AI Challenge Report: SaferMaps - An App to Help You Plot the Safest Route Home

Tom Cannon, Sophia Jones, Ferdinand Krammer, Toby Lewis-Atwell,
Benedict Rogers, Tom Ryder, Joshua Tenn

April 2022

Contents

List of Figures

List of Tables

1 Problem Brief

Recent evidence in the UK from the Office for National Statistics [1] suggests that large numbers of citizens feel unsafe when walking alone, particularly at night. Their findings suggested that busy areas and green spaces such as parks particularly led to individuals feeling unsafe, with, for example, 81% of women and 39% of men reporting feeling fairly or very unsafe in green open spaces at night.

Particularly when in unfamiliar surroundings, such as visiting an area or after relocating, it can be difficult to know the 'safest route' home. Individuals may often need time to develop the information or heuristics needed to make appropriate risk judgements; said judgments may take into account types of buildings (retail vs. residential), traffic, level of footfall and streetlight provision amongst other features [2].

Existing tools for navigation, such as Google Maps, take into account contextual data such as traffic jams when planning a route via car [3]. However, the same algorithm can sometimes route pedestrian users down dark alleyways, through unsafe areas and/or green spaces late at night. The use of AI in this area to route users can help them make appropriate risk judgements, using insights from social science, and also simplifies the routing process. These algorithms can also connect to other devices - such as personal safety alarms - to provide further functionality than the average application.

We propose an alternative application, SaferMaps, which considers further contextual information such as local crime hotspots, amenities, green spaces, canals and other features to plot a users' safest route home. We believe this application has the ability not only to increase users' safety, particularly when walking home at night, but also to increase *perceptions* of safety and provide a less stressful journey home.

2 State of the Art Review

2.1 State of 'safe' routing apps

The idea of ensuring the safety of online map users is not new, but has been approached in a variety of different ways. Given its large user base and use in almost every country, it is likely Google Maps view a crime-mapping based approach as infeasible, due to the differences in how crime data is reported and the accuracy of this data in some countries. Google has instead taken a more region-specific approach, such as allowing users to see when their taxi journey has deviated from the recommended route in India, where taxi-related crimes are more common [4].

Other apps working specifically in the UK context have taken wildly different approaches. Safe and the City [5] uses a community approach, relying mostly on user reports of incidents and notifications to social groups (families, friends) to increase safety. It provides notifications of when users enter a high-crime area but does not route the user itself.

WalkSafe [6] also shows users high-crime areas whilst not specifically routing users themselves. There are additional features, including the ability to connect to loved ones if feeling scared/threatened, and the ability to send alerts if one does not return home on time or does not 'check in' at regular intervals.

There are three key differences between SaferMaps and the applications described above:

- **Routing for the user** - The applications above both require the user to create their own path using the information provided. Our application uses this information and provides a route for a user to follow. We believe that these apps, particularly WalkSafe, simply provide too much information for users to make an useful decision about the safety of certain areas by themselves.

- **Simpler user interface without crime spots** - The existing applications show all of the local crime data on their interfaces for users to see. As described later in this report, we feel this type of

interface raises a number of valid ethical concerns. These applications make us aware of crimes we would not typically be aware of, which could potentially lead to areas being abandoned as 'crime-ridden' without good reason. This also provides the additive benefit of making the route clearer to users.

- **Use of large degree of contextual information** - Due to the large degree of openly available mapping data, we can extend our algorithm to include more than just crime data. For example, our application provides the opportunity to route as near as possible to open establishments such as pubs which may provide safety or an opportunity to call police.

We believe the various safety features that current safe map applications have are useful, particularly the ability to call for help when feeling threatened or unsafe. However, these applications require persistent use of the phone - for example, WalkSafe [6] have an option that requires consistent 'tapping', with alerts sent to an emergency contact if you do not tap in time. Particularly in inner cities (such as Bath), which are busier and with high traffic, this persistent phone use may in fact decrease safety by causing a distraction - especially if the phone is valuable and a possible target for theft.

An alternative is to, instead, take an approach similar to a personal safety alarm. These are discreet pieces of technology that either alert those nearby, or the police, and disorientate attackers, providing time to get away [7]. Due to being cheaper than an average phone, they are unlikely to draw attention yet can provide similar functionality to a smartphone. For example, a device could have lights to help route the user if they are walking an unfamiliar route, as well as alarm and SOS button functionalities. This alarm, which we call SafePal, can 'speak' to the app via Bluetooth technology. Furthermore, it still has some functionality should the phone run out of battery.

2.2 Routing Algorithms

A core element of our app is the routing algorithm, that maps a route from A to B taking safety into account. The two primary state-of-the-art routing algorithms suitable to our problem are the A* algorithm and Dijkstra's algorithm. We use a routing package, OSMnx [8], which extends the state-of-the-art package NetworkX. This implements a bidirectional Dijkstra's algorithm with binary heaps.

3 Description of SaferMaps (Ambitious Solution)

Our solution is an app, along with a separate hand-held Bluetooth device called SafePal, that provides users with walking routes conforming to safety specifications set by the user, and the ability to raise the alarm in dangerous circumstances. The app is visually similar to other popular routing apps, for a shallow learning curve for new users. There are few screens, text is minimal, and the colour scheme, font, and font size are all designed with accessibility in mind. The SafePal hand-held device is small and lightweight, and cheap, for convenience and accessibility respectively.

The app lets users specify their own safety preferences, choosing from options such as *whether to avoid green spaces* and *whether to prefer busy areas*. All choices are clearly explained in the app so users can make informed decisions. The app then takes an origin and destination, and outputs a route that conforms to the user's safety choices. Thus the app mimics the personal safety heuristics a user might employ if they were navigating an area they felt familiar with. While on the route, SafePal lets users check directions without drawing their phone, and gives the user multiple options for raising the alarm - enabling the user to send an alert in a broader range of scenarios than just an in-app button would (although this is optional).

In order to determine which route is most sensible, SaferMaps utilises a bespoke algorithm based on ground features, crime data and user reviews. In the algorithm, all possible paths (including alleyways,

pavements, footpaths etc.) are represented as edges on a network, and weighted with a safety score. This safety score is influenced by several factors: crime data in the area local to that path, factors such as whether the path runs through a green space, and user feedback. For a full review of the features included in the safety scores, please see Section 4. User feedback is delivered via the app, on a screen that allows users to select past routes they have taken and leave safety feedback (see Screen (h) in Figure 1).

The solution benefits from the combination of three primary disciplines: engineering, social science, and computer science. The use of crime data in SaferMaps requires careful considerations of social scientific concerns along with the associated ethics. The first consideration concerns which crimes affect individual safety, as well as how crime data can be used as a measure of this safety as there is no easy method to translate crime into safety. We also have to consider how crime data is incorporated in to SaferMaps - by giving the individual too much information about the safety of an area,there is the potential to unintentionally bias people's behaviour. The implications of integrating user feedback into the safety score has potential issues. For instance, research has shown that people suffer form 'perception bias' when making decisions at to there safety [9]. Both of these factors generate important ethical questions which we have considered in the design.

One of the ways SaferMaps distinguishes itself is with the integration of a small device that makes alerting authorities possible and safe in a broad variety of dangerous situations. For example, a user can deliberately place the device out of Bluetooth range of their phone to alert the authorities, allowing an alert to be sent in an incredibly discreet manner. Creative solutions to the problem of discreetly raising the alarm could only be solved with engineering experience and knowledge.

The core of our solution is a routing algorithm which required the use of graph theory, various types of machine learning (unsupervised learning, gradient descent, and supervised learning), simulation, and other computer science and mathematical fields. A broad knowledge of these fields was necessary for a functional and useful solution to our problem specification.

The solution is much stronger than the sum of the contributions from each discipline - SafePal is well integrated in the design of the algorithm for providing safety feedback, and the crime and user feedback data the app relies on could not have been used without social scientific interpretations of the data.

3.1 Green Space Avoidance

Despite the numerous physical, mental and environmental benefits associated with green spaces, they are often associated with lower levels of safety, both objective and subjective (i.e. perceived). Evidence for green spaces being statistically more unsafe is conflicted and, as expected, highly dependent on a number of factors, including the location of the space (e.g. in a rural or urban setting), and the time of year and day [10]. With regards to SaferMaps, if crimes are actually occurring in green spaces, this will be picked up by the crime model. Thus this component is focused on providing routes that do not pass through green spaces, because of people's *perception* of feeling unsafe walking through green areas. This perception is very real; a report based off the 2018-19 National Survey for Wales found that over 60% of those surveyed felt "very unsafe" when walking alone in their local area after dark [11]. If individuals feel unsafe by the route presented to them because it goes through a green space, they simply will not use the app. More than this, if someone feels unsafe walking through a green space, it may limit their day-to-day lives and eventually have a negative effect on their health in other ways [10]. For these reasons, having a preference to avoid green spaces in the app was implemented.

3.2 Crime Avoidance

Safety is often associated with crime, with the appearance of history of crime reducing feelings of safety and increasing fear. However, work has shown that individuals make decisions on their safety based on

their perception of the place and not what crimes have taken place in the area; these perceptions are often due to the physical state of 'decay' in an area and might be associated with higher levels of crime; they may also be associated with variety of socioeconomic factors amongst other features [9]. As such, while SaferMaps will incorporate individuals' perception of safety, the safety of different locations will primarily be based upon police crime data, which will be categorised based upon effect on the individual. To determine the safety of an area, different crime clustering models will be used for the different types of crimes which occur, with the resulting clustering being weighted to approximate the region's safety. So as to make these models more accurate, information is not only needed about the type of crime and how long ago it occurred, but also the time of day it occurred. As data will only be relevant for a limited length of time there is a need to be able to update the safety as time passes and the nature of crime in different areas changes. This would be best done by using data directly from the relevant police services.

It has been mentioned previously that people suffer form 'perception bias' preventing them from making accurate predictions of their safety. For people to use the app, they need to feel safe and as such, there needs to be a mechanism to take into account perceived safety: this will be done through the use of user feedback. The feedback will be used to inform the safety score of different areas. This quantification allows for the routing that takes into account both perceived and 'real' (as approximated by crime data) safety.

3.3 User reviews

There is only so much that ground and crime heuristics can suggest about the perceived safety of an area. Ultimately, the users of the app are in the best position to decide what *feels* safe and what doesn't. The ambitious solution gathers user feedback of perceived safety on routes and updates the global routing algorithm for future user routes. Ideally, areas where there are no reviews should have them *predicted* based on the ground and crime heuristics.

3.4 Role of SafePal

SafePal is a Bluetooth-enabled, low-cost addendum for the app. It is designed to combine the roles of a personal throw away alarm, low attention navigation aid and backup help contact point. The solution provides function to the app that phone software alone could not. It has the ability to call the police and run an alarm to call attention to the user, while being far more discreet and disposable than the users phone. It also makes the user able to navigate without holding their phone openly and provides methods to call for help in the case of the phone being stolen, as well as providing a disincentive for this occurring via a Bluetooth disconnect-activated tracking of the phone location. Should the alarm or police call feature be used, this will be reported through the App and serve to adjust safety rating maps.

4 Prototype description

The SaferMaps prototype design consists of the following components: (1) the app design user interface (UI), (2) the general routing approach, (3) green space and other area detection methods, (4) crime data cleaning and modelling, (5) dynamic review model updates, (6) prior review predictions, and (7) the SafePal physical accessory. Each of these components are described in more detail in this section.

Code is available in a GitHub repository (https://github.com/toboooo/cohort1-AI-Challenge) and data is stored in a shared folder (https://computingservices-my.sharepoint.com/:f:/g/personal/fgpk20_bath_ac_uk/EigrMBownTVCrA2MTQz6U40BZrMclCE5IJrEHdt1FQaEnw?e=UciTfH).

4.1 App Design

Figure 1 shows our vision for the SaferMaps app. Users create an account with the app so that their settings preferences and emergency contacts are saved. We anticipate that creating an account will create perceived user accountability, reducing the likelihood of users providing deliberately false or manipulative feedback. On the creation of a new account, users are asked whether they consent to their location data being sent to emergency services in the event of an alert being raised, as shown in figure 1(c).

Users must also adjust and confirm their settings when they create a new account. The settings, and their descriptions, can be seen in figure 1(d). The app has a range of safety related features, but the choice to enable them is left with the user. Some of the features, such as avoiding green spaces, could in some areas conflict with the safety scores derived from crime data. By giving the user choice, the app allows users to follow their safety intuitions and choose routes using the heuristics they would personally employ to increase safety on a route they are familiar with.

Once logged in and set up, the app has a side menu bar for navigation, and the following pages:

- Home
- Rate a Route
- Report a Problem
- Settings
- App Info

From the Home screen (figure 1(f)) users can set a start and end point and find a route. This screen will deliver directions and display the route, similar to other commercial routing apps. Users can also report live feedback, using a button that can be pressed if they pass a section of their route that felt unsafe. This will not raise an alarm - its only affect is updating safety scores in the app. Located in an easy-to-reach position on the screen is an Alert button: if pressed for 3 seconds, the app will send an alert to local emergency services. To avoid accidental calls, the screen will flash red when the button is pressed (see figure 1(g)) to make the user aware they have pressed the button, and must be held for a full 3 seconds. This is similar to the emergency call function on iPhones.

The Rate a Route function (figure 1(h)) provides one of the ways users can give feedback. On this screen, users select a route they have taken from a list of routes ordered by recency. Users may report how safe they perceived the route to be, and a text box allows for additional feedback. We envisage the unstructured additional feedback to be used to make long or short term changes to the safety scores of areas of the map by the discretion of a support team.

The Report a Problem screen is for users to report issues while using the app, such as bugs.

Finally, the app includes an Info screen (figure 1(i)) to inform users of the app's purposes and aims. The Info section reminds users that SaferMaps cannot guarantee the safety of users.

The app uses an accessible colour pallet: using black, white, red (#e31c3d), and blue (#0071bc) [12]. The minimum font size is 16, as recommended by [13], and the font is Arial, a common sans-serif font as recommended by [14].

The app includes 3 opportunities for users to provide feedback, all of which are incorporated into the safety ratings of areas of the map:

- Live feedback, given while a user is on a route. Feedback is provided via the "Live Feedback" button on the home screen, and relates to the specific area of the route the user is currently located on.
- Rate a route, given after a user has finished a route via the "Rate a Route" screen. The feedback relates to the entire route, unless more details are specified in the "Additional Feedback" text box.

Figure 1: SaferMaps app mock-up

- Pressing the alarm or police call button: if a user chooses to alert emergency services or trigger the personal alarm, the app will register the action as unsafe feedback for the location the alarm button was pressed.

The app will link to SafePal using the phone's Bluetooth connection. This will be a two-way serial connection, with the phone location data being used to calculate and transmit direction data to the LED interface on SafePal to enable discrete, low distraction navigation. SafePal will communicate to the app commands given by the user to contact the police, using the phone's inbuilt systems to this end. The two will also maintain communication if able, and will detect the removal of the users phone and alert both the phone and SafePal if this should occur. The phone will subsequently require user action to avoid identifying the theft of phone by this separation and beginning the phone giving law enforcement its location.

4.2 Routing Approach

The route planning application must take into consideration the safety of citizens, and the speed and distance of the route. The first thing to establish was which mapping and route planning software would we develop our system with. We decided on a Python package called OSMnx which is described in more detail in the following subsection. Next, we find and weight heuristics that could be considered either unsafe, or contributing to a person feeling unsafe. At this point (without a detailed survey) we focused on the following features: green spaces, canal paths, proximity to amenities, previous history of crimes, and the reviews of users (simulated in this prototype). Each of these features is then weighted according to a prior scale of safety and user input preferences; this is explored in the weighting sub-section.

4.2.1 OSMnx

OSMnx is a Python library [8] through which geospatial data from OpenStreetMap [15], an open-source mapping database, can be directly loaded into the Python workspace. The OpenStreetMap data for Bath is obtained as simply as calling `OSMnx.graph_from_place('Bath, UK')`, from which a 'graph' is returned. The graph is essentially a network of nodes, points of interest defined by a latitude and longitude coordinate, and edges, essentially roads, paths etc. that connect nodes together. For our work, only the edges were of interest, as these are the relevant component for routing. The edges in the Bath graph are visualised in Figure 2. Each edge has a set of information and, as will be seen in subsequent sections, the important ones for SaferMaps are the geometry (a set of coordinate pairs defining the edge) and max speed (i.e. for roads).

4.2.2 Weighting

The OSMnx in-built routing algorithm relies on a "max speed" feature of each edge. In short, the routing algorithm finds the fastest route based on the distance travelled and this max speed. This presents an opportunity for us to recycle this algorithm by weighting the "max speed" of each edge by our safety scores, therefore biasing the routing algorithm to choose safe *and* short routes.

In this prototype design, we have enabled the users to pick their preferences of what areas they would find safe. For instance, they have a choice to avoid green spaces and violent crime areas, prefer to walk near open amenities but aren't bothered by canals or drug related crimes. Each of these choices will influence the edges "max speed" features. The equation which was used for this prototype is below and an example of the final relative edge speeds of a certain configuration is shown in figure 3.

It should be noted, that for simplicity the users do not have full access to adjust every variable in the weighting equation. In this prototype, users will have the option to choose between 0 or 1 for green space

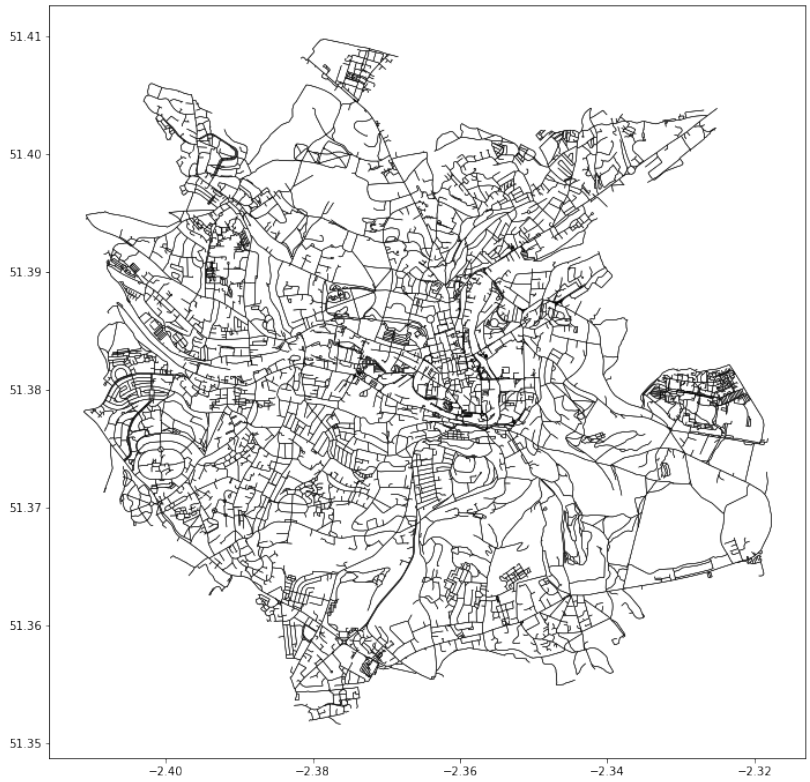


Figure 2: Edges of Bath from the OSMnx library.

importance, canal importance, and amenity importance (Also note, that the parameters are not set as binary in some of the demonstrations of the prototype, for instance in 3 they are set as 0.5). Further, users will be able to adjust β by adjusting the sliding safety co-efficient scale which scales between 0 and 1 at 0.1 intervals. Therefore the fixed parameters are currently: H, M, L and α and are set as 1, 0.75, 0.5 and 0.1 respectively. In future work, these parameters can be optimized for maximum safety and speed.

$$\mu_u = \mu_0 * \frac{\beta * G * C * A * (R * (1 - \alpha))}{(H + M + L)^{0.5}} \tag{1}$$

where:

$$\mu_u = \text{updated edge speed}$$

$$\mu_0 = \text{original edge speed}$$

$$R = \text{daily edge average review score}$$

$$\beta = \text{user defined safety co-efficient}$$

$$\alpha = \begin{cases} 0 & \text{if R == 1} \\ \text{review update learning rate} & \text{otherwise} \end{cases}$$

$$G = \begin{cases} 1 - \text{green importance} & \text{if edge in green space} \\ 1 & \text{otherwise} \end{cases}$$

$$C = \begin{cases} 1 - \text{canal importance} & \text{if edge on canal} \\ 1 & \text{otherwise} \end{cases}$$

$$A = \begin{cases} 1 + \text{amenity importance} & \text{if edge near amenity} \\ 1 & \text{otherwise} \end{cases}$$

$$H = \text{high crime importance} * KDE_{high}(long, lat)$$

$$M = \text{med crime importance} * KDE_{med}(long, lat)$$

$$L = \text{low crime importance} * KDE_{low}(long, lat)$$

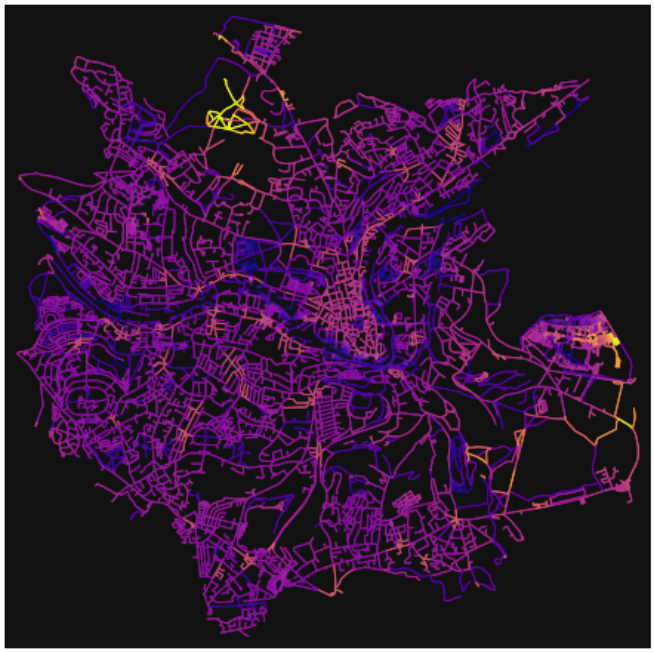


Figure 3: An example of the relative edge speeds (lighter is faster) of a map after the weighting equation with the following features had been applied: high crime importance = 1, med crime importance = 0.75, low crime importance = 0.5, green importance = 0.5, canal importance = 0.5, amenity importance = 0.5, review scores = 1 (do nothing), review learning rate = 0.1, safety co-efficient = 0.4

4.3 Green Space Avoidance

4.3.1 Green Space Data

In OpenStreetMap, locations have a set of descriptive features, from a wide range of possible features, found in [16]. The OSMnx function `osmnx.geometries.geometries_from_place()` allows specification of a set of tags, from which corresponding locations are returned. There are a wide variety of types of green space defined in OpenStreetMap, as reflected in Table 1. Each returned green space is defined as a polygon with a latitude and longitude for each node defining that polygon.

4.3.2 Evaluation of Green Space Data

The returned green spaces were evaluated manually, as shown in Figure 4. Green spaces are well defined in the urban parts of Bath compared to the more rural parts, specifically farmland areas, for example in the top left and bottom right of Figure 4. Note that this is despite trying the 'landuse: farmland' tag. By manually querying the features of this area on OpenStreetMap, there simply is not any tag to declare it as farmland. Because the perception of being unsafe in green spaces is predominantly in urban areas [10], and because it appears that the farms do not have footpaths passing through them, this was not deemed an important issue.

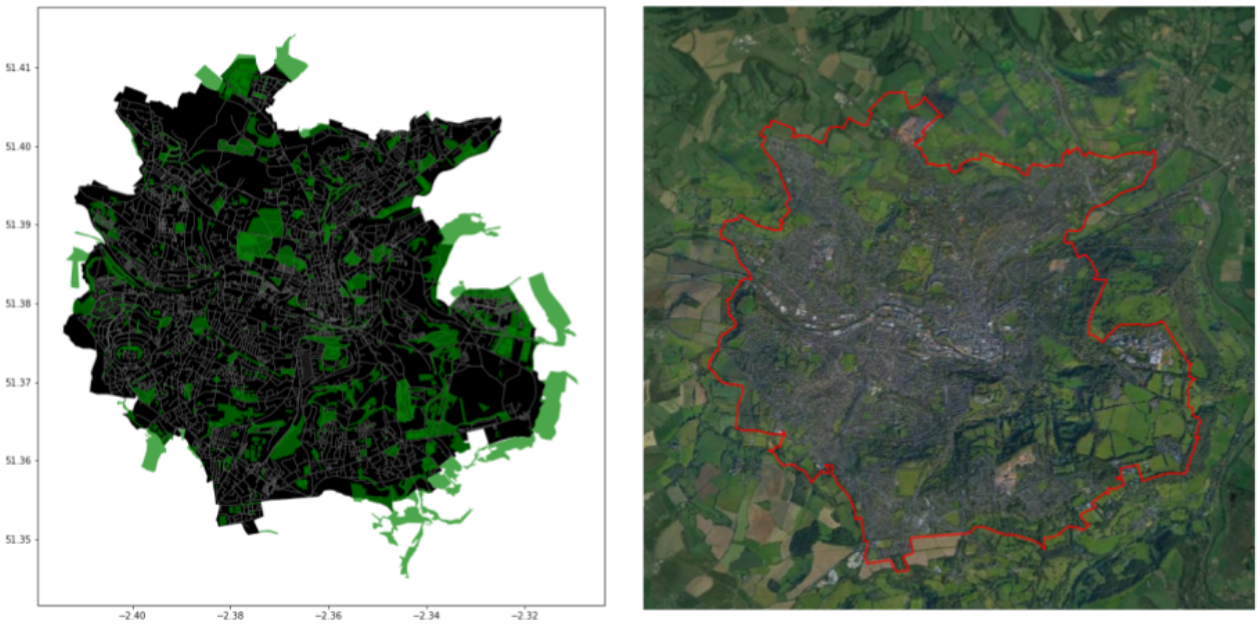


Figure 4: Left: The Bath area with edges (grey lines) and green spaces overlaid. Right: Satellite image of Bath from Google Maps, with area outline overlaid.

4.3.3 Green Space Penalty

The aim here is to prevent the route passing the user through a green space (if they choose). To implement this with the proposed route weighting approach, the general idea was to check if, for each edge, any of the constituent coordinates reside within the polygon of coordinates defining a green space. A single

Table 1: OpenStreetMap map features for green spaces.

Tag	Value
Leisure	park, pitch, playground, garden, dog park
Land use	grass, forest, recreation ground, allotments
Natural	wood, grassland, heath

check is made straightforward using the `polygon.contains(Point)` function from the Shapely library [17], and this is repeated for each point in an edge. However, this function is computationally expensive, so a brute force solution (check every edge in every green space) is highly time-consuming. Instead, the centre coordinate of each green space is found, and the distance between that centre and the first and last coordinate pair of each edge is computed (with the nearer one returned). Then, only edges whose 'distance' is less than 0.01 (see Figure 5 for scale) are checked for going through the green space. With this approach, all the edges found to have a coordinate inside a green space are shown in Figure 5. For these edges, they are essentially tagged as going through a green space, that is then used in the weighting equation (Equation 1). The effect of a user setting green importance to 1 (i.e. avoid completely) on the provided route is shown in Figure 6.

4.3.4 Extensions: canal proximity avoidance and amenity detection

The same method for detecting green spaces was extended to include edges which passed within (approximately) 10 meters of local amenities (pubs, bars, cafes, etc.) and to detect edges which ran by canals. These two methods had their own weightings which were user customisable and impacted the edge speeds which then impacted the routing algorithm. Figure 7 demonstrates the edges which are identified by these two heuristics.

4.4 Crime Model

4.4.1 Effect of crimes of the safety of the individual

The data set containing the crimes and their locations provides a list of 14 different types of crimes, with each category covering a range of different offences. These were then ranked in order of the effect on an individual's safety so that different weighting could be applied to the routing algorithm. To make the weighting simpler, it was further broken down into three categories of high, medium, and low effect, as indicated in Table 2.

To split the crime types by severity, one approach we considered was to use the table of offences as produced by the Crown Prosecution Service (CPS)[18]. However, while this gives a good idea as to the severity of different crimes it does not exactly match the categories provided in the local police data. Furthermore, many of the crimes listed as high severity have low to negligible effects on individual safety (e.g., online fraud). In order to rank the crime types on the severity of their effect on the safety of the individual, we needed to understand the nature of each type of crime. The below discussion of the severity of crime was informed by the categories as per police descriptions [19].

While Violence and Sexual Offences contains crimes which may not affect general members of the public's safety (such as domestic violence), it does cover crimes such as grievous bodily harm and common

Table 2: The effects of different crimes on the individual

High Effect	Medium Effect	Low Effect
Violence and sexual offences	Criminal damage and arson	Burglary
Theft from the person	Drugs	Vehicle crime
Possession of weapons	Shoplifting	Bicycle theft
Public order	Other crime	Other theft
Anti-social behaviour		
Robbery		

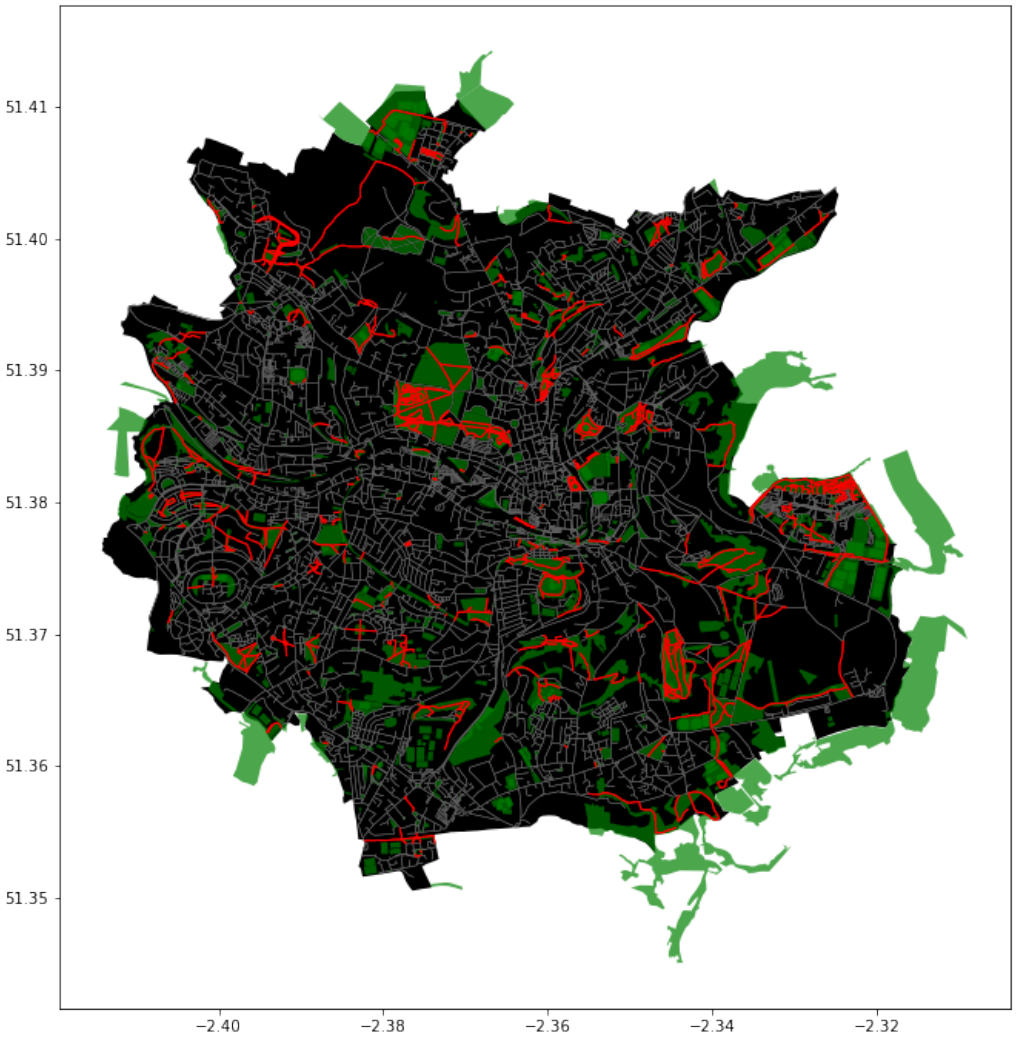
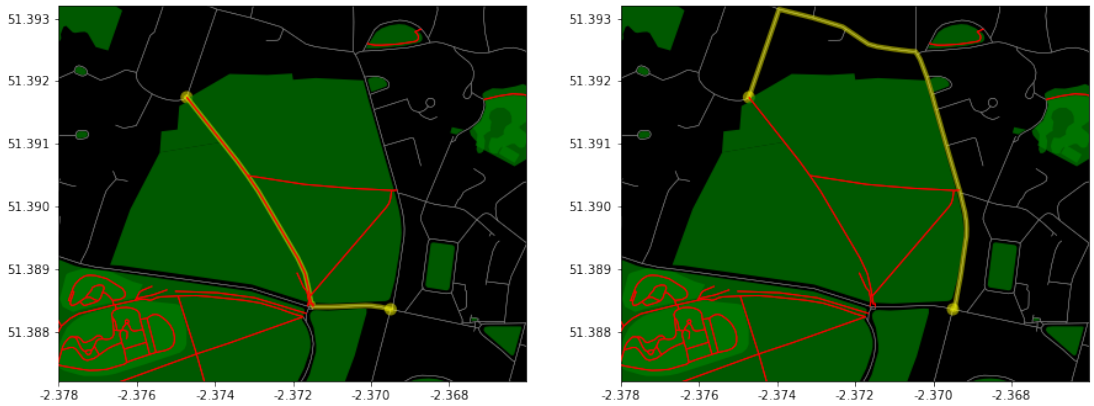
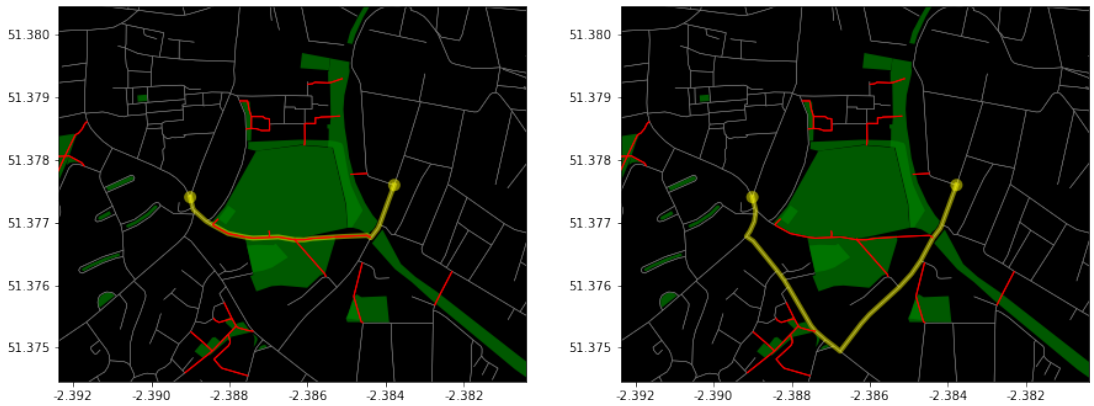


Figure 5: Edges in red are found to be at least partly in a green space. Note that edges that may look like they aren't detected, such as around (-2.38, 51.40), are between individual green spaces.

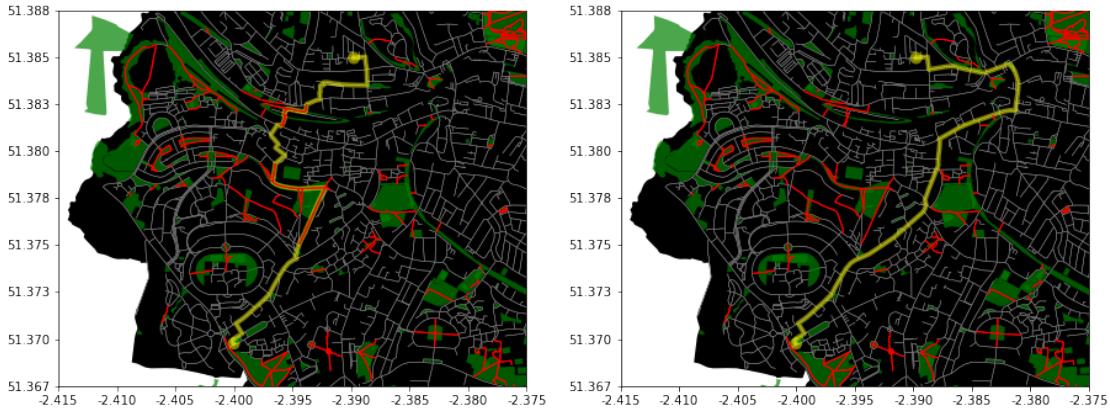
assault. Theft from the person directly involves the individual however they are generally crimes which don't involve the use or threat of physical forces. While possession of weapons may not directly affect a member of the public if weapons are not used, it was decided that individual safety would be greatly affected, due to the chance of more serious crimes occurring. Public order offences are offences which cause fear, alarm or distress which would in turn make the individual feel less safe. Similarly, anti-social behaviour involves others making a nuisance of themselves in an anti-social manner making people feel unsafe. Finally as robbery involves the use of force or threat to steal this affects the safety of the individual



(a) Victoria Park.



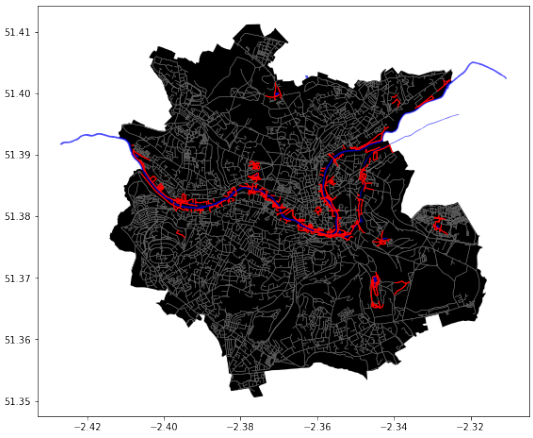
(b) Brickfields Park.



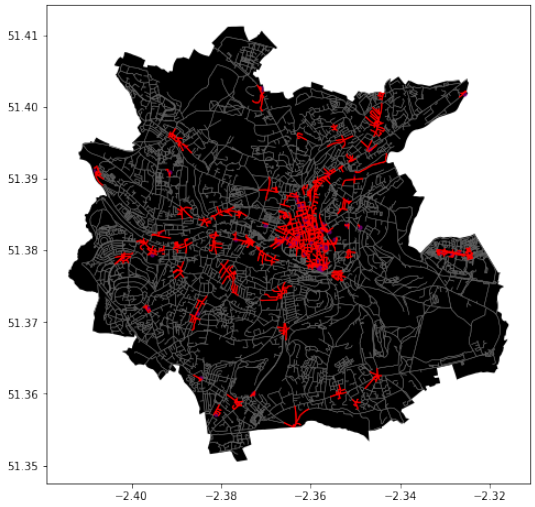
(c) Longer route through Bath.

Figure 6: Different routes through Bath. The yellow line shows the route. For each route, the left image shows the fastest route with no penalty applied to green spaces, and the right shows the route after the green space penalty is applied.

greatly as there is an increased risk to the individual. For these reasons we ranked having a high effect on safety.



(a) Red routes demonstrate the edges which are close to bodies of water.



(b) Red routes demonstrate the edges which pass within approx. 10m of local amenities.

Figure 7: Figures of edges near a canal path and those near local amenities

While criminal damage and arson may not involve direct effects on the individuals safety it is occurring around the individual they are likely to feel less safe. Likewise, while drug crimes may not directly affect the individuals' safety, associated offences will. Shoplifting occurs in many forms, most of which have no effect on whether or not the individual feels safe, there are occasional instances where they may not feel safe. Similarly, while the category of other crimes (a variety of offences) may be predominantly comprised of crimes which have no effect on the safety of an individual, it may include some crimes that implicate individuals' safety. It was thus decided that these four crimes would be included in the crimes which have a medium effect on someone's safety.

While they may be serious offences, it was decided that other theft, burglary, vehicle crime and bicycle theft have a low effect on a person's safety and so were combined in the lowest classification rating. Other theft was deemed to have a low effect on safety as it involves thefts such as fraud and blackmail. Burglary was deemed to have a low effect on safety on the streets, as it involves entering a building with the intention of stealing. Vehicle crime was also deemed to have a low effect on safety as it predominantly includes theft from or of a vehicle. The final type of crime is bicycle theft which in most cases is not likely to involve any threat to individual safety. Nonetheless there is the potential that if someone was walking past while a bicycle theft was in process and the person tried to intervene then there would be a threat the persons safety.

4.4.2 Crime Modeling with Unsupervised Learning

In order to be able to evaluate a potential route for its safety (as determined by the relative amount of crime associated with a particular area), a model of the crime distribution in Bath was required. This is a problem that seems ideally suited to unsupervised machine learning, and therefore a few different unsupervised models were tested and visualised to determine their suitability for this problem. All models were implemented using the scikit-learn Python library [20].

The crime data collected between March 2019 and January 2022 from the Avon and Somerset Constabulary was obtained from the UK police website [21]. This dataset contained crime data from the entirety of the Avon and Somerset counties, and so the data specific to Bath were selected by removing crime data with longitude and latitude outside of the city. In total, the final dataset contained 24714

crimes, each with longitude and latitude coordinates (which were used as the features for the unsupervised machine learning algorithms) and the crime type (which was used to separate the data into three categories according to the severity of the crime, as discussed in Section 4.4.1).

A Density-based spatial clustering of applications with noise (DBSCAN) [22] and agglomerative clustering [23] models were trained and visualised as seen in Appendix Section A.1. However, the crime data in Bath does not fall into many distinct clusters, and additionally these methods do not readily provide a means of evaluating the crime density at a given location. Therefore, it was determined that a method that provided an estimation of the crime distribution was required.

Therefore, a Gaussian mixture model (GMM) [24] was trained on the full crime dataset and visualised. Several different numbers of components were tested, and examples of a low number of components and a high number of components can be seen in Figures 21 and 22 in the Appendix Secton A.2. However, it is clear that the crime data for Bath is not Gaussian-distributed, which violates the fundamental assumption of the GMM, and it was thus deemed unsuitable for this task.

The final model tested was kernel density estimation (KDE) [25, 26]. For some set of samples, drawn from an unknown probability distribution, this model attempts to approximate the probability density function for the data through the equation:

$$f(x) = \sum_{i=1}^{N} K(x - x_i; h)$$

Where N is the number of sample points, x is the point at which a density estimation is to be made, h is the bandwidth hyperparameter that controls the smoothness of the approximated density function and K is the kernel function, where, for example, the exponential kernel is given by:

$$K(y - x; h) = exp\left(\frac{y - x}{h}\right)$$

While it may be stated that the KDE model is very simple, note that it boasts the advantage that it is very transparent and explainable, and the density modelling provided by KDE can be easily understood through simple visualisation.

KDE models were trained and visualised with serveral different kernel functions, namely, the Gaussian, Tophat, Epanechnikov, exponential, linear and cosine kernels. These visualisations show the log-likelihoods of the models over the longitude and latitude coordinates of Bath and these may be found in the Appendix Section A.3. From these visualisations, the exponential kernel was determined to provide the best representation of the crime distribution in Bath. Following the selection of the exponential kernel, the bandwidth hyperparameter, h, was fine-tuned, and the plots visualising this fine-tuning may be found in the Appendix Section A.4. Given the nature of this problem, it was not possible to quantify the quality of the KDE model fit. However, for the final value of the bandwidth (which was chosen to be 0.001), visualised in Figure 8), a lower density path that corresponded to the Avon river was visible, and therefore this was determined to be the indicator that the model was adequately fit (since there are very few crimes occurring within the river).

After finding an appropriate value for the bandwidth parameter, the crime dataset was split into three subsets, each containing the data for crimes corresponding to one of the three levels of severity (high, medium and low). A KDE model was trained for each of the three subsets of crime data, so that their separate crime density estimations could contribute differently to the weights of the routing algorithm (that is, the high severity model would provide stronger contributions to the weights than the lower severity model). Visualisations of these three models may be found in the Appendix Section A.5. As a sanity check that the model was able to affect the routing algorithm such that it avoided high-crime

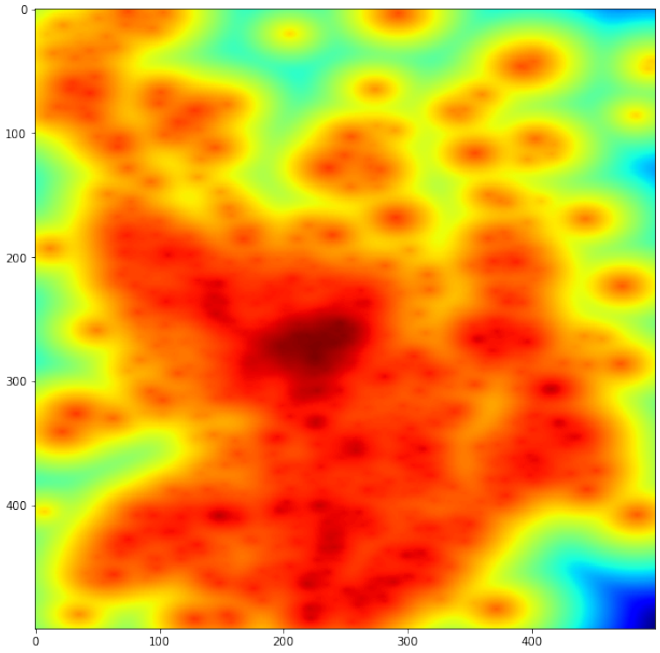


Figure 8: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 0.001, trained on the entire crime dataset. Note the path of lower crime density corresponding to the Avon river that may be seen approximately along the line x = 340.

areas, an example route was visualised before and after re-weighting the routes according to the density estimation, as seen in Figure 9.

4.5 Reviews and model updates

All the analysis so far has provided a static transformation based on ground and previous crime analysis. The results resemble a similar relative edge-speed map as Figure 3. In this project this static view of edge speeds has been called the prior map. However, another source of safety information is user reviews. In equation (1) the variable "R" is corresponding to a value which is an average of user ratings of an edge for a particular time step (in this case a 24 hr period). The speed of the edge is then updated as per equation (1). This dynamically updated map is called the posterior map. The user reviews are treated as a sampling method.

An example is demonstrated below. In figure 10 the yellow box demonstrates an area which is simulated to have bad user reviews, over several time-steps. Figure 11 demonstrates how this alone can change the weighting of the edge speeds and figure 12 demonstrates how the route changes to adapt to these poor reviews. This is of course a trivial and artificial example, however it perfectly demonstrates the capability of the updating equation to update the route based on a dynamic sampling technique - in this case user reviews.

4.6 Review predictions

Most areas have substantial ground data (from OSMaps) and crime data (from UK police) however a key limit of scaling this method is the availability of review data. For instance, would the app provide a successful routing suggestions if there were no user reviews? In this section it is demonstrated that user reviews can be *predicted* from known area data. This would provide a substantial benefit in areas which

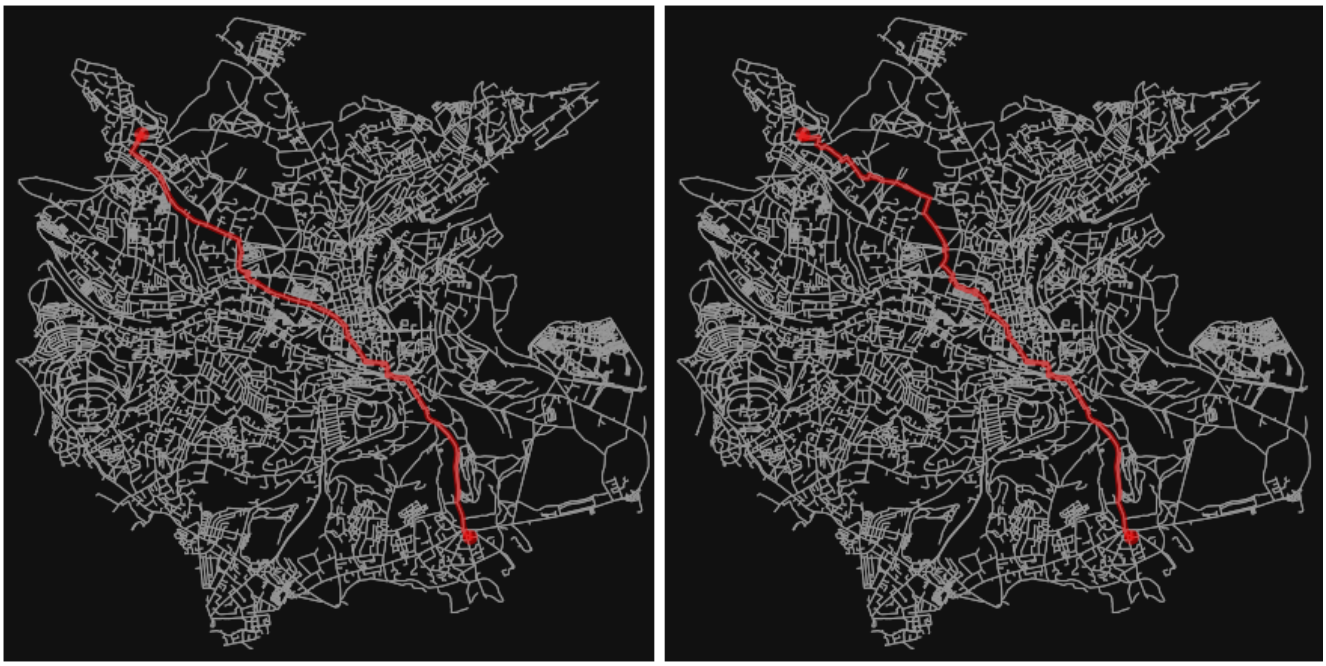


Figure 9: Changes in the routing algorithm when using the crime KDE model. Left shows the default route between a start and destination before re-weighting using the model. Right shows the route between the same start and destination after re-weighting the edges using the KDE model. Note the substantial change to the path in the north-west corner of the map to avoid the crime in that area.

haven't been reviewed or during the early stages of the app use.

The edge data is stored in a GeoPandas DataFrame. In this project we have supplemented the standard GeoPandas DataFrame by adding in which edges pass through green spaces, along canal paths, near amenities and average user reviews. However, this is just a scratch on the surface of the possible features which could be added to each edge row. The OSMaps database has 1000's of ground features which can all be added into the DataFrame. This takes considerable amount of time and wasn't practical for this project. However, one extra feature was also added to the DataFrame; if an edge was part of an alley. The alley feature wasn't weighted in the edge speed updates, but instead it was used as an indicator to adjust user reviews and demonstrate that it is possible to predict the user reviews.

In this demonstration, it is assumed that only reviews to the west of the circus are known. Every edge with the positive alley feature in the west was then reviewed poorly (a sensible assumption). Figure 13 demonstrates a edge speed map where the only weights are the reviews (this also accurately describes alleys).

The formation of the problem now is that we have half data to train on and half data to predict on. This is in the form of a very classical tabular machine learning regression problem. X-train is all of the edges west of the circus and all features except the review score column. Likewise Y-train is all of the edges west of the circus but only the review score column. Each of the features were first transformed using a custom one-hot-encoder. Next an XGBoost regressor model was trained; n_estimators=100, max_depth=4, eta=0.1, subsample=0.7, colsample_bytree=0.8, tree_method="gpu_hist", enable_categorical=True (although, this problem is trivial so any parameters should work)

Once the model is trained, it is used to predict the reviews of the entire data-set (west and east of Bath). These reviews are then substituted into the GeoPandas DataFrame column for reviews and plotted in figure 14.

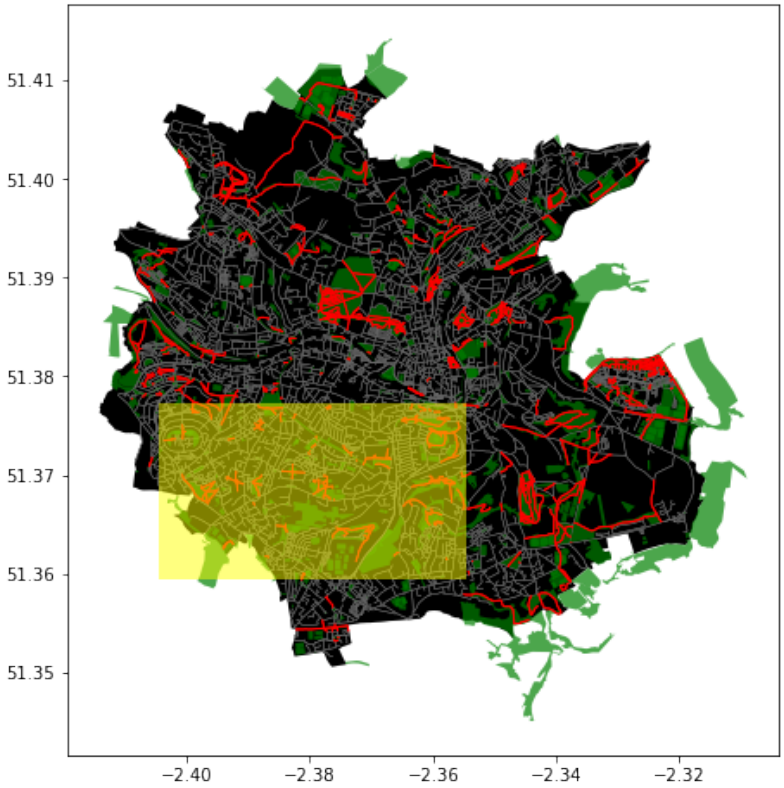


Figure 10: The yellow box in this figure is the "bad review area" where in this simulation each edge is highly likely (50%) to be given a bad review on any given day.

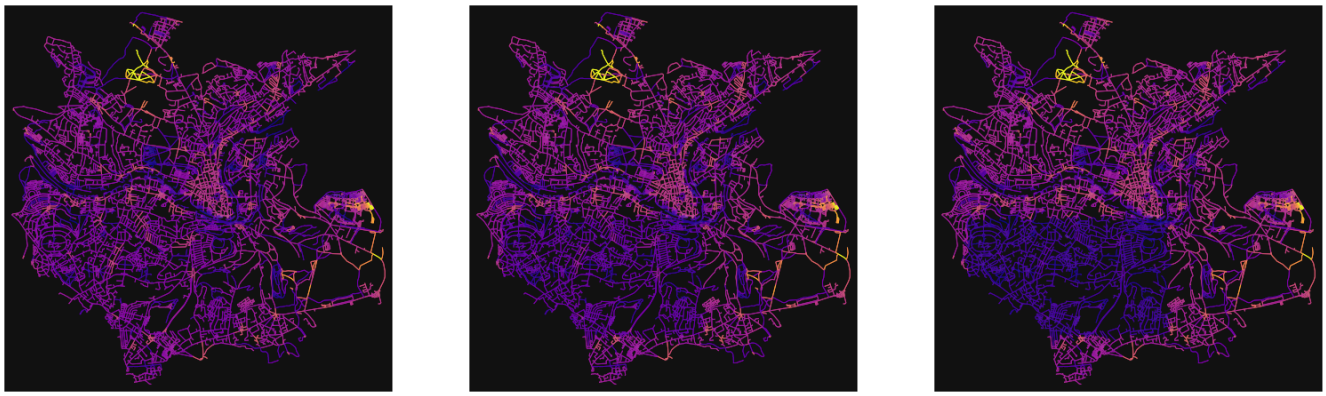


Figure 11: images representing the edge speeds as they change during each review update step. Left is after 0 steps, middle is after 3 steps and right is after 8 steps, it's clear to see that the edge speed within the area of poor reviews quickly becomes an unfavourable.

Figure 12: images representing how the route changes at each review update step. Left is after 0 steps, middle is after 3 steps and right is after 8 steps, it's clear to see that the route adapts to avoid the poor review area.

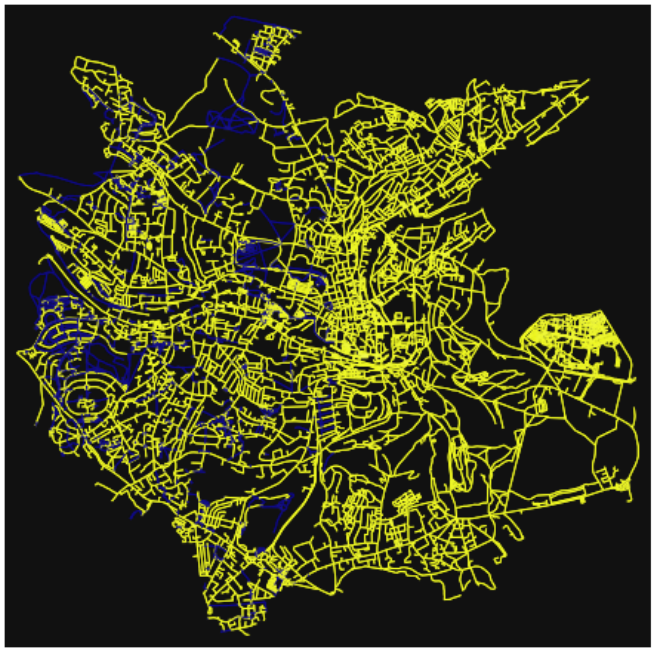


Figure 13: Figure showing the edge speeds which have only been manipulated by poor reviews in areas where there are alleys to the west of the circus (light is faster)

This section demonstrates the predictive power of machine learning methods in predicted user reviews in areas without any data. The edge speed could have been predicted directly, however the predicted reviews enables a hot start to the review and model updates if reviews to begin in this area without ignoring all other features. See the future improvement sections to see how this could be extended with more features to provide considerably more realistic and useful predictions.

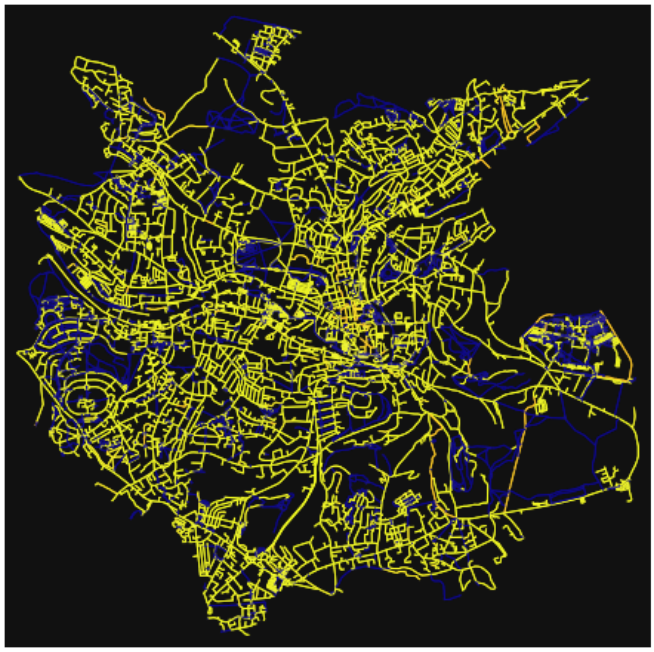


Figure 14: Figure showing the the edge speeds which have only been manipulated by only the *predicted* reviews. It should be seen that the west of Bath is identical to the original in 13, and indeed the model does actual identify that the alley feature corresponds to the slower speed in the east of Bath. (light is faster)

4.7 SafePal

4.7.1 Overview

In the field of safety app services, a niche has been identified in the area of addendum systems [5, 6]. It is not always sensible or conducive of a feeling of safety for a user to have their phone out to aid in navigation. Further, while having a phone out to navigate can mark a person out as a target of muggings, it will also distract from surroundings. Were their phone to be stolen, this would also leave them with no easy avenue to call for help without reliance on strangers. Bluetooth provides an avenue to somewhat combat this. With an addendum a link can be set up between a phone and a carried device. The link can be severed by removing the phone from the 10-20m range of the Bluetooth connection, which will then be able to alert the user and police to the predicament at hand. The process by which this is done is when connection is severed, both the phone and SafePal begin a 5 minute count down. The phone brings up a notification and requires its pin to prevent law enforcement being contacted, and SafePal will begin to regularly bleep and flash in blue. If the phone has been forgotten this will give the user sufficient time to return to it. Should it be forgotten in an irretrievable location such as on a train, law enforcement will be able to find the phone and see to the return of the device to the owner. If stolen, the phone thus acts as a disincentive in broadcasting its location.

The navigation function of the app is to be transmitted to a simple LED interface on SafePal to aid user navigation without having the phone out. This will enable inconspicuous use of the navigation app and give the user fewer distractions from their surroundings.

Another function that the device can fulfill is that of a personal alarm. The premise of such devices is to be set off and thrown when a person is accosted, setting off a throw away siren to draw attention to the user and require the potential perpetrator to either chase down the alarm, giving time to escape or accept

attention being drawn to them. With an app alone, this function is mitigated by the value of the phone. Further, SafePal is to have the capacity to use the phone connection to contact the police if a button on the addendum is pressed, providing a discrete way to call law enforcement without the conspicuous use of a phone. Were a phone a sufficient siren, to throw it away would deny the victim of a vital means of documentation and communication and risk damage of their phone with related ramifications. Activation of the the police call or alarm features will be transmitted by the phone to the app hosts. This will allow live crime data to be added to the mapping database and safety weightings. The data from this could be used to identify temporary clusters of crime and would serve in this way to make the system more responsive to changing safety circumstances. The Bluetooth connecting tab would be very cheap to reduce and replace, and could easily be armed with higher-powered speakers to be a more effective alarm so making a more effective device.

While many of these functions are available commercially, there has not been a known example of a single device unifying them and having all available in a convenient package will improve the safety of the app user.

4.7.2 Morphology

From this, the following requirements were established as desirable features of SafePal:

- A Bluetooth connection with the users phone
- The ability to contact the police via the phone in an emergency
- The ability to cause the phone to track the holder if the blue tooth connection is unexpectedly severed to dissuade and identify thieves
- Functionality as a personal throw away alarm
- A directional indicator interface to aid with user navigation
- Sufficient cheapness to throw away without significant concern
- The ability to be used with relative subtlety

To fulfil these requirements, a morphology study was undertaken. This identified in a brainstorm fashion approaches to fulfill each of the above requirements to identify features and design choices to implement into SafePal. Candidate device designs were then identified and linked to identify an efficient design solution. The morphology chart can be found in Figure 15, with the selected features in Green.

4.7.3 Device Prototype Design

Usability was the primary concern in the design of SafePal, it having been stressed during discussions that discrete usability would be critical to the success of the product. Careful consideration was given to the ergonomics of the prototype. This involved giving users a minimal interface with two buttons and five LEDs intended to make use straight-forwards in stressful situations. The police call button is in a purposefully awkward location on the device to ensure any pressing is purposeful, and requires a sequence of five independent presses in under five seconds to create a response to avoid accidental calls which waste police time. Police calls can also be cancelled for 10 seconds after activation by pressing both buttons simultaneously, and the central LED flashes blue during this period to alert the user of the immanent call. The panic alarm button, on the other hand, is located at the location a users thumb will rest when holding the device naturally, so if in a state of fear a user will come straight to that button without need

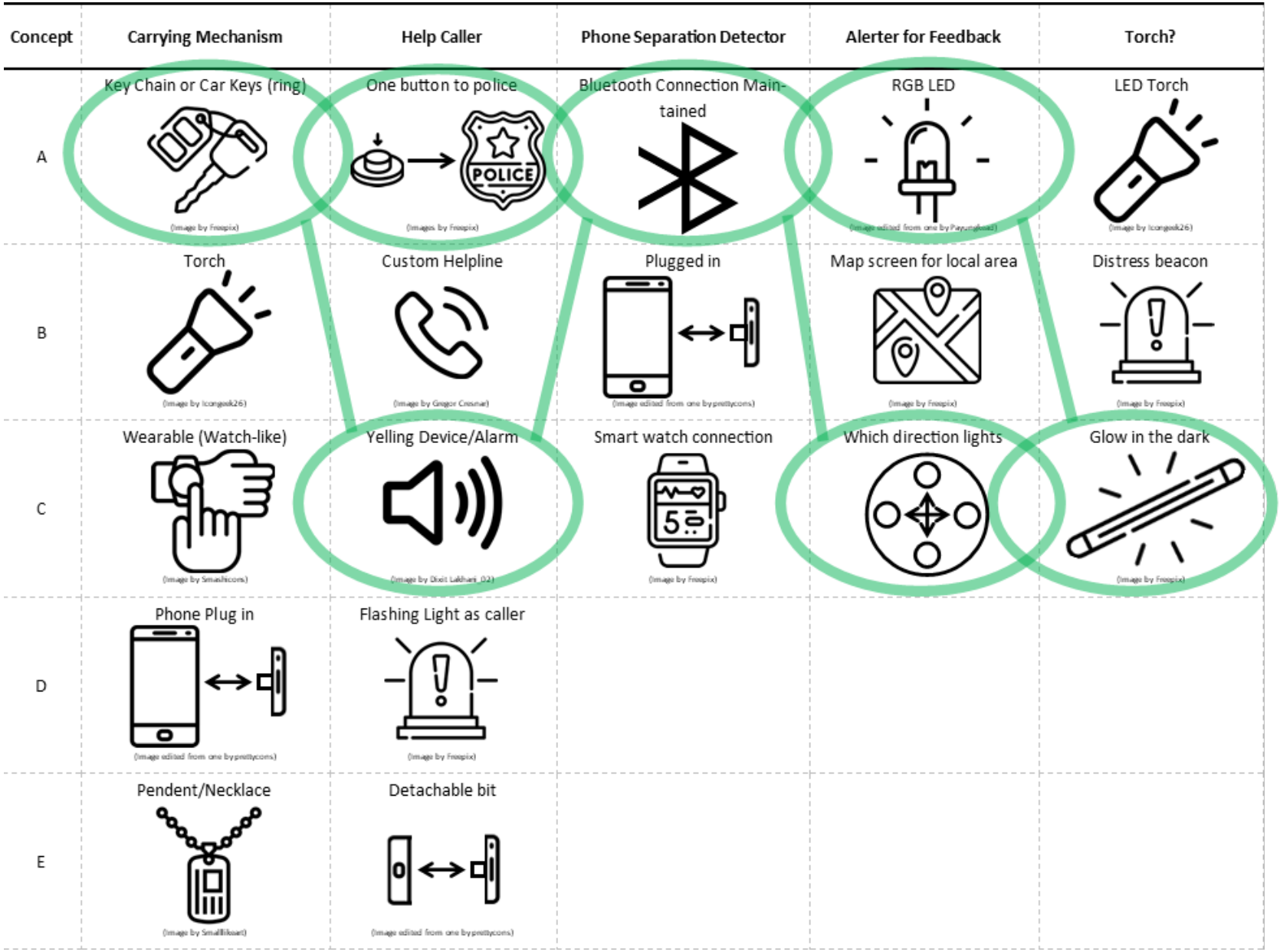
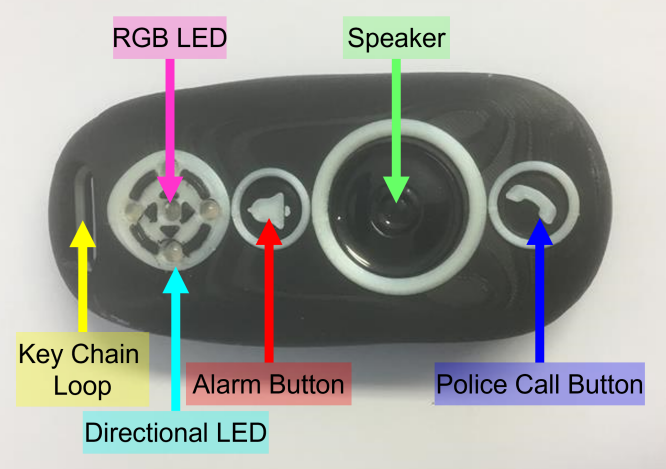


Figure 15: The SafePal morphology study

Figure 16: Key features of the SafePal hardware



to negotiate the geometry of the device. An inbuilt speaker is present on the trial device to act as the alarm, as would be the case on commercial versions.

The device has been designed as a key chain for ease of use, this making it simple for users to remember to take when they leave on a journey, keeping it close to but separate from the users phone and ensuring that it need not have a users address attached to it to facilitate potential stalking. This balance between ease of divestment and memory justifies this form for the product being selected. The bulk material of the part is PolyLactic Acid (PLA). It is a cheap plastic but suitably durable and water proof. It has the added advantage of being a bio-degradable plastic It further has as a prototype been built with the intended glow-in-the dark embedded body, which will keep a low level of glow for around 5 hours after exposure to light allowing the device to be used in the dark.

While a printed circuit board would in production be specifically made for SafePal, the test part was built using an Arduino Nano 33 BLE. This microprocessor has Bluetooth built into it and would be able to connect to an app when required. The final function of the device, which the Bluetooth would connect to, is the LED UI. To simplify and avoid the complications of calibration, the device does not have an accelerometer and is unable to establish its orientation relative to the users phone. The users phone, however, would be able to identify its location. Supposing the user holds the device with the keys hanging off the front, as would be the natural way to hold it, four of the LEDs act as a guide and will receive location and navigation data from the phone. This will be displayed indicating the direction the user should go to follow their safe route and when they should turn. That information would be supplied by this to a user discretely and without the risk of using the phone while walking, which to a greater extent diminishes situational awareness and would mark the user out as a potential target for criminals.

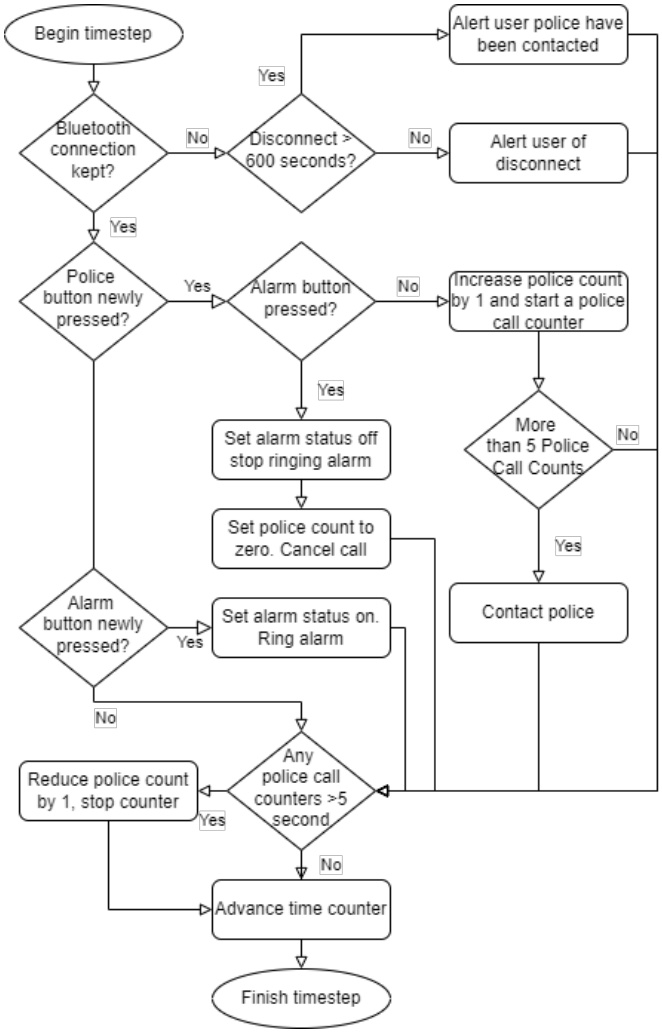
The central fifth LED serves the purpose of flashing blue when a police call has been triggered before it is sent, to alert the user to this and allow cancellation if accidental.

A Circuit Diagram can be found in Figure 42 in Appendix B.

4.7.4 Working Algorithm of SafePal

SafePal when active operates a simple time loop, which iterates at 1000hz frequency. The setting of each LED and the speaker are set to variables which are then read and outputs set. Each loop, SafePal checks if buttons have been activated then updates the display LED and speaker parameters as appropriate. A loop iteration follows the following flow chart:

Figure 17: The time step flow chart for SafePal



5 Future directions

- *Utilise additional sources of data* - Street lighting plays a clear role in the perception of safety. For this reason, it was attempted to incorporate street lighting into the edge weighting, using the data from OSMnx with the relevant tags as 'lit': 'yes' and 'highway': 'street_lamp'. For Bath, the corresponding data is visualised in Figure 18. Unfortunately, the data appears to be incomplete. One instance where this is apparent is the edge highlighted in blue as Widcombe Hill. A brisk walk or using Google street view will reveal that Widcombe Hill does indeed have street lamps, but not according to OSMnx. Assuming OpenStreetMap data isn't updated, this functionality could be implemented in the future by incorporating information derived from night time satellite image data.

- *Work with local organisations for updates* - Working specifically with local organisations such as the local police, Bath and North East Somerset Council, and Highways England would ensure high-profile road/bridge closures, temporary traffic lights or police cordons are known well in advance. This will ensure our routes are not only as safe as they can be, but are also feasible, so there is little chance users will have to re-route in the middle of their journey. Cooperation from local authorities

could also provide better data-sets - at present, the crime data we use is general, and we aimed to clean the data to only consider crimes most relevant to someone walking down a street. However our work is an approximation; therefore future work could entail working with authorities to develop a data-set of crimes directly relevant to a person walking down the street, including times of crimes.

- *Improve edge localisation in green space detection* - In the developed green space detection algorithm, if any part of an edge goes through a green space, the entirety of the edge is penalised. While this is appropriate in most cases, for longer edges, where one end may be far away from the green space, this isn't ideal. One solution to this would be to find specifically what part of the edge is in the green space, and then break the edge into two separate edges, one of which is tagged as going through a green space, the other isn't.

- *Improve feature heuristics and review predictions* - As is mentioned in the section "Review predictions" the OSMaps API has 1000's of ground features which can be added to the GeoPandas DataFrame. Some of these features may be able to make logical linear contributions to the weighting in equation (1). However, most of these will combine in a complex way to the safety or perceived safety of a traveller. Therefore a proposed future method is to add these 1000's of features to each of the edges in the DataFrame. Eventually the user reviews will outweigh the heuristics in equation (1). But in areas where reviews are sparse, the reviews can be much more accurately predicted from the many more features, and should provide a powerful and accurate safe routing algorithm.

- *Account for age of crimes with additional models* - Three KDE models were trained for this work, one for each crime severity level, so that different types of crime could affect the routes with greater or lower magnitude. However, these models do not account for how old a crime in the dataset is. A potential solution to this could be train separate models, for example, one on crimes that occurred within the previous year and another older crime data. Thus, if one wished to weight newer crimes more strongly than older crimes, the temporal models may be used in the routing algorithm in just the same way as the different crime severity models.

- *Adaption of SafePal to Production Standards* - The functionality of the final device was captured by the prototype in entirety, demonstrating the plausibility of SafePal. SafePal, however, has not been actively connected to Bluetooth during testing to date. This would take further work to achieve. Manufacturing would require updates also. A commercial device would be scaled down by a factor of around 3, in order to make it easier to pair with a set of keys. The scale change would require a smaller chip, as the current scale was limited by the prototype Arduino and battery pack. A more refined design could easily reduce the scale of these parts. Finally, while the casing basic design proved apt, 3D printing is not an economical approach to manufacturing a mass-produced part. An injection moulded case of similar design would be a more practical design, but is not a practical approach for a prototype part because the approach needs substantial and expensive mould parts.

- *Reducing harmful bias in crime data* - as discussed in Section 6, there are known racial disparities in policing, which will be reflected in crime data. An area of future work would be to run a full analysis of the harmful bias in the data we are using, and take steps to reduce harmful side-effects in app. An example harmful side-effect could be labelling areas populated with a high proportion of black people as dangerous due to biased data.

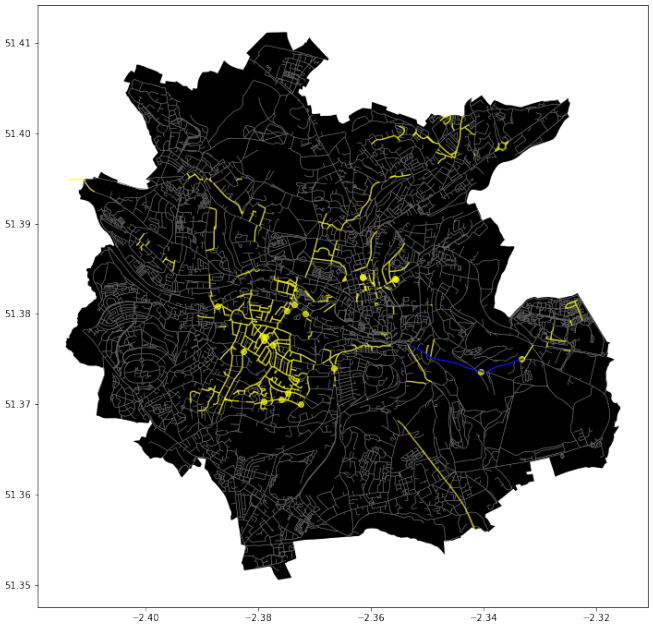


Figure 18: The yellow shows edges with street lights and areas that are lit up, according to OSMnx data. The blue edge shows Widcombe Hill.

6 Critical reflection - A-R-T and ethical concerns

The design of our application was made considering the ART-AI themes of **accountability, responsibility, and transparency,** as well as ethical issues surrounding **confidentiality and data biases** amongst others. Below, we outline these considerations, in addition to how future work can ensure these objectives are met.

Accountability: We feel that the vast majority of existing applications, such as Safe and the City, pushed the issue of accountability of finding a safe route onto the user, despite poor safety judgement from the average individual. Our approach was more balanced – by routing the user we acknowledge our application is at least partially accountable should something go wrong, but decided this is an important feature for our app to be useful.

We use user-feedback as an input to our algorithms, and recognise that using user feedback entails the risk of manipulative input. However, we require a user to sign up for the application – this hopefully increases user accountability to the extent it defers nefarious use. A future step could involve analysing patterns of nefarious use such as consistently down-voting a route compared to others in a certain area.

Responsibility: As outlined in the above section, by asking users to create routes, existing apps also make the user more responsible. We once again aimed to balance this responsibility more evenly between the app and user. By routing the user, we in effect take responsibility for their safety, and we have aimed to clarify this responsibility via the notice that we will contact the police on their behalf if they commit to sharing their location data, and by reminding users that our app can never guarantee safety. However, by providing a list of optional features – such as the slider option between 'safer' and 'speedier' routes – we in effect make the user partially responsible for an experience that is moulded to their personal view

of safety.

Transparency: The application – and indeed, any application – cannot make guarantees of safety. The heuristics we employ are proxies for safety and feeling safe. To maintain transparency, we provide the user with options such as switches for green space avoidance amongst others. This creates transparency by allowing the user to see exactly what is being optimized and the extra information 'bubbles' provide the clarification. In addition, efforts were made to include a highly transparent algorithm that could be easily explained where needed. However, we made a conscious decision not to share a visual representation of the crime data with users, reducing some transparency. We felt that showing the crime data to a user risked biasing a user against certain areas even if only a small number of crimes had occurred there, potentially risking the livelihoods of businesses in the area from reduced footfall. Ideally, in future work, we would aim to understand how to be transparent with this visual representation without biasing a user about the crime level of certain areas.

Confidentiality and informed consent: We are clear in the description of the use of location data and sending this to the local police service that location data and email addresses etc. will only be used for that specific purpose. We are also clear in asking for consent for the sharing of this data, and tell the user which features will be unavailable if they do not want to provide this information.

References

1. Office for National Statistics. Perceptions of personal safety and experiences of harassment, Great Britain. 2021. Available from: https://www.ons.gov.uk/peoplepopulationandcommunity/crimeandjustice/bulletins/perceptionsofpersonalsafetyandexperiencesofharassmentgreatbritain/2to27june2021
2. Foster S, Wood L, Christian H, Knuiman M, and Giles-Corti B. Planning safer suburbs: Do changes in the built environment influence residents' perceptions of crime risk? Social Science and Medicine 2013; 97:87–94. DOI: 10.1016/J.SOCSCIMED.2013.08.010
3. Google. Stay informed and get around safely with Google Maps. 2020. Available from: https://blog.google/products/maps/stay-informed-and-get-around-safely-google-maps/
4. Google. "Stay Safer" launching for Indian Android Maps users - Google Maps Community. 2019. Available from: https://support.google.com/maps/thread/8723647/\%E2\%80\%9Cstay-safer\%E2\%80\%9D-launching-for-indian-android-maps-users?hl=en
5. i3Intelligence. Keep Safe on The Go — Safe & the City. 2022. Available from: https://www.safeandthecity.com/
6. WalkSafe. FAQ — WalkSafe. 2022. Available from: https://www.walksafe.io/faq
7. Perkins C, Beecher D, Aberg DC, Edwards P, and Tilley N. Personal security alarms for the prevention of assaults against healthcare staff. Crime Science 2017; 6(1):1–19. DOI: 10.1186/S40163-017-0073-1/TABLES/1
8. Boeing G. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Computers, Environment and Urban Systems 2017; 65:126–39
9. Zhang F, Fan Z, Kang Y, Hu Y, and Ratti C. "Perception bias": Deciphering a mismatch between urban crime and perception of safety. Landscape and Urban Planning 2021; 207:104003
10. Ceccato V, Canabarro A, and Vazquez L. Do green areas affect crime and safety? *Crime and Fear in Public Places*. Routledge, 2020 :75–107. ISBN: 0429352778

11. What factors are linked to people feeling safe in their local area? Available from: https://gov.wales/sites/default/files/statistics-and-research/2020-03/what-factors-are-linked-to-people-feeling-safe-in-their-local-area.pdf [Accessed on: 2022 Apr 26]

12. v1.0 WAG. Contrast & Color. Carnegie Museums of Pittsburgh. Available from: http://web-accessibility.carnegiemuseums.org/design/color/

13. Web A. Minimum font size? Accessible Web. Available from: https://accessibleweb.com/question-answer/minimum-font-size/

14. Best Fonts To Use for Website Accessibility. Bureau of Internet Accessability. 2017. Available from: https://www.boia.org/blog/best-fonts-to-use-for-website-accessibility

15. OpenStreetMap. Available from: https://www.openstreetmap.org/#map=17/51.37777/-2.35337&layers=T

16. Map features - OpenStreetMap Wiki. Available from: https://wiki.openstreetmap.org/wiki/Map_features [Accessed on: 2022 Apr 25]

17. Gillies S. The Shapely User Manual. 2022. Available from: https://shapely.readthedocs.io/en/stable/manual.html [Accessed on: 2022 Apr 26]

18. Table of offences scheme c class order. Available from: https://www.cps.gov.uk/sites/default/files/documents/publications/annex_1b_table_of_offences_scheme_c_class_order.pdf [Accessed on: 2022 Mar 25]

19. What do the crime categories mean. Available from: https://www.police.uk/pu/about-police.uk-crime-data/ [Accessed on: 2022 Mar 25]

20. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, and Duchesnay E. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 2011; 12:2825–30

21. data.police.uk. Data downloads. 2022. Available from: https://data.police.uk/data/

22. Ester M, Kriegel HP, Sander J, and Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 1996 :226–31

23. Rokach L and Maimon O. Clustering Methods. *Data Mining and Knowledge Discovery Handbook*. New York, USA: Springer, 2005. Chap. 15:320–52

24. McLachlan G. Mixture Models. New York, USA, 1988

25. Parzen E. On Estimation of a Probability Density Function and Mode. The Annals of Mathematical Statistics 1962; 33:1065 –1076. DOI: 10.1214/aoms/1177704472

26. Rosenblatt M. Remarks on Some Nonparametric Estimates of a Density Function. The Annals of Mathematical Statistics 1956; 27:832 –837. DOI: 10.1214/aoms/1177728190

A Unsupervised Learning Visualisations

A.1 DBSCAN and Agglomerative Clustering

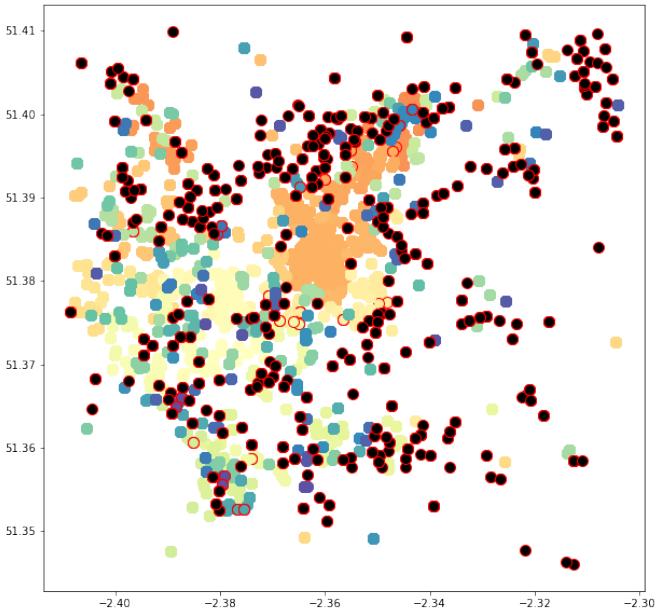


Figure 19: Visualisation of the crime clustering DBSCAN model. Separate clusters are coloured differently to their neighbours. Points with red outlines are cluster edge points as found by the DBSCAN algorithm. Black points with red outlines are noise points.

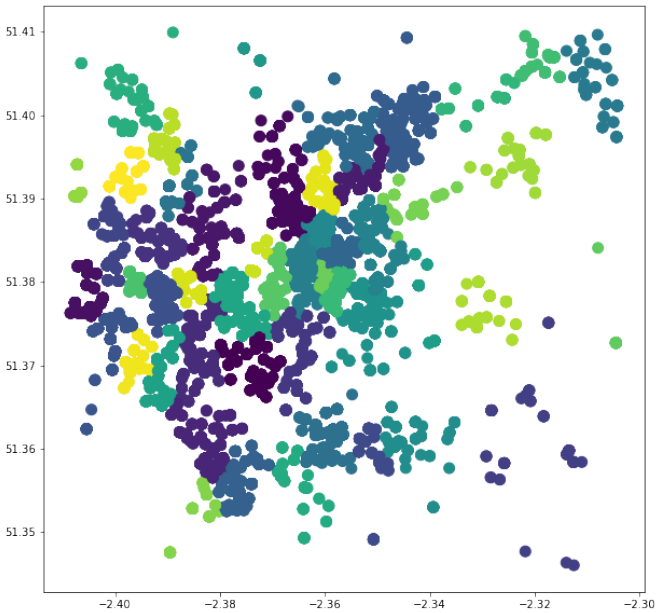


Figure 20: Visualisation of the agglomerative clustering crime model. Separate clusters are coloured differently to their neighbours.

A.2 Gaussian Mixture Model

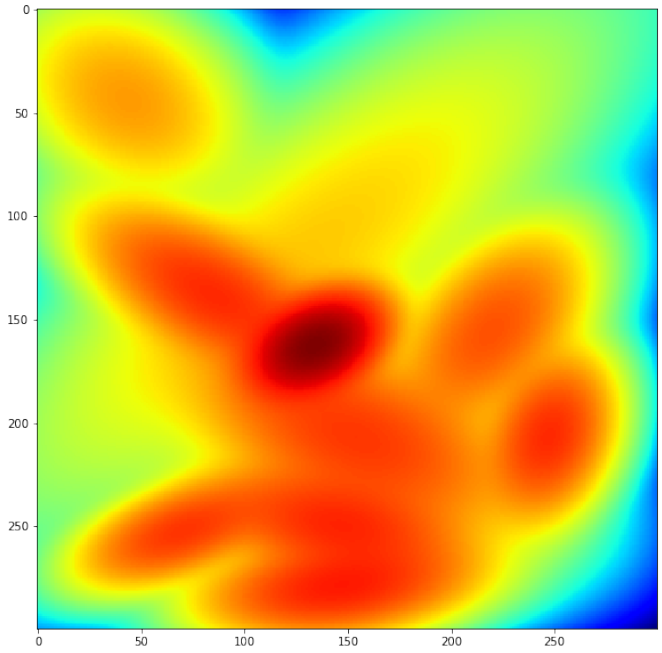


Figure 21: Visualisation of the GMM model with 10 components. The heatmap corresponds to the log-likelihood of each point under the GMM model trained on the Bath crime dataset. Red indicates higher log-likelihood under the model, which corresponds to greater amounts of crime.

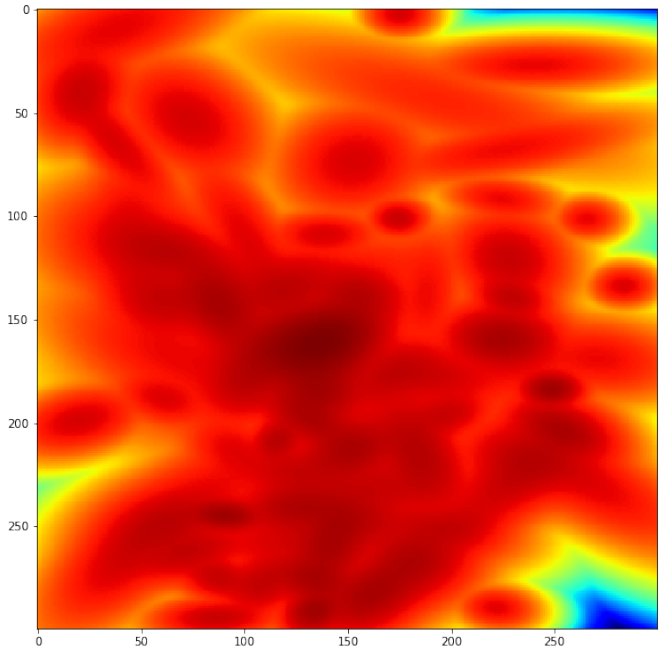


Figure 22: Visualisation of the GMM model with 100 components.

A.3 Kernel Density Estimation Kernels

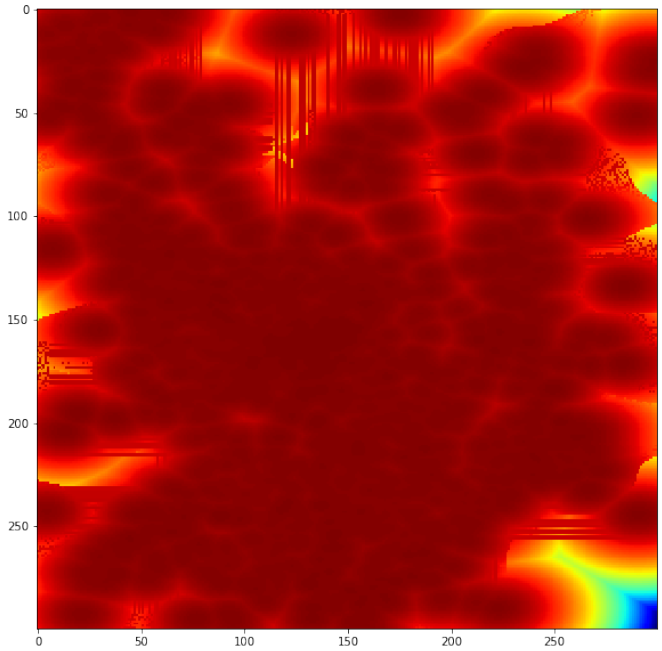


Figure 23: Visualisation of a Gaussian kernel KDE model.

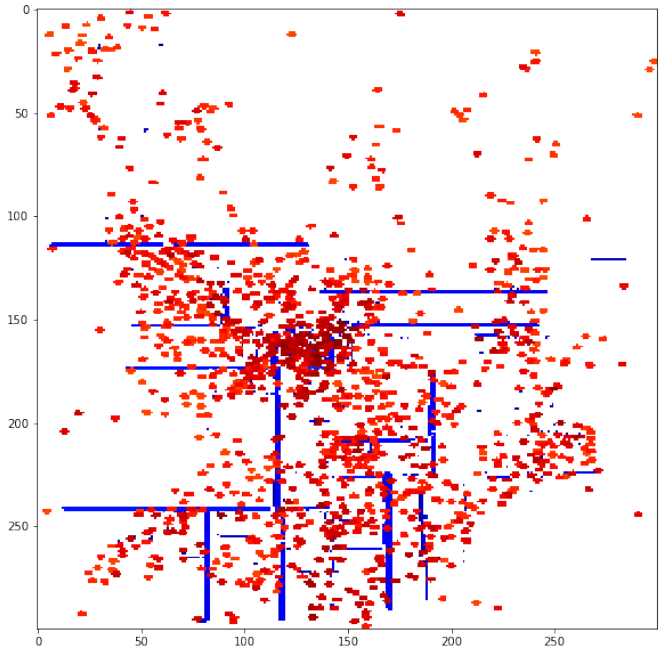


Figure 24: Visualisation of a Tophat kernel KDE model.

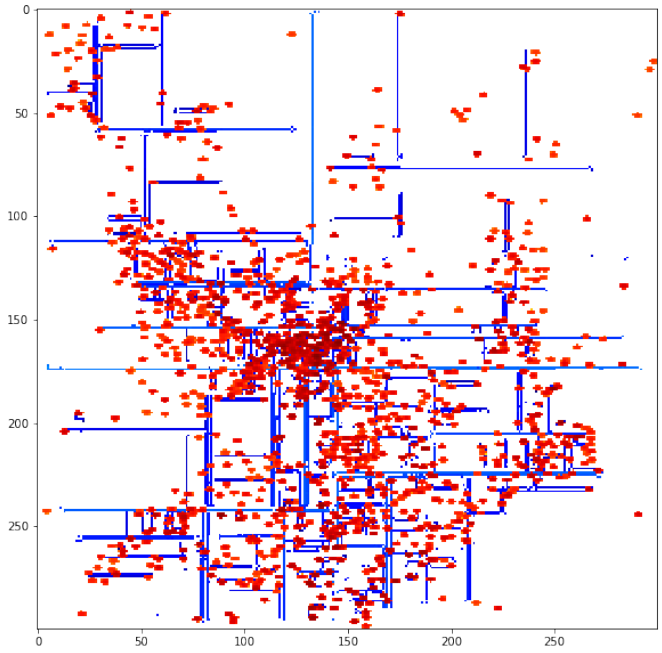


Figure 25: Visualisation of an Epanechnikov kernel KDE model.

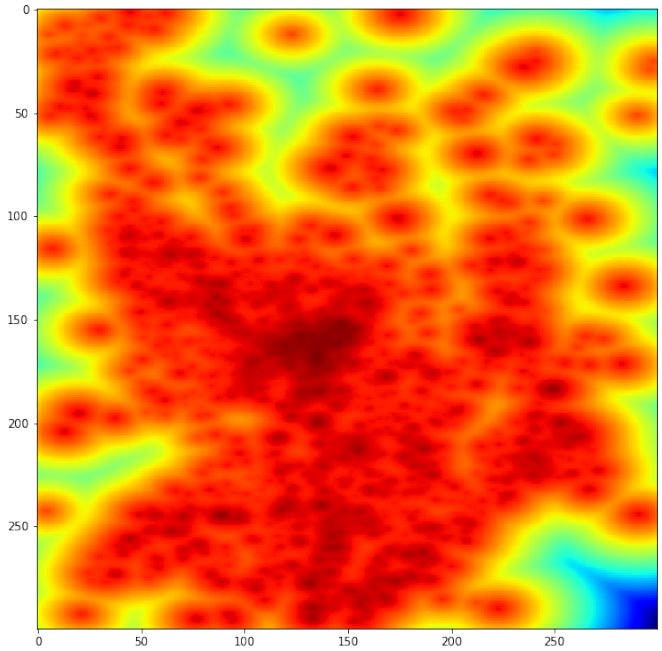


Figure 26: Visualisation of an exponential kernel KDE model.

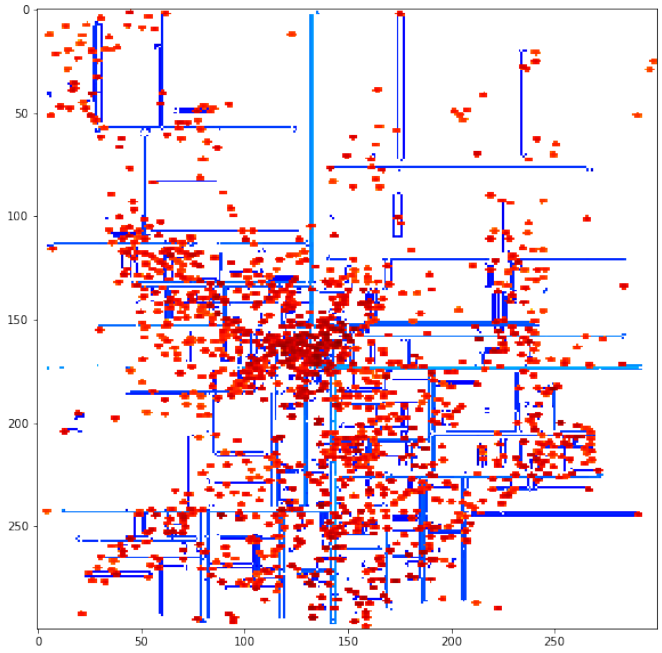


Figure 27: Visualisation of a linear kernel KDE model.

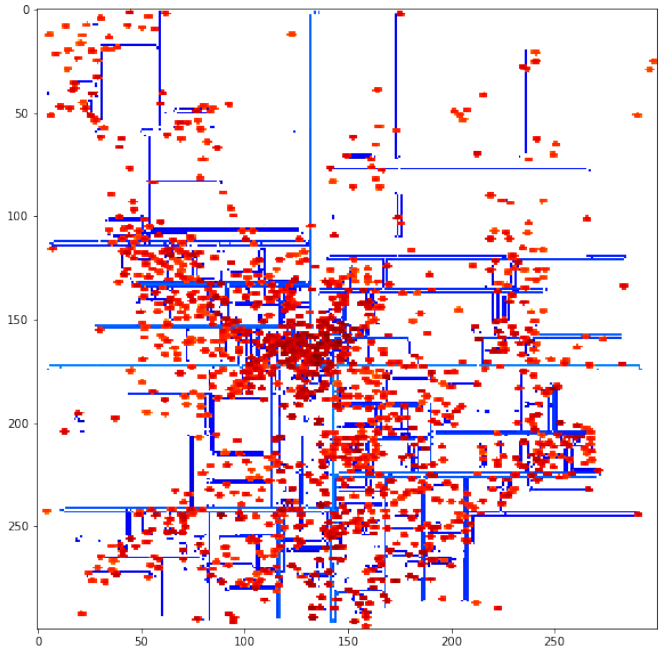


Figure 28: Visualisation of a cosine kernel KDE model.

A.4 Bandwidth Tuning

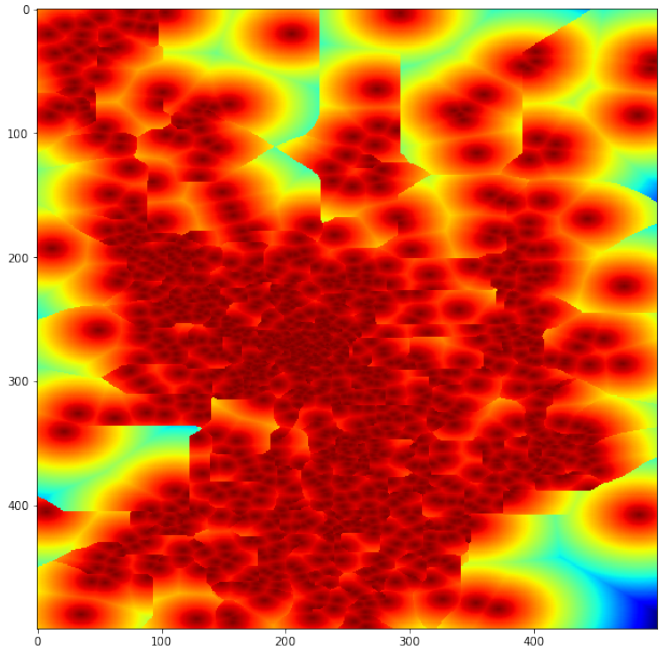


Figure 29: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 1e-9.

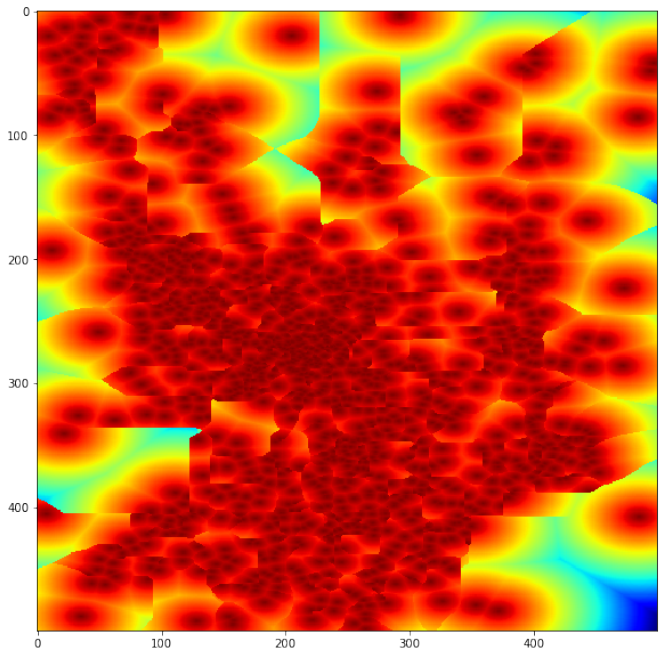


Figure 30: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 1e-8.

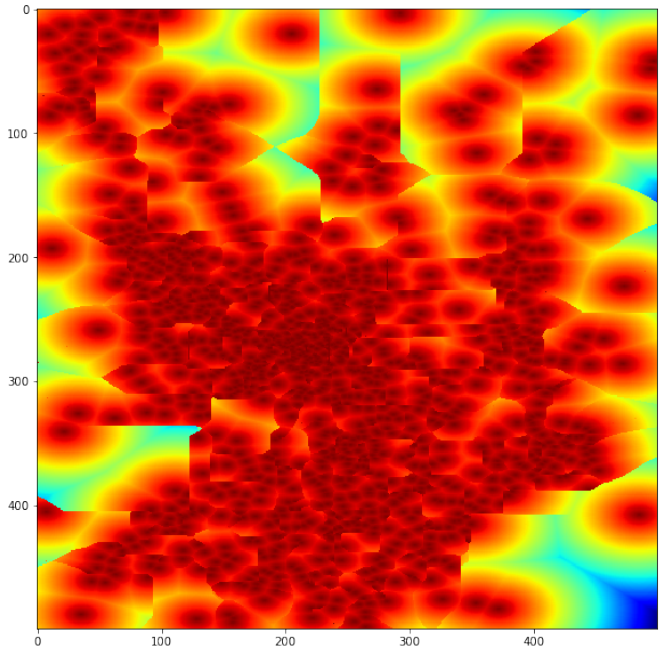


Figure 31: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 1e-7.

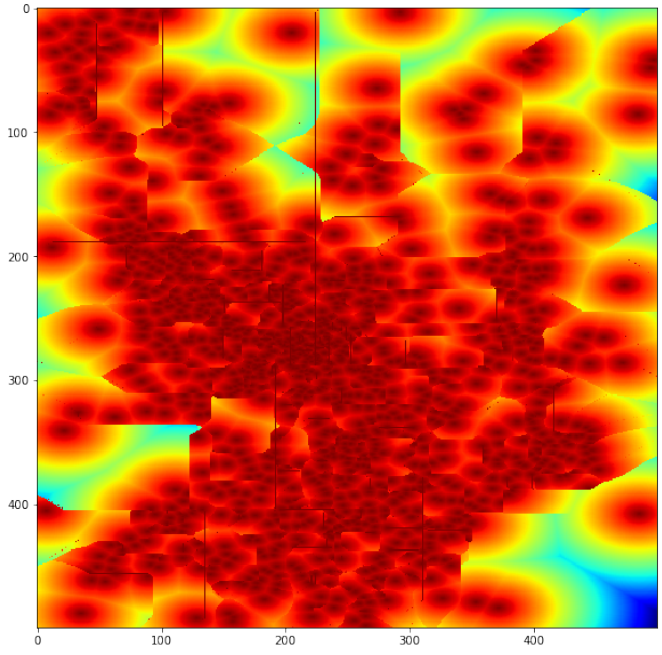


Figure 32: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 1e-6.

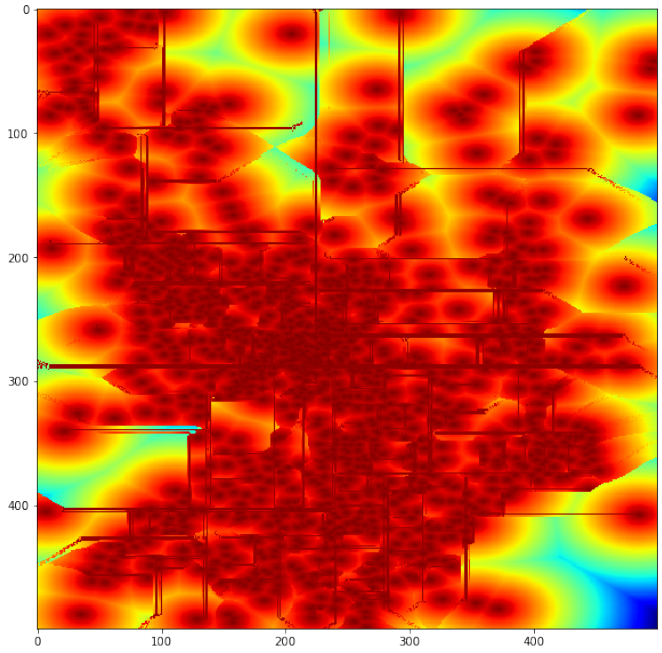


Figure 33: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 1e-5.

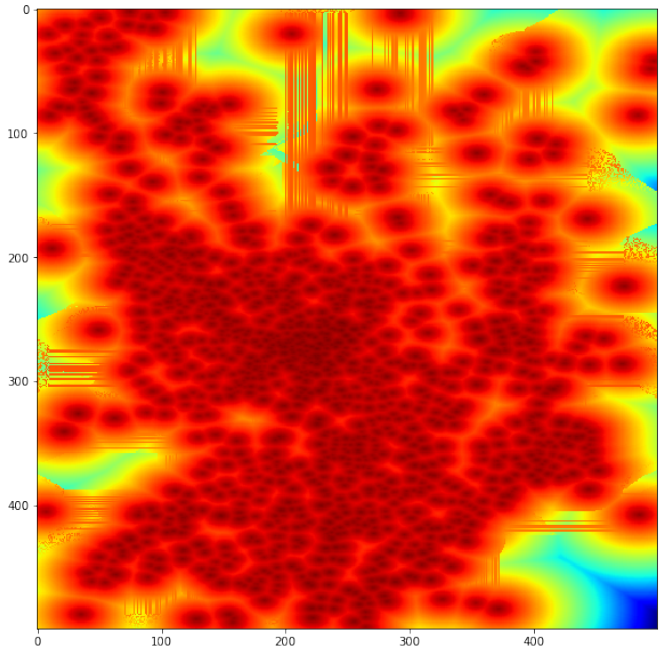


Figure 34: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 0.0001.

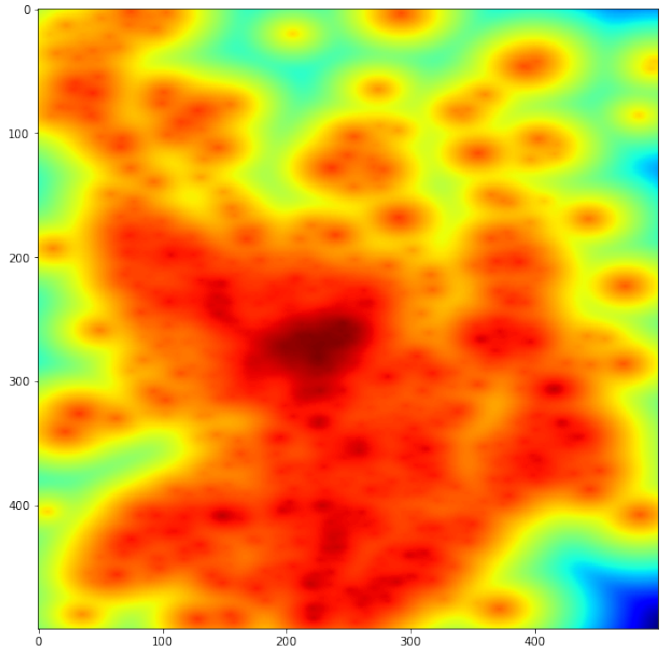


Figure 35: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 0.001

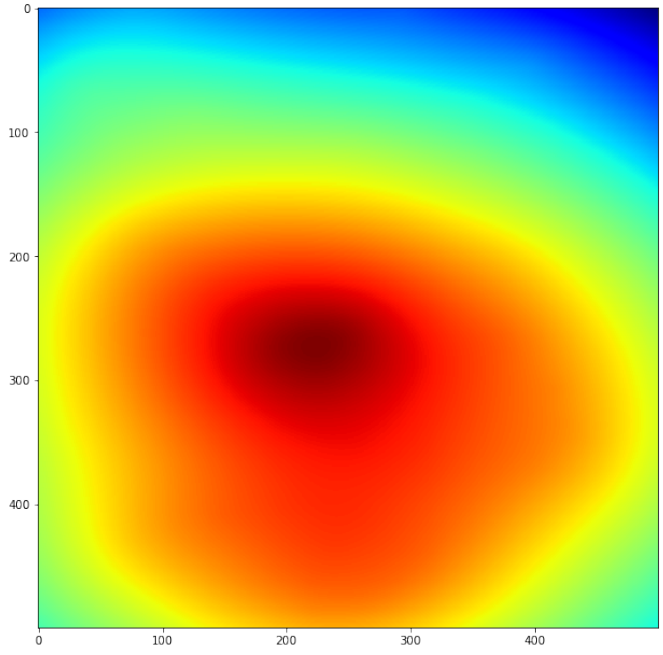


Figure 36: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 0.01

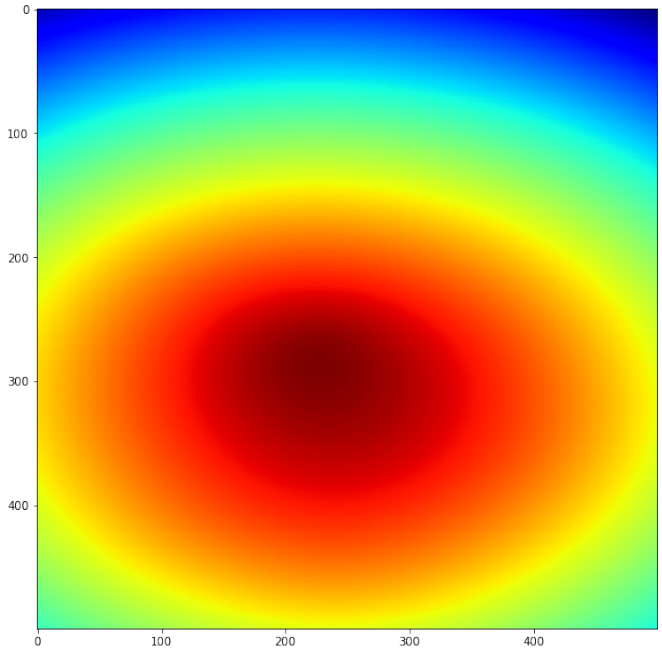


Figure 37: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 0.1

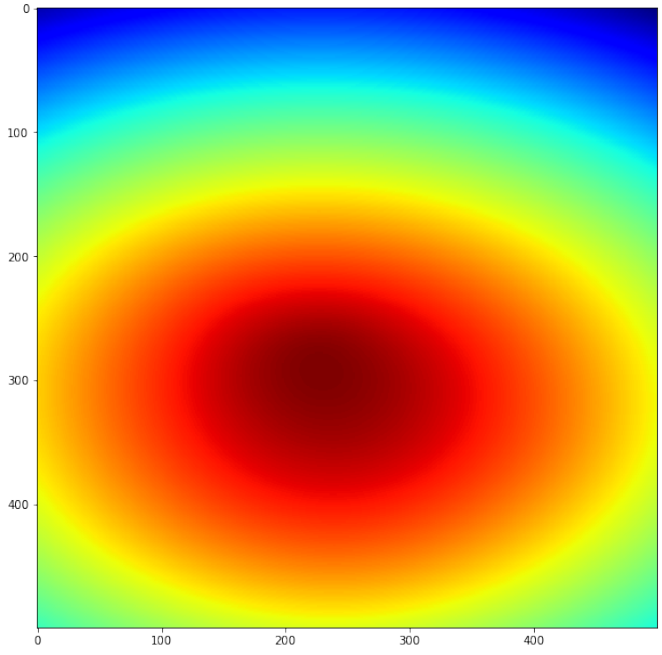


Figure 38: Visualisation of the exponential kernel KDE model with bandwidth parameter equal to 1.

A.5 Different Crime Severity Models

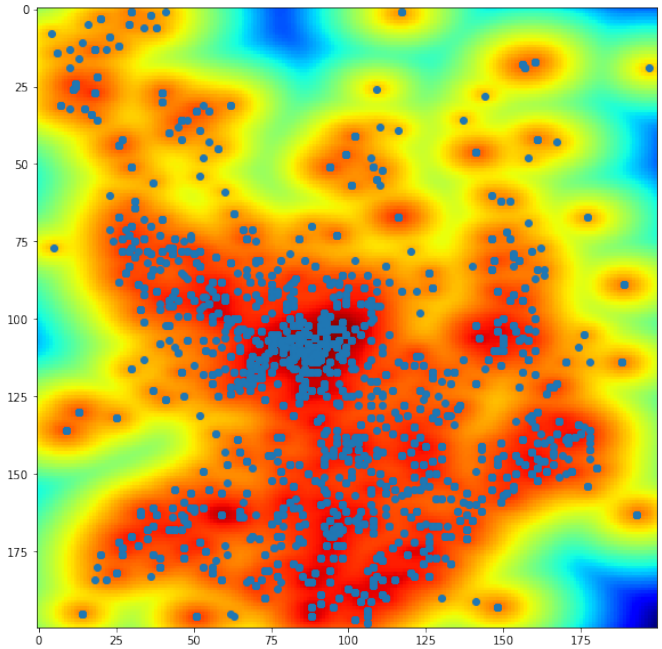


Figure 39: Visualisation of the KDE model trained on high-severity crime data (plotted on top of the heatmap).

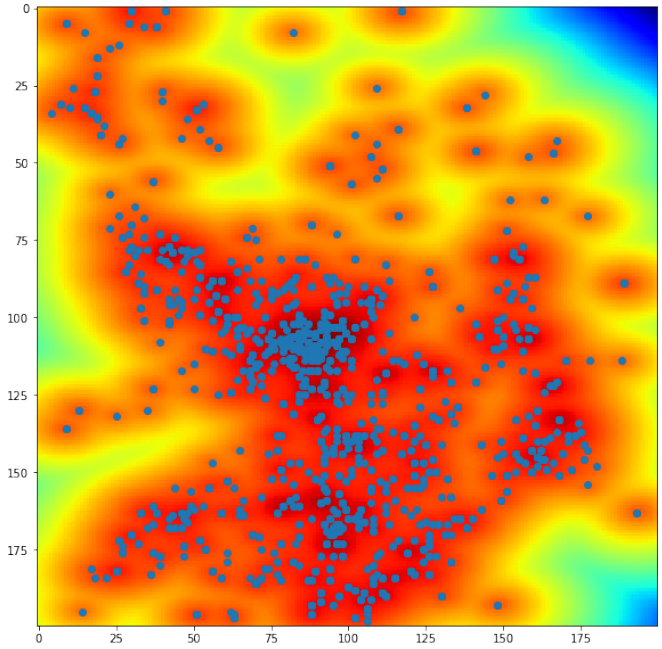


Figure 40: Visualisation of the KDE model trained on medium-severity crime data (plotted on top of the heatmap).

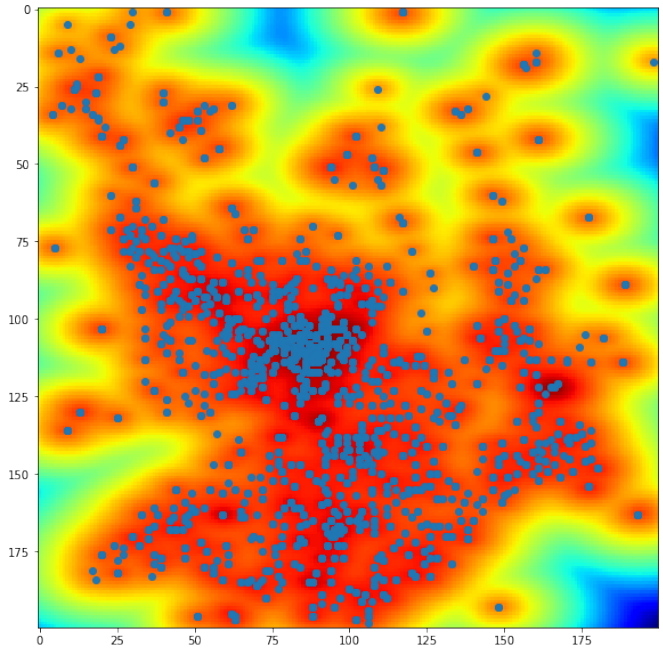


Figure 41: Visualisation of the KDE model trained on low-severity crime data (plotted on top of the heatmap).

B Circuit Diagrams

Figure 42: The circuit diagram of SafePal