-Power Consumption: offers low-power consumption with various power modes, including a deep-sleep mode able to consume as low as 5 µA.

3.1.3 Box Case

Our case device serves several essential functions, including protecting the components, concealing them, and Bluetooth directionality using a reflective box, as outlined in Chapter 2.1.4.
To achieve a cost-effective and lightweight prototype, we used cardboard as the primary material during construction. We selected a sturdier cardboard weighing 350-400 grams to ensure durability and resilience for the most demanding usage conditions. As shown in Figure 3.1, we marked the area where we had to apply the

surface with aluminium tape to absorb and reflect any signals from behind or sides. Aluminium tape with adeguate thickness is a cheap, efficient and practical solution to achieve Bluetooth directionality [13], which we will ensure its effectiveness by static tests in Chapter 3.3.1.

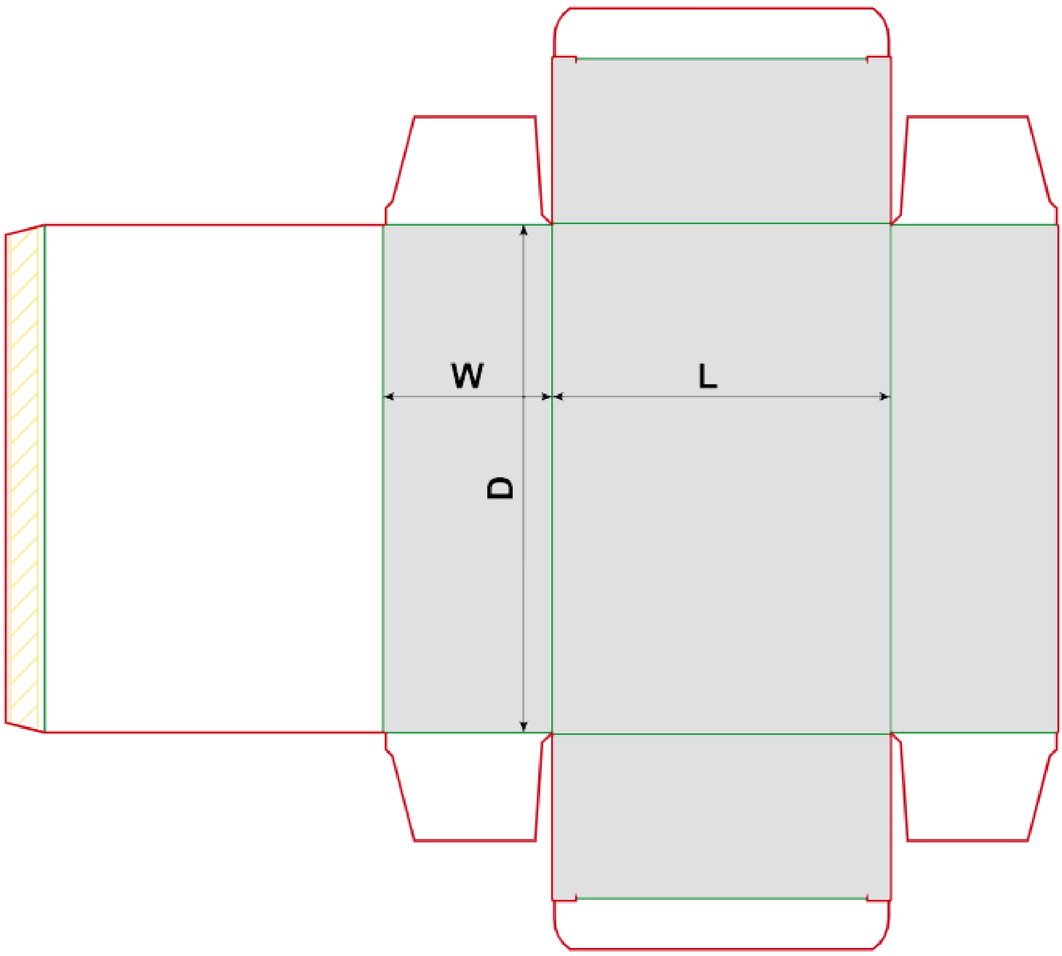

Figure 3.1: Cardboard Box Case (L=83mm, D=165mm, W=35mm).

Additionally, we reinforced the corners with adhesive tape to improve the case's overall durability. After placing the development board and battery inside, we filled the empty spaces with foam sheets, to protect the components inside from compression. The following figures below represent our cardboard box case from various angles, showcasing how we positioned the components and filled any excess space with foam sheets for added protection.

Figure 3.2: Box Case: front view.

Figure 3.3: Box Case: back view.

Figure 3.4: Box Case: inside view.

We conclude by adding a height-adjustable lanyard fixed to the top of the box, making it easy to wear around the neck.

3.1.4 Programming

This section explains our MicroPython choice for device programming, a subset implementation of Python designed to run on microcontrollers and other resource-constrained devices. It provides minimal Python features and modules while maintaining compatibility with the Python language and syntax. In this subset, non-essential features for embedded systems, such as multi-threading, are unavailable. The reasons that make us comfortable using MicroPython are the following:

1. High-level programming: MicroPython is a high-level, interpreted programming language, easy to write, read, and debug code compared to lower-level languages such as C/C++;

2. Python libraries: Python is a popular programming language with a large community and many available libraries, making it easy to find helpful and useful resources;

3. Interactive environment: MicroPython has an interactive programming environment for running and testing code directly on the ESP32 without compiling and uploading code each time;

4. Abstraction from low-level hardware details: MicroPython abstracts many of the low-level hardware details of the ESP32, allowing developers to focus on the high-level logic of their applications.

After choosing MicroPython, we connect to the board via USB and upload the firmware using the esptool bootloader[1].

3.1.5 Bluetooth for Contact Detecting

As discussed in Chapter 2.2.2 about signal strength for Contact tracing, we choose the Bluetooth BLE at 2.4Ghz as a low-power technology for proxemics calculation. The range of Bluetooth BLE 2.4 GHz frequencies depends on several factors, including the transmit power, the physical environment, and obstructions or interference. The signal strength of Bluetooth BLE devices ideally ranges up to 100 meters (328 feet) but decreases rapidly with distance, as described by the Friss transmission equation [30]. However, interferences or environmental obstacles such as walls, furniture, and other obstacles that block or absorb radio waves can

[1] https://docs.espressif.com/projects/esptool/en/latest/esp32/

significantly reduce this range. For example, in a real scenario, a device can transmit data reliably at distances of less than 10 meters (33 feet) in a crowded room. Overall, the effective range of Bluetooth BLE devices can vary widely depending on the specific use case and environment. It is essential to test and optimize the wireless connection in real-world scenarios to determine a Bluetooth BLE device's maximum range and signal strength, as we have done in Chapter 3.

For Bluetooth management, MycroPython provides an interface to the Bluetooth controller on-board through the application programming interface (API) intended to match the low-level Bluetooth protocol and provide building blocks for higher-level abstractions [34].

These APIs implement the telecommunications system described in Chapter 2.2.2, where there is a node that acts as a broadcaster and others as receivers, performing the following activities:

1. Broadcasting: transmitting data from a Bluetooth device to all other devices in the range that are listening for broadcasts;

2. Scanning: cyclically reading available Bluetooth-enabled devices. It allows a device to discover and connect to other in-range Bluetooth. The scanned signal provides the device's MAC address and the Bluetooth RSSI, which measures Bluetooth's signal power scanned from a device, which we use for proximity tracking.

Our devices are designed to perform cyclically broadcasting and simultaneously scan activities of Bluetooth's signals, and we will achieve this concurrency using APIs and through source code development which we will show next.

3.1.6 Contact Data Storage

Due to our budget constraints and specific requirements, we decided to collect on-site data, as outlined in Chapter 2.2.1. Logging the data to the device's flash memory proved a cost-effective solution, particularly since we confirmed that the memory space was sufficient to store all contact information generated from a day's experimentation. Our purchased ESP32 devices were equipped with a 4MB flash memory, which was utilized for Micropython and source files, leaving approximately 2MB of unused space. By allocating five bytes for each contact entry, which consisted of 1 byte for device ID, 1 byte for RSSI, and 3 bytes for timestamp, we can effectively track classmate contact devices for more than one day while managing our data storage space efficiently.

3.1.7 Power Supply

In order to power the development board for stand-alone use, it was essential for us to determine the optimal voltage and battery capacity. The manufacturer specified an operative voltage of 5V and a regular CPU consumption of 15mAh. Using a bench power supply, we determined that the minimum operative voltage was 3.3V, with a consumption of 80mAh during standard usage with the Bluetooth BLE antenna turned on. To ensure the device could operate for an entire school day of six hours, we determined it required a minimum battery capacity of 480mAh. In order to achieve a practical and time-saving solution, we decided to connect a power bank to the device's micro-USB port using a 18650 lithium battery with a capacity of 1500mAh. This battery type is widely used in portable electronics due to its small size and relatively long lifespan. Its compact dimensions, measuring 18mm in diameter and 65mm in length, make it an ideal choice for our prototype application, where a small, cheap, and lightweight battery is required. By selecting this option that exceeds our minimum operating requirements, we were able to find a solution that met our needs while staying within budget.

3.2 Software

In this paragraph, we present our software implementation of a contact tracing device which uses the online algorithm for calculating social relations. Here, working on a prototype, we want to confirm the algorithm's effectiveness, so we split the algorithm into two parts and implemented an offline version. In the first part, the device's responsibility is only to perform data collection recording contact information, and after, we apply the algorithm already described in Chapter 2.2.2 to infer social interaction. With access to both the original contact data and the interaction data, we can ensure the accuracy of the algorithm. In the final online version, we can directly record interactions instead of contacts, as outlined in the pseudocode we previously described.

3.2.1 Contact data collection

This section explains how we perform data collection using the Bluetooth module to detect contacts and the operating system module for data storage. Our programming methodology involved cyclically performing Bluetooth advertising and scanning operations every second and storing data through the operating system module, as explained in detail in Chapter 3.1.5. Additionally, to improve system performance and capture only relevant data, we compiled a list of the Mac addresses of the enabled recording devices and utilizing a buffer, we store data only every 60 seconds.

After the Mycropython boot process, the following source code contained in the “main.py“ file for monitoring and logging Bluetooth devices contact starts. For the understanding of the code, a step-by-step detailed explanation works are provided.

```
import bluetooth
import os
import time

# list of monitorable devices MAC addresses
ids = { b'\xe0Z\x1b_\xe66': 1, b'\xe0Z\x1bu\x9a\xe2': 2, ... }
```

First of all, the code imports the necessary libraries, including the Python “bluetooth” module for BLE communication, the “os” module for file handling, and the “time” module for measuring time intervals and defines a dictionary called ids which contains the MAC addresses of the monitorable BLE devices along with their corresponding IDs.

```python
class BLE:
    def __init__(self):
        self.ble = bluetooth.BLE()  # Bluetooth istance
        self.ble.active(True)  # Enable Bluetooth
        self.ble.irq(self.ble_irq)  # Bluetooth Event Handler
        self.isFirst = True
        self.logBuffer = bytearray([])
        self.logTimestamp = 0
        mac = self.ble.config('mac')[1]

    def advertiser(self):  # Broadcast
        self.ble.gap_advertise(100_000, connectable=False)

    def ble_irq(self, event, data):
        if event == 5:  # A single scan result.
            addr_type, addr, adv_type, rssi, adv_data = data
            mac = bytes(addr)
            if mac in ids:
                device_id = ids[mac]
                self.writeLog(time.ticks_ms(), device_id, rssi)

    def search(self):
        self.ble.gap_scan(0, 1_000_000, 150_000)

    def writeLog(self, timestamp, device_id, rssi):
        if device_id == 0 and self.isFirst:
            return
        if device_id == 0:
            self.logBuffer += bytearray([0, timestamp & 255, (timestamp >> 8) & 255, timestamp >> 16])
        else:
            self.logBuffer += bytearray([device_id, -rssi, timestamp & 255, (timestamp >> 8) & 255, timestamp >> 16])
        if device_id == 0 or time.ticks_diff(timestamp, self.logTimestamp) > 5000 or len(self.logBuffer) > 4096:
            with open('data.bin', 'ab') as log:
                log.write(self.logBuffer)
                self.logBuffer = bytearray([])
                self.logTimestamp = timestamp
                log.close()
                self.isFirst = False
```

The BLE class is then defined, containing methods for enabling BLE, broadcasting the device's BLE visibility, searching for BLE devices, and writing the log data to a file. The advertiser method broadcasts the device's BLE visibility using the gap_advertise method, which sets the duration of the advertisement as the first parameter, 0.1 seconds as duration and the second parameter as the non-connectable Bluetooth property, while ble_irq is an event handler that is called when a BLE event occurs, specifically when a single BLE scan result is obtained. In ble_irq, the code checks if the scanned BLE device's MAC address is in the ids dictionary, and if it is, it logs the interaction between the device and the monitor using the writeLog method. The writeLog method takes in the timestamp, device ID, and RSSI (signal strength) of the BLE device and writes it to a log file. It adds the log data to a log buffer and only writes it to the file called "data.bin" when certain conditions are met, such as when the buffer is full or a certain time interval has elapsed.

```
ble = BLE()
ble.advertiser()  # Bluetooth Visibility BroadCast
ble.search()
```

In the main program, an instance of the BLE class is created, and the advertiser and search methods are called to initiate BLE broadcast and scanning.

3.3 Tests and Calibration

To test the device and calibrate the RSSI threshold for face-to-face contact detection, we reproduce all scenarios outlined in Chapter 2.1.4 with a distance of 1,2 meters from two devices each other, and we perform both static and dynamic tests for all scenarios. For static tests, we place two devices on a table by registering timestamp and Bluetooth RSSI values to assess the devices' software functionality, the case box Bluetooth directionality and signal strength at a fixed distance for proxemics. For dynamic tests, we record two people moving and interacting or not in space following a script that indicates the movements to be made over time. These tests help us evaluate the device's ability to accurately track distance and detect physical interactions between individuals in a dynamic environment. By performing static and dynamic tests, we can ensure the reliability and accuracy of the device's measurement and detection capabilities, allowing for practical use in real-world scenarios.

3.3.1 Static Scenarios

To perform static tests for checking device functionality over scenarios describing two people positioning over space, we reproduce each by placing two devices on a table and registering timestamps and Bluetooth RSSI over time in a fixed distance of 1.2 meters; Figure 3.5 shows an example of a static test for the Scenario A.

Figure 3.5: Static test for Scenario A.

Recording all RSSI values and applying statistical approaches such as average, mode, and standard deviation, we determine a first threshold that can be tuned in dynamic scenarios. In Table 3.1, we report the summary of the data analysis carried out on the registered users to verify the device's functionality and determine the RSSI threshold for a valid contact.

Scenario	Min	Max	μ	σ
A	-63	-42	-57	5
B	-67	-53	-62	2,3
C	-93	-62	-71	7
D	-91	-70	-78	5
E	-82	-57	-66	7.5
F	-89	-71	-75	2.5

Table 3.1: RSSI Bluetooth Scenarios' Summary.

The data indicates that the reflective box functions effectively detect face-to-face contact. The RRSI values show a significant variance in measured values between valid and invalid scenarios where the interlocutors face each other and those where they are not. Upon analyzing the data further, we can conclude that the reasonable threshold for a valid contact is likely -58db. However, verifying this value through dynamic testing is essential before concluding.

3.3.2 Dynamic Scenarios

In dynamic testing, we capture the movements and interactions of two people in space over space and time in a distance range of 1.2m for validation or tuning the RSSI threshold identified by static tests. Here, we create a pre-determined action paper outlining what people's movements must do over time. This stepladder starts with 30 seconds of test prep scenario C, where people are one after the other, and explains to sequentially perform a mix of valid scenarios A and B for 30 seconds, followed by 60 seconds of a mix of invalid scenarios C, D, E, F and ends with another 30 seconds of valid scenarios. After playlist people execution with recording the timestamps and Bluetooth RSSI, we focus our analysis on valid scenarios timeframes for validating or tuning the first RSSI threshold identified by static tests.

The chart in Figure 4.1 shows recording results highlighting the two-time frames of 30 seconds where valid scenarios occur, from 30 to 60 seconds and 120 to 150 seconds. Analyzing these timeframes, we see values falling below the threshold during these time frames due to the unstable Bluetooth signal and natural human movements. Upon analyzing these valid contact timeframes, we discovered many values of -57db missing the threshold by one-decibel point. Therefore, to tune the RSSI threshold in naturals interaction dynamics, we decided to reduce the threshold to -58db (represented by the red line in the figure), thus increasing the contact sensitivity of the device.

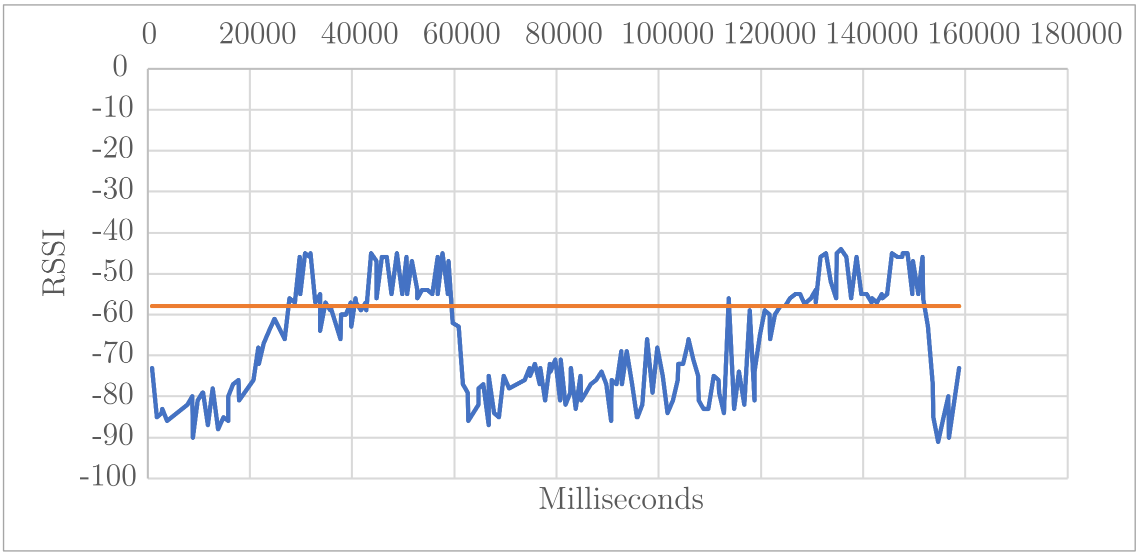

Figure 3.6: Dynamic test results.

Finally, after testing and calibrating the device, we are ready for its use in a real case study. -

Chapter 4

Case Study

This chapter demonstrates our framework's practical application and effectiveness in applying it to a case study of school class students. We provide an overview of the participant selection process and the methodology used to conduct the experimentation. After on-field data collection, achieved by building one prototype for each student, we processed data through our data pipeline process presented in Chapter 2.2.1. Additionally, during the data preprocessing phase in the data pipeline, we compare the result of our algorithm relations calculation with another paper's statistical approach, which will generate our and the paper's networks on which we will apply our process of improving social inclusion. This case study application demonstrates the real-world applicability of our framework and highlights its potential to be implemented across various scenarios.

4.1 Participants

This trial study was conducted on third-year elementary school students, which comprises 16 students, including 8 males and 8 females. In adherence to ethical guidelines, the project was reviewed and approved by the University of Bologna's ethics committee and the hosting institution obtaining parental consent. The required documentation, including privacy documentation and a document outlining the informed consent of the parents, was provided for approval.

4.2 Experimentation Procedure

This section provides a comprehensive overview of how we planned and executed each phase of the experimentation process. Our goal is to enhance the accuracy

and reliability of our results by eliminating any potential errors caused by improper use of the device. By following a structured approach, we aim to make the experimentation process as streamlined and efficient as possible while maintaining the highest precision and validity to ensure its success.
The experimentation begins with the explanation to the students of the following phases that compose it:

1. The behaviours explanation to be followed during the experimentation, prohibitions and permitted activities;
2. The extraction of the device from an urn and its wearing;
3. The data collection phase;
4. The final phase of removing devices.

The teacher sets clear guidelines for the students before beginning the experimentation. Running, jumping, and exchanging the device during the experiment is strictly forbidden. Various activities are made available to the students on the desk, such as puzzles, cards, sheets and coloured pencils, and students are left free to do any activity. The teacher allows the students to move freely around the classroom if they follow the rules. One at a time, the students are called to extract a device from an urn, and the staff helps them wear it correctly, ensuring that it is adjusted to the same height for everyone. During the experimentation, the teacher and the assistant remain at the main desk to ensure the students do not violate the rules without influencing their behaviour. Finally, the experiment concludes with each student's name being called, and their devices are removed and switched off.

4.3 Data Processing and Results

Once the data collection phase was complete, the devices stored a log file containing crucial information described in Chapter 3.1.6, such as the device's ID, timestamp, and signal strength in the flash memory. With this data at our disposal, we can now explore the practical application of the data pipeline's subsequent steps, tailored to our application context. This section delves deeper into how this data can be leveraged to extract valuable insights and drive meaningful action following the data pipeline outlined in Chapter 2.2.1.

Data integration

We used Ampy[1], a serial connection tool, to download the log files from the devices. After downloading all the files, we concatenated them into a single file containing all the information to have a single file for performing data preprocessing.

Data preprocessing

Here, we applied all the data conversion steps that led us to infer the social relationships between the subjects; The practical conversion steps are the following:

1. Binarization: Starting from raw RSSI signal strength values, we apply the threshold -58 db identified by through tests in Cap. 3.3.2, changing the RSSI values to 1 when surpassing the threshold by categorizing it as a contact; 0 otherwise. Figure 4.1 shows an example of raw RSSI values with the red line as the threshold, and Figure 4.2 shows its binarization;

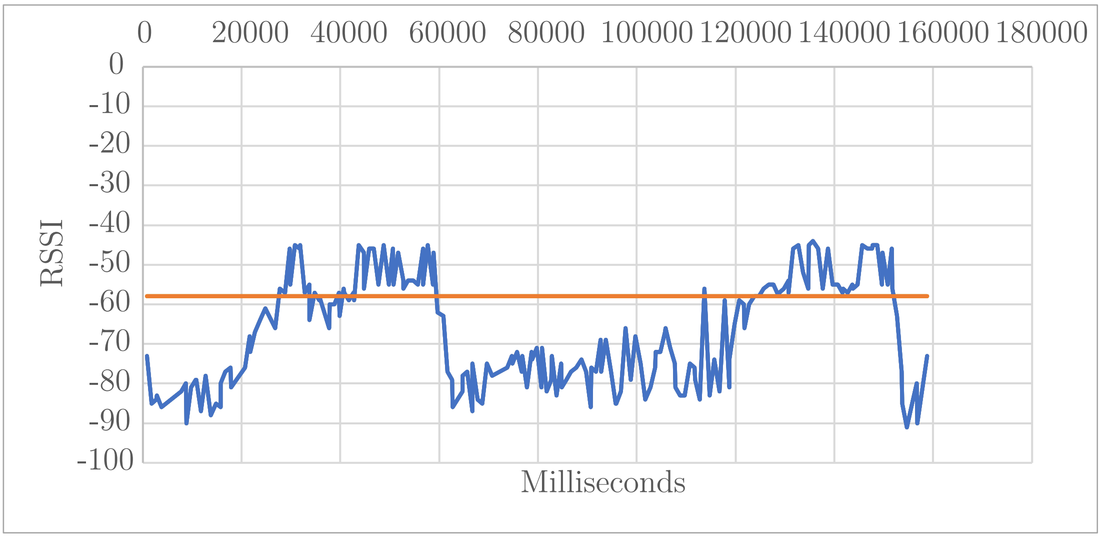

Figure 4.1: Device id 6 logs the RSSI data of device 26.

[1] https://pypi.org/project/adafruit-ampy/

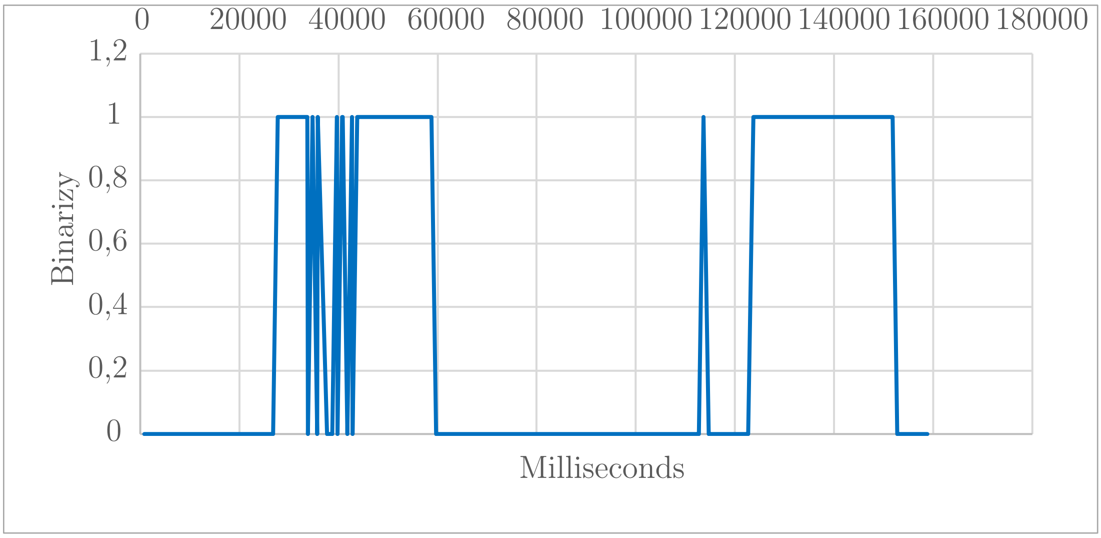

Figure 4.2: Signal Binarization.

2. Social relation computation: calculating social relations using our algorithm described in Chapter 2.2.3 compared with the referenced statistical approach described by Barrat et al. [11];

3. Network construction: converting the information into the Gephi format, thus building the network of social relationships between the subjects.

Data visualization

Here we provide visual results of preprocessing activities. Figure 4.3 shows our algorithm, and Figure shows 4.4 the results of applying the routine from Barrat et al. [11] results. The figures demonstrate an effective method for organizing nodes based on degree centrality. Nodes are grouped by degree centrality size weighted by frequency, representing them through a proportional scaling system. Using this visualization technique makes it easier to observe the relationships between different nodes and the extent of their influence within the network.
Upon comparing the outcomes derived from both algorithm approaches, it becomes apparent that there are slight differences in the visual and statistical results. The Figures show some nodes moving slightly in the centrality grouping, such as node 20 being demoted to one group. On the other hand, by examining Table 4.1, it is observable that the online algorithm generated more connections but had a lower cumulative frequency. In addition, the online algorithm resulted in a longer cumulative duration of interactions. This outcome was expected due to the implementation of the timeout system, which ensured that a given interaction could

be continued over an extended period.

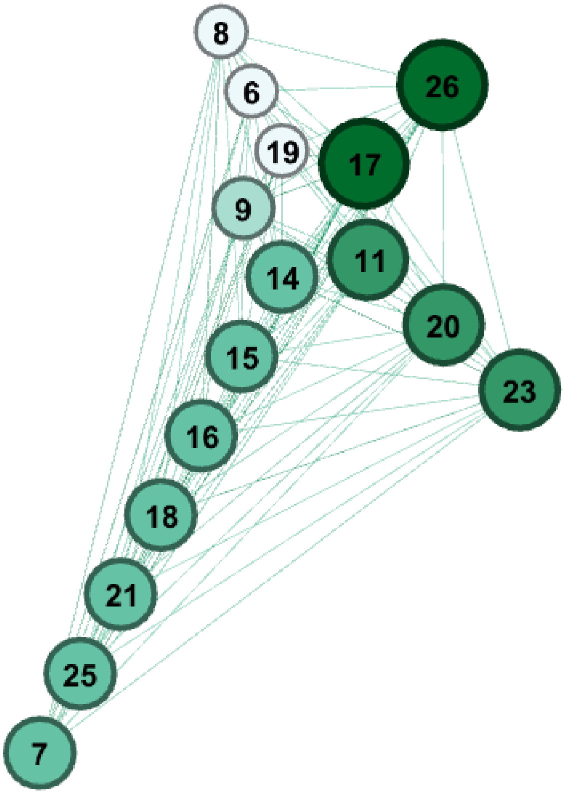

Figure 4.3: Network constructed with our algorithm.

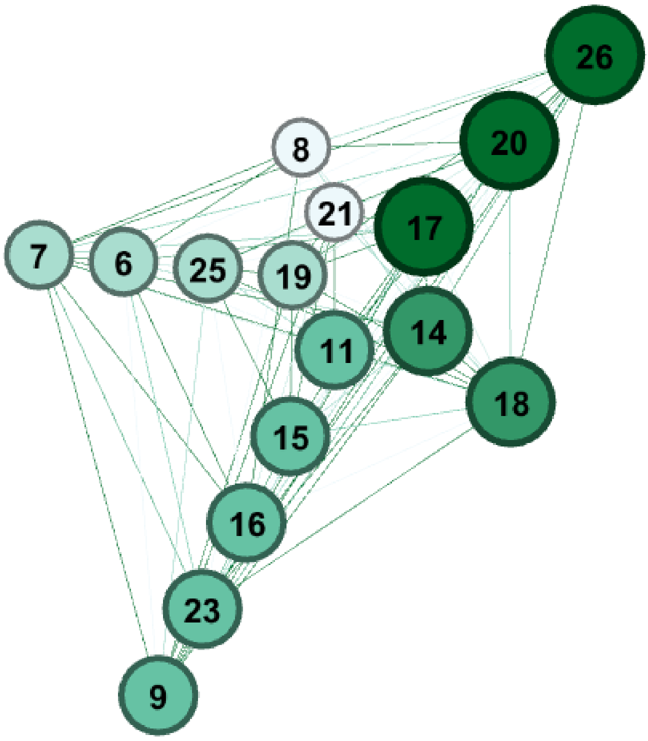

Figure 4.4: Network constructed with article algorithm.

	Our Network	Barrat et al. Network
Number of nodes	16	16
Number of edges	104	96
Number of interactions	724	889
Interactions' cumulative duration	27305	19051
Network diameter	2	2
Averange path length	1.13	1.2
Averange node's degree	13	12
Averange nodes' edge weight by frequency	6.962	9.26
Averange nodes' edge weight by duration	262.54	198.44
MIN Node's edge weight by frequency	1	1
MAX Node's edge weight by frequency	40	61
MIN Node's edge weight by duration	2	2
MAX Node's edge weight by duration	1755	1149

Table 4.1: Networks statics.

Data analysis

We start by analyzing both algorithm results; our algorithm result in the network of Figure 2 reveals 104 relationships and 724 interaction events, resulting in a cumulative duration of 27305 seconds. The length of paths in the network has an average of 1.13, with a diameter of two. Students have an average degree centrality of 13 and interact with classmates 6,96 times for a cumulative interaction time average of 256.54 seconds. Notably, we discover the lowest interaction frequency of two, while the highest is 40. The duration of interactions ranges from two to 1755 seconds. On the other hand, the results by Barrat et al. in Figure 4.4, generated through the offline algorithm, shows 96 relationships and 889 interaction events, resulting in a cumulative duration of 19051 seconds. The length of paths in this network has an average of 1.2, with a diameter of two. Students have an average degree centrality of 12 and interact with classmates 9.26 times for 198.44 seconds per session. The frequency of interactions ranges from a low of two to a high of 61, while the duration of interactions ranges from two to 1149 seconds.

Now, we apply the point-to-point procedure to improve social relations for the most isolated student on both networks described in Cap.2.1.3:

1. Calculating the assortativity coefficient by degree, we identify slightly neutral networks' nature with our of -0.088 and by Barrat et al. of -0.068 networks

results;

2. Using weighted degree centrality by duration (Table 4.2 and 4.3) and sorting in ascending order, we identify the most isolated student id 8 for both networks;

Node ID	Weighted degree centrality
8	1556
15	1862
25	2098
20	2484
6	2606

Table 4.2: First five our network weighted degree centrality order in ascending order.

Node ID	Weighted degree centrality
8	1124
15	1400
25	1403
20	1770
16	1803

Table 4.3: First five Barrat et al. result network weighted degree centrality order in ascending order.

3. Using eigenvector centrality, we identify the most popular student, ID 11 for our network and ID 17 for the network by Barrat et al., yet to be connected with ID 8 in Table 4.4 and 4.5. For choosing the most popular student, we first check the neighbour's nodes of the most isolated student for both networks. In our network, the neighbour's nodes are (14, 15, 16, 17, 18, 20, 21, 23, 26, 6, 7) the network by Barrat et al. neighbour's nodes are (14, 15, 16, 18, 20, 21, 23, 26, 6, 7). And then, we select the first most popular student not already connected to the most isolated one.

Node ID	Eigenvector centrality
17	0.28247368198897055
26	0.28247368198897055
20	0.267568361872141
23	0.267386026201629
11	0.26728183438990777

Table 4.4: First five our network eigenvector centrality in descending order.

Node ID	Eigenvector centrality
26	0.2874458437966611
17	0.2873108028973236
20	0.28593278142054396
14	0.2699442597882633
18	0.2670333191792888

Table 4.5: First five Barrat et al. result network eigenvector centrality in descending order.

4. For choosing the best path from the most isolated to the most popular student, we identify the shortest paths from id 8 to id 11 in our network and from id 8 to id 17 in the network by Barrat et al.. Using betweenness centrality and sorting in descending order, we identify the best path from the shortest paths by selecting the highest betweenness centrality for the passing node, which is for our network from id 8 to id 11, passing from id 17 and for the network by Barrat et al. from id 8 to id 17, passing from id 20 (Table 4.6 and 4.7); these passing nodes are already connected to both students.

Path	Betweeneess centrality
8, **17**, 11	0.014672919672919674
8, 26, 11	0.014672919672919674
8, 14, 11	0.011180856180856182
8, 20, 11	0.01094997594997595
8, 23, 11	0.010711880711880714
8, 21, 11	0.009362674362674364
8, 15, 11	0.009203944203944204
8, 7, 11	0.008965848965848967
8, 6, 11	0.008496873496873497
8, 18, 11	0.008160173160173161
8, 16, 11	0.007922077922077922

Table 4.6: Betweenness centrality for passing node in shortest paths on our network.

Path	Betweeneess centrality
8, **20**, 17	0.021905105476534056
8, 26, 17	0.02021988593417165
8, 18, 17	0.01889644746787604
8, 14, 17	0.015658283515426374
8, 16, 17	0.015370713942142516
8, 6, 17	0.014467120181405895
8, 15, 17	0.014010169724455441
8, 23, 17	0.010990860990860992
8, 21, 17	0.010835566549852264
8, 7, 17	0.007922077922077922

Table 4.7: Betweenness centrality for passing node in shortest paths on the network by Barrat et al.

At this point, applying the first four steps of the procedure, we have identified the most isolated student, the most popular student and the student who will act as an intermediary between the two students in establishing a first relationship approach. The next phase will evaluate the solutions' quality and the impact of network changes to connect the most isolated to the most popular student.

Data evaluation

Here we inspect how the k-core of the most isolated student changes to connect the most isolated with the most popular student to quantify the identified solution's quality. We also compare the original and modified network's assortativity coefficient to verify if the changes respect the nature of the network and quantify the solution quality and how much has changed. Due to the type of network, which reproduces the social relationships of the students in a class where all the individuals know each other, the k-core is the same for all the nodes; respectively, for our network is eleven and for the network of the paper is ten and does not change for both modified networks.
The assortativity coefficient for our original network was -0.0883 and changed to -0.0971, slightly changing in the homophilic direction. On the other hand, for the network by Barrat et al. was -0.0684 and changed to -0.0959, hugely changing in the homophilic direction, here can be evaluated another solution respecting the network's nature.

Data reporting

This section provides our report as an analysis result of the social relations network for the social inclusion improvement of the most isolated student; this report is intended for school staff and carers affected by the process, such as teachers, support teachers and psychologists.
Confidently, the most isolated student has identifier 8 because he was identified by analyzing our and the article's social relation networks. Further analysis was conducted on our and the network by Barrat et al. to identify two potential solutions to improve this student's inclusion level. Our solution identifies the student with ID 11 as the most popular not yet to be in contact with the most isolated student. To put these two students in contact, the student with ID 17, whom both students already know, is chosen as the best intermediary; therefore, we recommend asking them to put the two students in contact. On the other hand, for the network by Barrat et al., we identified the student with ID 17 as the most popular not yet to be in contact with the most isolated student. To put these two students in contact, the student with ID 20, whom both students already know, is chosen as the best intermediary; therefore, we recommend asking them to put the two students in contact.
Upon evaluating the effectiveness of the solutions presented, it became apparent

that students in a single class were almost all already in relations with each other; for this reason, the resulting social inclusion improvement can have minimal impact. However, considering the nature of social relationships in terms of affinity and the number of connections between the students, we recommend connecting the most isolated student with ID 8, with the most popular student with ID 14, via the intermediary ID 17, as the solution most aligned with the structure of the network. After careful consideration, it is advisable to implement this solution to improve social inclusion for the isolated student.

Chapter 5

Conclusions and Future Work

This thesis presents our framework for measuring and enhancing social relations, comprising theoretical and practical components.
In the theoretical part, we draw upon social theories and human behaviour to introduce methods for measuring social relations and network analysis measures for assessing the level of inclusion and solution quality. We also outline how to improve social relations by identifying key individuals, such as the most isolated and the most popular students, and using a method to connect them into a new social relationship that will modify their existing relationships and improve inclusion.
The practical section demonstrates how data can be managed throughout the social relation improvement procedure, and we showcase our social relation algorithm to construct the social relation network. We developed a device prototype for data collection to apply this framework in a real-world scenario, which we deeply described and tested. We then presented our framework in action in the school class real-world case study.
Future research could evaluate the industrialisation of the prototype to reduce cost and dimension and incorporate a camera and facial recognition technology to trace interaction quality. However, privacy concerns would need to be carefully addressed. Consider the possibility of substituting the weighted version of autovector centrality or betweenness centrality in the social inclusion improvement procedure and map students' personalities considering substituting the assortativity coefficient by degree with personality.

Bibliography

[1] Erving Goffman. *The Presentation of Self in Everyday Life.* New York: Anchor Books: A Division of Random House, 1959. ISBN: 978-0-385-094023.

[2] Edward T. Hall. *The Hidden Dimension.* Anchor Books, 1966. ISBN: 978-0-385-08476-5.

[3] Janet Sternberg. *Misbehavior in Cyber Places: The Regulation of Online Conduct in Virtual Communities on the Internet.* Rowman Littlefield, 1994. ISBN: 978-0-7618-6011-2.

[4] Social Affairs European Commission Directorate-General for Employment and Inclusion. *Resolution of the Council and of the representatives of the governments of the Member States.* Dec. 2000. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A42000X1228&qid=1669382480149.

[5] M. E. J. Newman. "Assortative Mixing in Networks". In: *Physical Review Letters* 89.20 (Oct. 2002). DOI: 10.1103/physrevlett.89.208701. URL: https://doi.org/10.1103%2Fphysrevlett.89.208701.

[6] M. E. J. Newman. "Neocortex size as a constraint on group size in primates". In: *dunbar* 89.20 (Oct. 2002). DOI: 10.1016/0047-2484(92)90081-J.. URL: https://doi.org/10.1016/0047-2484(92)90081-J.

[7] Piotr Sztompka. *Socjologia.* Znak, 2002. ISBN: 83-240-0218-9.

[8] European Commission. *European Parliament resolution on social inclusion in the new Member States.* Nov. 2004. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52005IP0244&qid=1669382480149.

[9] European Commission. *Communication from the Commission to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the Regions - Joint Report on Social Protection and*

Social Inclusion. Feb. 2006. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52006DC0062&qid=1669382480149.

[10] European Commission. *Joint Report on Social Protection and Social Inclusion*. Mar. 2007. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52007SC0329&qid=1669382480149.

[11] Alain Barrat et al. "High resolution dynamical mapping of social interactions with active RFID". In: *CoRR* abs/0811.4170 (2008). arXiv: 0811.4170. URL: http://arxiv.org/abs/0811.4170.

[12] European Commission. *Joint Report on social protection and social inclusion*. Jan. 2008. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52008SC0091&qid=1669382480149.

[13] T.S. Rappaport. *Wireless Communications: Principles And Practice, 2/E*. Pearson Education, 2010. ISBN: 9788131731864. URL: https://books.google.it/books?id=VmPT8B-5%5C_tAC.

[14] Jiawei Han and Jiawei Han. *Data Mining: Concepts and Techniques*. eng. Jan. 2011. ISBN: 9780123814791.

[15] John Blicharski. "Social inclusion and higher education." eng. In: *Widening Participation Lifelong Learning* 14.2 (Aug. 2012), pp. 79–81. ISSN: 14666529.

[16] Tim Coombs, Angela Nicholas, and Jane Pirkis. "A review of social inclusion measures". In: *Australian & New Zealand Journal of Psychiatry* 47.10 (2013). PMID: 23737598, pp. 906–919. DOI: 10.1177/0004867413491161. eprint: https://doi.org/10.1177/0004867413491161. URL: https://doi.org/10.1177/0004867413491161.

[17] D. David J. Crowley; David Mitchell. *Communication Theory Today*. Stanford University Press, 2013. ISBN: 978-0-8047-2347-3.

[18] European Commission. *Council conclusions on enhancing the social inclusion of young people not in employment, education or training*. Mar. 2014. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52014XG0201%2802%29&qid=1669382480149.

[19] Rogers Jenny. *A-Z of inclusion in early childhood / Mary Dickins*. eng. Maidenhead, 2014.

[20] William Little. *Introduction to Sociology - 2nd Canadian Edition*. eng. 2nd Canadian Edition. 2016.

[21] Elias Avramidis et al. “Using sociometric techniques to assess the social impacts of inclusion: Some methodological considerations”. In: *Educational Research Review* 20 (2017), pp. 68–80. ISSN: 1747-938X. DOI: https://doi.org/10.1016/j.edurev.2016.11.004. URL: https://www.sciencedirect.com/science/article/pii/S1747938X1630063X.

[22] Eric Wesselmann, James Wirth, and Michael Bernstein. “Expectations of Social Inclusion and Exclusion”. In: *Frontiers in Psychology* 8 (2017). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2017.00112. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2017.00112.

[23] Eric D. Wesselmann, James H. Wirth, and Michael J. Bernstein. “Expectations of social inclusion and exclusion”. eng. In: *Frontiers in Psychology* 8 (2017), pp. 112–112. ISSN: 1664-1078.

[24] Mark Newman. *Networks*. Oxford University Press, 2018. ISBN: 978–0–19–880509–0. DOI: https://doi.org/10.1093/oso/9780198805090.001.0001.

[25] Timothy J. Williamson et al. “Effects of Social Exclusion on Cardiovascular and Affective Reactivity to a Socially Evaluative Stressor”. In: *International Journal of Behavioral Medicine* 25 (2018), pp. 410–420.

[26] European Commission. *New skills / Social inclusion*. Feb. 2019. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=PI_EESC%3AEESC-2019-01610-AS&qid=1669382480149.

[27] Christoforos Mamas. “Employing Social Network Analysis to Examine the Social Participation of Students Identified as Having Special Educational Needs and Disabilities.” In: *International Journal of Disability, Development Education* 67.4 (July 2020), pp. 393–409. ISSN: 1034912X.

[28] European Commission. *PROPOSAL FOR A JOINT EMPLOYMENT REPORT FROM THE COMMISSION AND THE COUNCIL*. Nov. 2022. URL: https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A52022DC0783.

[29] Katharina-Theresa Lindner et al. “Promoting Factors of Social Inclusion of Students With Special Educational Needs: Perspectives of Parents, Teachers,

and Students". In: *Frontiers in Education* 7 (2022). ISSN: 2504-284X. DOI: 10.3389/feduc.2022.773230. URL: https://www.frontiersin.org/articles/10.3389/feduc.2022.773230.

[30] Wikipedia. *Friis transmission equation.* [Online; last edited on 24 October 2022, at 15:44 (UTC)]. 2022. URL: https://en.wikipedia.org/wiki/Friis_transmission_equation.

[31] www.psychologytoday.com. *Face-to-Face Social Contact Reduces Risk of Depression.* Psychology Today Canada, 2022.

[32] Wikipedia. *Field of view.* [Online; last edited on 1 April 2023, at 19:53 (UTC).] 2023. URL: https://en.wikipedia.org/wiki/Field_of_view.

[33] Wikipedia. *Signal strength in telecommunications.* [Online; last edited on 17 January 2023, at 01:41 (UTC).] 2023. URL: https://en.wikipedia.org/wiki/Signal_strength_in_telecommunications.

[34] Paul Sokolovsky Damien P. George and contributors. *low-level Bluetooth.* URL: https://docs.micropython.org/en/latest/library/bluetooth.html. Last updated on 20 Mar 2023.

[35] Kelley D. Haynes-Mendez et al. "Diversity, Equity, Inclusion, and Internationalization: Past, Present, and Future of STP". In: *Teaching of Psychology* 0.0 (0), p. 00986283221126424. DOI: 10.1177/00986283221126424. eprint: https://doi.org/10.1177/00986283221126424. URL: https://doi.org/10.1177/00986283221126424.

[36] Kelley D. Haynes-Mendez et al. "Effects of social exclusion on cognitive processes: Anticipated aloneness reduces intelligent thought". In: *Teaching of Psychology* 0.0 (0), p. 00986283221126424. DOI: 10.1177/00986283221126424. URL: https://doi.org/10.1177/00986283221126424.

[37] University of Massachusetts Amherst. *Applied Behavior Analysis.* URL: https://blogs.umass.edu/psych581-awoodman/module-5-measurement/. Module 5.

[38] APA Dictionary of Psychology. *social relationship.* URL: https://dictionary.apa.org/social-relationship.