Detecting Social Events Using Mobile Phone Network and Social Media Data

Author: Marc Tatam

Supervised by: Riccardo Di Clemente

Abstract:

This report investigates the feasibility of a novel approach of using neural networks on a combination of mobile phone and social media data to detect social events. Most of the literature in this field only briefly touches on the possibility of detecting social events from either data source and none investigates the usage of both data sources. The main aim of this report will be to provide a proof of concept of combining these two streams of data to create a method for automatically detecting social events. The chosen method for performing this task is the use of neural networks to detect social events from each data stream individually and to combine these results. The results are then examined to identify some of the benefits and drawbacks of this approach.

I certify that all material in this dissertation that is not my own work has been identified.

Table of Contents

1. Introduction and Motivation

Across the world, almost every settlement hosts events to bring its citizens together and to create a sense of community, be it a village fete in a rural village or hosting the world cup in billion-dollar stadiums designed to cope with tens of thousands of spectators. Regardless of size, any event which aims to bring people together can be classed as a social event. These events have a huge impact on our lives, even if one does not partake in them. A small village fete may attract visitors from neighbouring villages, making it difficult to park, whereas the world cup is guaranteed to bring visitors from much further away causing strains on the local infrastructure. For pre-planned events like these, challenges can be managed as it is likely that some sort of prior consultation has taken place with the regional government. However, this is not possible for all social events such as those created without warning, for example, impromptu protests or unexpected events like a bomb threat. Initially, it may seem that events like a bomb threat would drive people away rather than together, but such events can also attract media coverage and crowds of people who are curious about what is going on. The ability to locate, detect, and classify such events is highly beneficial to manage and mitigate their impact.

Since the turn of the century, few inventions can claim to have had as much impact on everyday life as the mobile phone. This technology allows us to be more connected than ever, be that through direct messaging or sharing where a person is and what they are doing. Consequently, this abundance of information could be used to reflect a person's real-life activities, such as their attendance at a social event. As mentioned, certain events could trigger gatherings of thousands if not tens of thousands of people, causing a spike in the level of load on the mobile phone network in that area, meaning that the event could automatically be detected based on the data volume, which allows some sort of response to be deployed automatically. For mobile phone network providers, this could be the deployment of additional infrastructure, and for the local government, this could mean deploying extra policing resources to maintain order in the affected area. Likewise, there may be lots of posts on social media when an event occurs, through the posting of pictures or by direct reference to an event in a certain area. Subsequently, this may cause an exceptional level of usage of certain words which could indicate what the event is. There is a major piece of evidence which supports this theory which is the 2014 football world cup semi-final. Whilst many football fans are aware of the result of this game, it is an infamous defeat of Brazil 7-1 by Germany, fewer know that this single game produced 35.6 million Tweets worldwide [1].

This report sets out to validate the potential to detect social events by using information in the form of mobile phone network activity data and Tweets from the city of Milan, Italy, during the last two months of 2013. While there have been variants of this approach explored (these will be discussed in the literature review), there are none that use combined social media and phone network data to detect social events, hence, the motivation for testing this approach.

Consequently, this report will provide a summary of an approach which attempts to tap into this wealth of information to allow proper responses to social events, with the main aims being:

- To investigate the feasibility of developing neural networks that can automatically detect social events from combined Tweets and mobile phone data
- To develop an approach that will allow the combination of the nature of an event and the location of an event.

2. Summary of Literature Review and Specification

Having presented the motivation for the development of the approach, the previous work in this area will be explored, and consequently a specification will be set out.

2.1. Literature Review

Mobile phone networks have a hierarchical structure with base transceiver stations (BTS), which are the physical antennas at the lowest level [2]. The BTSs are connected to the network infrastructure which connects to the control station that in turn allows internet connectivity [3]. Generally, more densely populated areas will have more BTSs so that the network can deal with the additional load. A side effect of this is that positional accuracy is lower in regions with fewer BTSs. The hierarchical structure creates two main types of mobile phone network data, both of which are collected by detecting which BTS a person is connected to at any given moment. The two types of mobile phone network data are Call Detail Records (CDR) and aggregated data. A CDR is recorded every time a mobile phone makes an interaction with the network [4], be it a call made, a text sent or received or trying to get a webpage from the internet. On one hand, CDRs are significantly more granular and allow for the tracking of an individual's movements over time, but on the other hand, there are privacy concerns in using CDRs and therefore they are much harder to obtain. Aggregated records provide the level of activity over some period in an area across all participants. For example, when monitoring call hours, one person calling for one hour would have the same weight as four people calling for 15 minutes each. The benefit of aggregated data is that it is much easier to obtain, however, it has the disadvantage of being far less granular.

There are two main event types exhibited by a mobile phone network, network events and social events, both of which are detectable from CDRs using neural networks [5]. Network events have an impact across the entire network at the same time and are characterised by a large increase in demand, due to factors such as a new show being released on a streaming site. Social events are events such as art exhibitions or football matches which would create a physical gathering of people [6]. The latter type of event is characterised by an increased level of activity in a certain area. When reading the literature, it appears that this is the first area for potential future development, as previous methods for detecting social events used CDR data or obfuscated CDR data (CDRs with personal information removed) as opposed to aggregated data. As it is much easier to obtain aggregated data as well as aggregated data being derivable from CDR data and therefore approaches using aggregated data can be used in scenarios where only CDR data is available, developing a method of detecting social events using aggregated data is preferable.

To reduce the complexity of detecting social events from mobile phone data, it is necessary to identify the land use of various regions, as the activities in certain areas may vary by time and even by day [7]. The first method that has been explored for land use detection consists of using the activity pattern in an area. When an area is more densely populated, likely, the activity level of the mobile phone network in that area will be higher. Since different land uses will attract people at different times, for example, offices are unlikely to have a high number of workers present late at night or leisure areas will have a low level of occupancy during a weekday, it should be possible to extract the land use based on the mobile phone activity level [7]. Also, the spatial configuration from mobile phone data can be detected by monitoring how people move between different parts of the city [8]. For example, most people would probably move from a residential area to the area where they work in the morning, therefore a person's location during different periods of a day may indicate what the land use is in each of these locations. However, this method is difficult to apply with aggregated data, therefore using the activity pattern will be the preferred method in the approach.

The event types displayed on social media are more varied than those displayed on mobile phone networks. While social media will display both network and social events (which are pull events), they will also show behaviour characterised by push events [9]. Push events are events that drive people away from a specific area such as road closures or natural disasters. Though social media can be used to detect social events [10], there are some drawbacks to their use in an isolated approach. For example, if one was to purely use venue names to detect social events, venues that have sold their naming rights such as the O_2 in London would see increased usage when the O_2 mobile phone network is down leading to a false positive.

Hence, combining mobile phone network data with social media data would be highly beneficial as it can reduce the number of false positives. Generally, it seems to be more likely that social media would display the cause of an event rather than the location of an event. Therefore, a combination of social media data and mobile phone data would permit locating events as well as their cause.

2.2. Specification

The main objective of the approach will be to create a model capable of detecting social events using correlated mobile phone and social media data. The main hypotheses are as follows:

- Can Twitter be used to identify social events and their causality?
- Can mobile phone network data be used to identify social events and their locality?
- Do Twitter and mobile phone network data both identify the same social events if combined?

So far there has been very little research performed in this field, thus the key aim will be to provide supporting evidence that these hypotheses hold. Achieving some predefined accuracy in detecting events is secondary.

There are several key functional requirements:

- It should be possible to download Tweets from the Twitter API.
- The implementation should be able to determine the usage of a specific word from the Tweets.
- It should be possible to open the files containing the mobile phone data.
- Land use classification should be determined from the mobile phone data.
- Functionality should be developed to normalise and prepare the data for event detection.
- An automated method for detecting events from the two sets of data should be developed.
- A way of correlating the detected events should be defined and created.

In terms of non-functional requirements, any implementation must be able to run smoothly across various devices, without significant configuration requirements. Any graphs or maps should either be images or stored using some file format that is accessible on a wide range of devices, and they should be easy to interpret. There are unlikely to be any legal or ethical issues as the source mobile phone data is already anonymised and the Twitter users will have agreed to have their data shared when signing up to the website.

This report will largely focus on events that were held at Milan's Giuseppe Meazza (also known as the San Siro stadium), all of which are football matches, as this venue is much larger than any other in the local area at a capacity of 75,817 [11]. Two teams have the San Siro Stadium as their home stadium, Football Club Internazionale Milano (Inter) and Associazione Calcio Milan (AC Milan). This large size means events are likely to be easily detected. Lots of people in Milan are likely to follow one of the teams on social media even if they cannot make it to the match, and therefore it is probable that these social events will have a large presence on Twitter. A list of other notable venues and events that took place during the time frame of any data will also be created, with the full list included in the appendices.

The time frame for the data used is from the 1st of November 2013 to the 1st of January 2014, with the data epoch being measured from 23:00 UTC on the 31st of October 2013. The approach will use static, historical data to prove feasibility as opposed to real-time data.

The overall design and development activities will be initially set out in a Gantt chart so that initial expectations on development time can be set. However, each development stage will be broken down into tasks that will be moved across a Kanban board. This will be facilitated by using GitHub as a version control system to allow testing of one stage to be performed independently of other stages. Using a version control system will also allow for the rollback of any changes if they are problematic. Overall, the combination of

these tools will support a high degree of organisation and for sufficient and efficient distribution of time across different development stages. Generally, components of the implementation should follow a microservice style architecture where the input and output of a given stage should be predefined, but the actual implementation should not matter as long as it meets the input and output requirement.

3. Design

Now that it has been defined which techniques can be used for this approach, the overarching design will be explored. The design follows two distinct processing tracks, one for mobile phone data and the other for Twitter data, with the two meeting at a final stage where events are detected, as is visualised in figure 1 below:

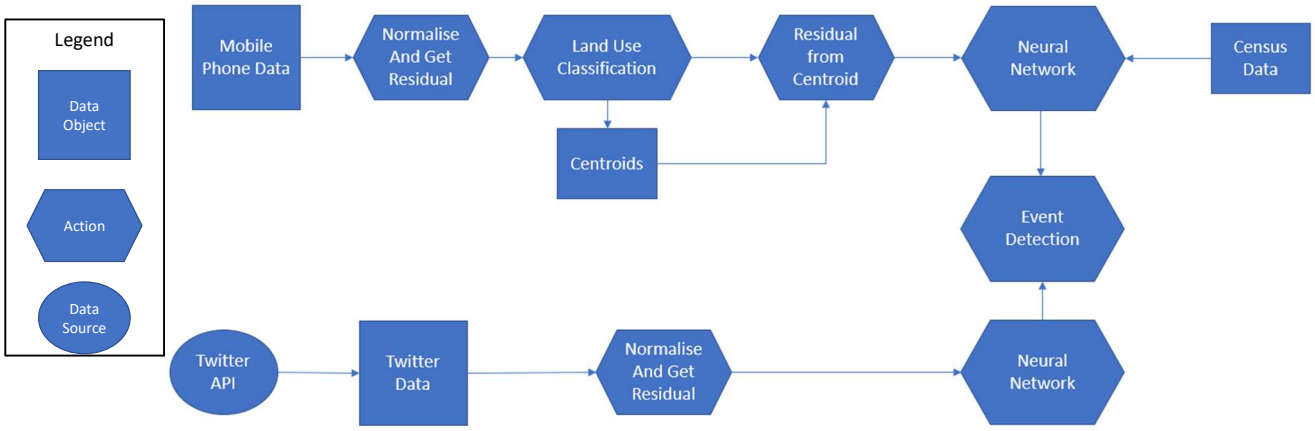


Figure 1: Diagram illustrating the overall architecture of the system.

Each track has a distinct purpose. The key purpose of the mobile phone data track is to identify the locality of events and the key purpose of the Twitter track is to provide causality of an event. Both tracks provide temporality of events which will be used to correlate the events that they detect. Furthermore, having two independent neural networks enables some filtering of uncorrelated events which would increase the overall reliability of the approach because fewer false positives would be produced. In the context of the motivation of the approach, which is to enable an automated response to social events, it is preferable to have false negatives over false positives due to a response likely requiring lots of resources and therefore a false positive could be inefficient because of a waste of resources. It is assumed that a false negative would eventually be noticed by a human in any case, be it an operator seeing that the mobile phone network is at high capacity or a police unit calling for support.

3.1. Mobile Phone Data

The primary goal of the usage of mobile phone data is to provide locality for events. The first stage in using the mobile phone data is the loading of the data which has been sourced from Telefonica Italy via [12] and which is delivered in a whitespace-separated text file format containing the activity of 10,000 cells each measuring 250 metres by 250 metres. The mobile data was recorded from the 1st of November 2013 to the 1st of January 2014, with observations recorded every ten minutes. Each ten-minute period is segregated into the activity of different mobile phone country codes. To save time when loading the files and since separate values for each country code do not provide any benefit, the country codes are merged. For the scope of this implementation of the approach, it is beneficial to have an hourly resolution as this will

prevent overfitting in later stages of the pipeline. For this reason, the data is converted to an hourly resolution and saved to a CSV file so that this step does not have to be repeated.

Normalisation of the data is performed using a Z-Score approach. This normalisation technique has been chosen as it has been demonstrated to be effective for land use classification [13]. Furthermore, alternative normalisation techniques such as minmax normalisation can cause “normal” points to have very small separation if there are anomalous data points, and this can complicate the identification of the trends underpinning “normal” data. As Z-Score normalisation is less susceptible to this problem and it is expected that anomalous data points will be present as they may represent social events, this normalisation method was chosen. The Z-Score normalised average pattern for each cell is saved into a JSON file. This average pattern is split into the separate weekend and weekday patterns, since for certain land uses the activity on the weekend is expected to deviate from the activity on a weekday [7].

Land use classification will be determined for each cell using K-Means derived centroids. The choice of this method will be outlined in the development section. The centroids are stored in a similar way to the cells in a JSON file with separate weekday and weekend activity patterns.

Residual activity from the centroids’ overall activity will be determined. This permits the identification of times where activity levels are much higher than expected which would indicate that a social event is occurring. Since the resulting data is likely to be transferred directly into a neural network, it is not stored anywhere.

3.2. Census Data

The usage of census data was not included in the original specification but was justified by early implementations of the mobile phone data event detection seeming to detect the crowdedness of a cell, rather than providing the location of an event. In essence, this means that if a cell was busy, such as those in the city centre, the network would produce a false positive. Consequently, the decision was made to include the number of residential buildings, number of non-residential buildings and the total population of the cell in the inputs of the neural network. The inclusion of the population of an area assumes that a higher population would indicate a higher likelihood that an area is going to be crowded, which would also hold for a higher number of residential buildings. The number of non-residential buildings was included as a higher number of these may indicate an increased prevalence of shops or other commercial buildings which may create crowded areas. The source for the data was Istat, which is the Italian government statistics agency.

The source data is delivered in a GeoJSON file and is not mapped to the same geographical layout as the mobile phone data. As a result, it is necessary to calculate the relevant values for the chosen features for each mobile cell. This was done by including the values for each census zone that was either completely or partially located within a cell or the zone fully encapsulating a cell. The inclusion of full encapsulation contributing to the value was selected as the only zones that were large enough to contain an entire cell are those that cover agricultural areas, and the number of those zones is likely to be extremely low anyway. Once the values for each cell were calculated, the data was Z-Score normalised and the result was saved to a CSV file.

Processing the census data is a prerequisite to training the neural network for the mobile phone data which is done by providing some sample cells and days on which events occur.

3.3. Twitter Data

The purpose of the social media data is to provide causality for events, and the data is sourced from Twitter. Twitter was chosen as a data source as it provides an API to search historical data for research purposes which other text-based social media platforms do not offer. The Tweets used will all be from any IP address that Twitter has determined as being registered within 12km of the centre of Milan and created

in the same period as the mobile phone data. First, the Tweets must be obtained using a GET request on the RESTful Twitter API [14]. The API provides two types of data: the number of Tweets over a period and the Tweets and their contents over the same period. For this approach, the content of the Tweets is required so that causality for the events can be obtained. The downloaded Tweets are then stored in a JSON file, with each day's worth of Tweets in a separate file. Having multiple files protects against file corruption or accidental deletion, which was necessary as Twitter limits the number of requests that can be made to its API. As a result, repeatedly downloading the Tweets could lead to this limit being reached and thus prevent progress in the development of the approach.

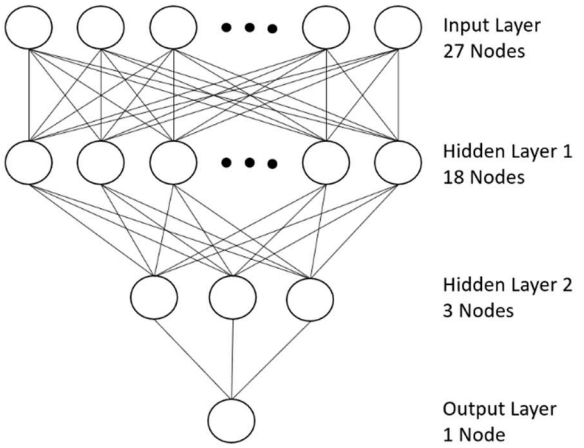
When a specific word's usage within the set of Tweets is required, the number of instances of that word is obtained by searching the JSON files, with the resolution of the usage being daily. A daily resolution is chosen as this was deemed to produce the highest variance between days with social events and those without while limiting the compromise needed in the precision with which events can be detected. Once the usage counts are finalised, the number of Tweets per day is counted and both the Tweet count and word usage are Z-Score normalised. The justification of Z-Score normalisation is the same as with the mobile phone data. After the Z-Score has been calculated, the residual activity is calculated using the two Twitter data sets to make it possible to identify higher word usage than normal.

The detection of events is carried out by using the activity associated with a specific set of words as inputs to the neural network. The training for this network is performed by giving the network the same set of words as will be used to detect the events and some training outputs. After the network is trained, the activity level for each day of the data's timeframe is fed to the network to identify the days on which events occurred. While it may seem that only giving a specific set of words could be problematic, in practice to detect unplanned events this list of words could include phrases such as "protest" or "bomb threat" that may indicate those causes of an event. With more exploration over time, it would be possible to build up a comprehensive list of words that could reflect social events of interest.

3.4. Neural Network Design

The original approach was to use a filtering algorithm to detect abnormal activity levels in the data which would indicate events taking place. However, during the delivery of the implementation, it was discovered that using a filter was ineffective, as determining parameters for the threshold of a filter would be difficult. Given this constraint, it was decided to use neural networks due to their ability to adapt event detection to a set of data through training.

The neural network for the mobile phone data will have 27 input nodes. One node for the value of each hour of the day of the input and one node for each of the three parts of the census data. There will be two hidden layers because it is believed that a neural network can model any arbitrary classifier with two hidden layers [15]. A first hidden layer will have 18 nodes as previous research has indicated that having two-thirds of the input nodes produces the most effective results [16]. A second hidden layer will have three nodes. The output layer only has a single node to make it easy to identify whether there is an event or not. This layout is visualised in figure 2 below.



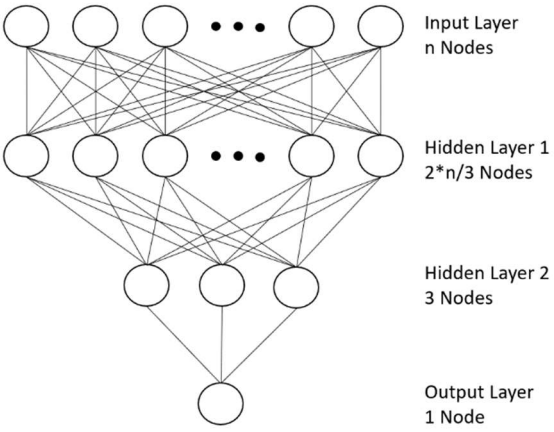


Figure 2 (Left): Layout of the mobile phone data neural network.
Figure 3 (Right): Layout of the Twitter neural network.

In the Twitter neural network, it was not possible to predetermine the number of input nodes before the initial data analysis. Therefore, the number of input nodes is represented as a variable number of nodes in figure 3 above, dependent on the number of words in the word list. There are also two hidden layers, using the same reasoning as with the mobile phone data neural network, the first hidden layer will have two-thirds of the input nodes. In the case where this value is fractional, the decimal value will be truncated as opposed to rounded as this is more efficient computationally, which may have an effect if implemented in an automated system handling large amounts of data. The number of nodes in the second layer is kept constant at three as changing this will not have a material impact on the outcome. Once again, there will be a single output layer.

The final stage which combines the two tracks is the event detection stage. Here the implementation checks whether events were detected on a given day for a given cell on both the mobile network and Twitter tracks. If this is true, then it indicates that there is an event. The data for detected events is saved as a serialised Python dictionary, with the keys being the cell ID and the value being a list of the days on which events were detected. A serialised dictionary was used here as the results are going to be plotted immediately and therefore do not need to be human-readable.

4. Development

After having outlined the design of the approach, the development process is described. The entire development process took place in Python as there is a vast range of guides, libraries, and tools available to make the development of the implementation of the approach much easier. The first step of development was to set up a virtual environment, which was used instead of installing modules globally. The use of a virtual environment means that the implementation is highly portable as the virtual environment can easily be moved across devices, unlike a global Python module installation. Most of the required modules could be installed automatically using pip. However, there were issues installing three dependencies of the GeoPandas module, requiring the manual installation of the relevant Python wheels. As mentioned in the specification, the architecture was inspired by the microservice architecture, which is represented by saving the processed data to a file when moving between stages so that the implementation of the stage can be changed as long as the file structure remains the same. Persisted data was saved in three file types: JSON, CSV or pickle serialised. These three types were chosen as there are already Python packages available handling the input and output of these files. JSON and CSV files were used where it may have been useful for the transferred data to be human-readable, whereas serialised pickle files were used when the contents of the file needed to be easily loaded into a Python object. As this implementation was a proof of concept,

it did not follow any particular development methodology as targets may change frequently. However, if one had to be selected, Agile would be most representative as the development took place incrementally, with each step of the pipeline being developed over a sprint, with the implementation evolving over a sprint. The code was stored and backed up in a GitHub repository to ensure that any changes made could be reverted, if necessary, which was a crucial part of the development process.

4.1. Initial Data Analysis

Several initial data analyses were performed to evaluate feasibility before continuing with the systematic development of the approach.

The first area to be analysed was the viability of detecting events using the mobile phone data. The initial results were promising, showing large spikes from the residual centroid activity, as shown in figure 4 below. The highlighted areas are the periods when football matches took place. Unsurprisingly the largest spike occured in the final highlighted area, which is when the Milan derby was played and which was expected to have the highest level of activity. However, some unexplained minor spikes can also be seen. These could be caused by non-football events such as corporate events in the stadium, as the mobile cells containing the San Siro contain very little else and therefore could easily be impacted even by a slight change in activity.

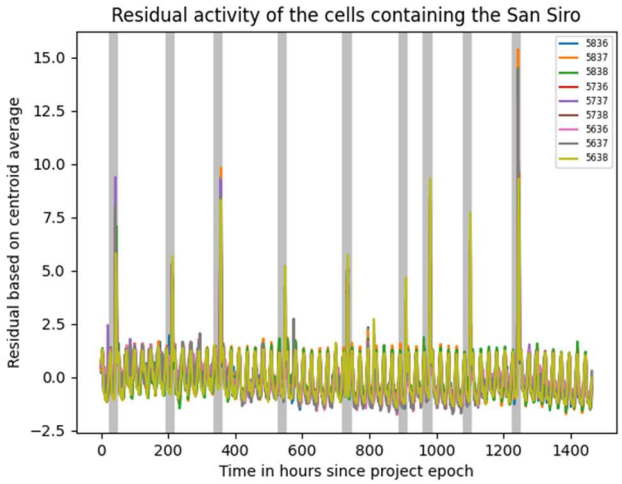


Figure 4: Centroid residual activity of the cells containing the San Siro stadium. The legend identifies the cell IDs depicted in the graph.

However, when focusing on smaller venues such as the Alcatraz club, the spikes in activity are not as easy to interpret, as shown in figure 5. This presents a problem, as it suggests there is a bound on the minimum size of the events that can be detected. Since the Alcatraz has a capacity of only 3,000 people [17], this would represent a significant restriction on the minimum size of events that can be detected in an urban area. Consequently, as all football matches were detectable, for an event to be detectable its attendance may need to be in the 10,000s. The actual threshold and driving factors would need to be determined in later work. It does not rule out that small events would be detected in rural areas, as it is likely that normally there will be a low level of activity and therefore a social event would still cause a significant increase in load in that area. It should be noted that the activity level is a lot lower in December than in November in the cells containing the Alcatraz club when interpreting the results.

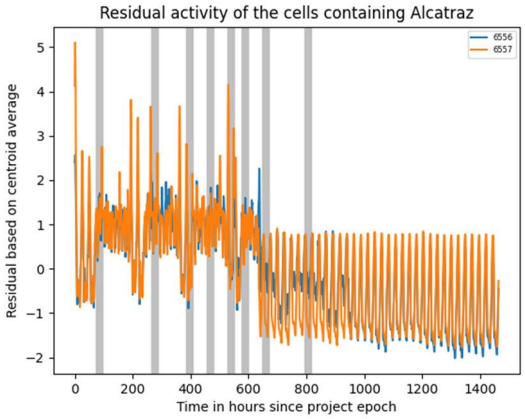


Figure 5: Centroid residual activity of the cells containing the Alcatraz venue.

The Twitter data is less sensitive to event attendance size. This is evidenced in the figures below for the word usage of “pixies”, “bastille” and “arctic monkeys” all of which are bands that played in Milan during the data’s timeframe. In the graphs, a dashed line represents the day on which the respective event occurs.

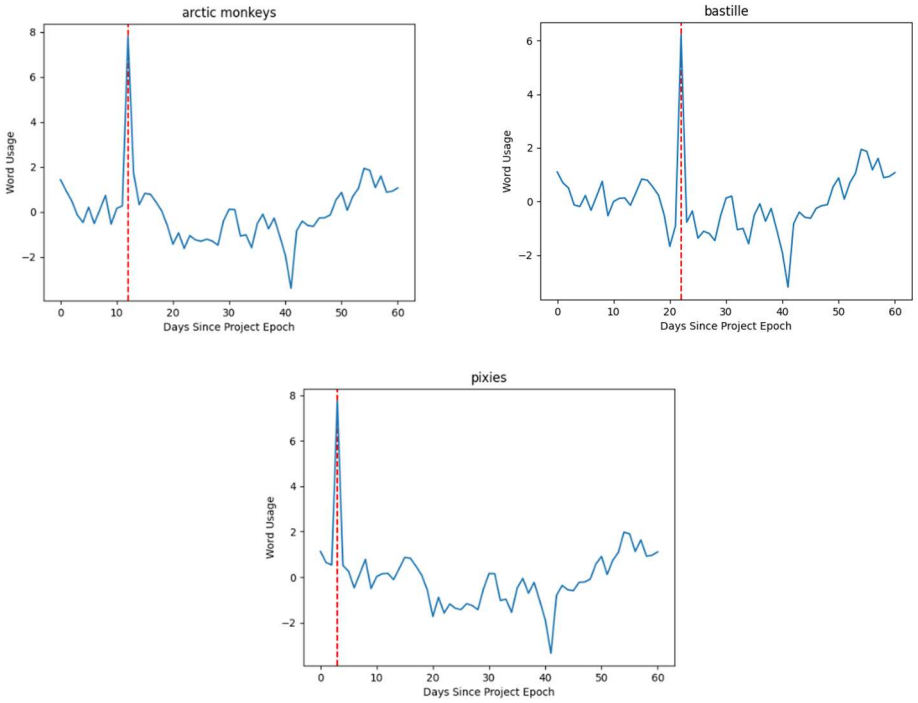
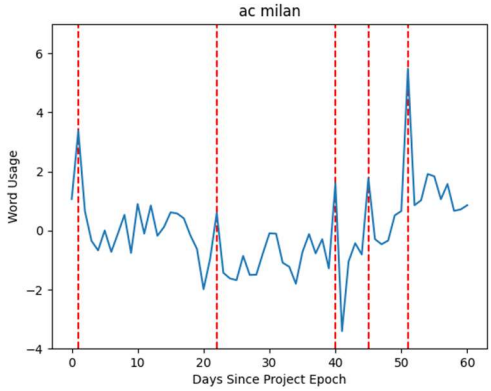


Figure 6 (Top Left): Residual word usage of ‘arctic monkeys’.
Figure 7 (Top Right): Residual word usage of ‘bastille’.
Figure 8 (Bottom): Residual word usage of ‘pixies’.

As the tests for the proposed approach focus on football matches, it is important to compare the social media interactions of the two football clubs in Milan so that the impact any difference may have on the results can be assessed. While it could be argued that the interactions on most match days are not comparable due to there being different opponents (AC Milan played Ajax in the Champions League and Roma in the cup, whereas Inter did not have any such notable opponents) the Milan derby took place during the time frame of the data. The word usage of Inter had a residual Z-Score of slightly below 6 on that day, whereas for AC Milan it was significantly lower than 6 (see figures 9 and 10 below). There are a few potential reasons for the higher number of Inter related interactions on Twitter. One of the reasons may be that the fanbase of Inter may be younger, Inter may simply have more fans or the cause is that Inter was

the designated home team for this fixture and therefore had a larger allocation of tickets than AC Milan. As the number of interactions on social media is representative of crowd size, Inter being designated the home team may be the most likely cause [18].



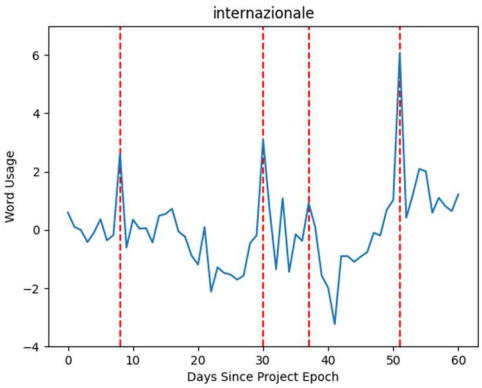


Figure 9 (Left): Residual word usage of 'ac milan'.
Figure 10 (Right): Residual word usage of 'internazionale'.

4.2. Neural Networks

Initially, it was believed that an existing off-the-shelf package would be used to implement the neural networks, however, as the implementation of the approach developed, uncertainty in the structure of the neural networks, as well as features such as the activation function, arose. Therefore, it was decided that the flexibility of a custom framework was required, and so one was created with help from a guide found online [19]. In the implementation, all of the associated functions and classes are stored in a single Python module containing four classes: 'layer', 'fclayer', 'activation' and 'network'. The first class, 'layer', is the base class which each layer extends from, with 'fclayer' and 'activation' layers containing the required methods for implementing a fully connected layer and activation layer respectively. The final class represents a complete network and contains the methods required for adding layers, training the network, and predicting outcomes. The activation function used was the sigmoid function, with the predominant reason for choosing this function being that it is well suited for classification problems. The main reason for this is that the increase in input only has a specific effect on output over a certain range of values, outside this range the effect is minimal. Additionally, the key issue with the sigmoid function, namely the vanishing gradient problem, will have a minimal effect when it is used here since the implementation contains only two hidden layers.

4.3. Census Data

The pre-processing required for the census data was performed from the file "format_census.py" using the "format" and "zscore" functions. The "format" function calculates the values from the required fields for each cell and saves the result to a CSV file. Detecting whether a census zone applies to a given cell was performed by converting the cell to a shapely object and then using the inbuilt methods of GeoPandas to select if the zone was applicable. The "zscore" function opens the CSV file generated by "format". In the next step Z-Score normalises the values using pandas built-in functions and then saves the result to another CSV file.

4.4. Mobile Phone Data

Operations on the mobile phone data are separated into two main stages. The first stage is preparing and using the data for the land use classification, the second is preparing and using the data in the neural network. Although the results from the first stage are used in the second stage, the second stage can be run multiple times without requiring a rerun of the first stage.

4.4.1. Opening the Data

The mobile phone data is imported and processed with help of the pandas module which contains all the necessary tools that are needed for this stage of the process. Development of the functions which open the CSV files took place in a Python module named “open_cdr.py”. This file contains three key functions, the first of which is “open_cdr”, that takes a date as a parameter and opens the relevant file of the mobile phone network data. The second function is “merge_countries”, which gets the mobile phone network data and merges the country codes in that file. The last function is “merge_all”, which opens each file of data, merges the country codes and concatenates all of the DataFrames into a single DataFrame.

4.4.2. Normalisation and Land Use Classification

All the steps required for the land use classification phase are performed using the functions in the “land_use_classification.py” file. The first step is to format the data so that the activity on each day of the week is totalled, this is performed in the function “parse_data”. The results of this function are saved to a CSV file that can easily be opened with “open_data”. Once this file is created, each day’s activity is totalled and the data is split into two groups, one for weekends and one for weekdays using the “sort_daytype” function. From these groups, the activity patterns are normalised using inbuilt pandas functions. The data for each cell is converted to an instance of the cell class to represent a cell using the “format_data_all” function. The cell class contains the average normalised activity for each hour of the day on weekdays and weekends separately. Once this step has been performed, the mean activity on weekdays and weekends across all cells can be calculated which is used to calculate the residual for each cell. The minimum and maximum values for each hour of the day, after conversion to residual data, are then stored. Finally, the data is ready for land use classification. K-Means was the selected algorithm to identify the land use of each cell. Several factors led to the choice of using K-Means, the first of which was that an unsupervised clustering technique was required as there was no directly obtainable data for the land use in each cell. Secondly, as mentioned in the literature review, there was supporting evidence demonstrating that land use classification could be performed effectively using this algorithm [7], with five centroids producing the best results. Euclidean distance was used despite the high dimensionality. The centroids were represented in their own class, which contains many of the same attributes as the class for the cells, but also includes an attribute with a list of all the cells that have been assigned to that centroid. The calculated mean cell activity pattern and the centroid pattern are saved to a pickle file using the “save_cells” and “save_centroids” functions. While libraries are available which can provide implementations of K-Means, it was necessary to generate a custom implementation to create a seamless method of storing the result of the classifications in the chosen file type.

4.4.3. Centroid Residual

The calculation of the centroid residual is performed using the “load_nn_cells” function from the “dataneuralnetworks.py” file. This function prepares data points for input into the neural network. It takes a list of data points which contain a cell and the date requested. The function then loads the average pattern of each cell and centroid from the relevant JSON files as well as the census data for the requested cells. The activity of the cells on the requested days is loaded, normalised and stored as a NumPy array. The centroid to which the given cell is closest is then obtained using the “get_centroid_ind” function. In the next step, the centroid’s mean pattern is generated. The function then subtracts the average overall pattern as well as the centroid’s mean pattern from the desired cell’s pattern. Next, the resulting array and the census data is concatenated so that the data is ready for use by the neural network.

4.5.Twitter Data

The Tweets were obtained using the Twitter V2 search API [14]. As mentioned in the design section, this API uses a REST architecture, accepting GET requests. To access the API, an access token must be used to ensure security, which was stored in a JSON file and was loaded when required. For downloading Tweets

there are two functions, the first of which is “get_tweets” and the second is “get_tweets_next”. These two functions support Twitter’s page-based model for retrieving large numbers of Tweets. To obtain the first page, the “get_tweets” function is used, and subsequent pages are retrieved using “get_tweets_next” within a loop until all required Tweets have been collected. Page management was provided by saving the token for the next page locally so that the process could be resumed without requiring duplicate retrieval of any page. The Tweets were stored in one JSON file per day. Within these functions, the requests package is utilised to perform the HTTP request to Twitter, with a “query_builder” function building the GET query string which is part of the URL. The code terminates when the final page is reached and thus the “next page” token field does not exist.

Another feature of the code is that it checks if the HTTP response code from the API is 200, which indicates that the request is successful. If the HTTP response code is 428 (Twitter specifies in the documentation that this is the response code when the rate limit is exceeded), the code parses the amount of time that is required to wait before making another request and then waits for that period. This feature was incorporated as it allows the code to run unattended. Automated responses to problems such as rate limits are important as this supports reliability. Reliability is an important factor as this would allow other systems to be built around the approach to efficiently allocate resources. If an unexpected HTTP response code is received the process will terminate.

The first task in the development of the Twitter data is to identify a set of words to be used to identify social events. This was done by having a start set of words which were extremely likely to occur on the days events take place, and which for the football matches were “ac milan” and “internazionale”. The first step for this is to search the JSON files and save all the Tweets where these words appear into a list. This set of Tweets is then parsed into individual words based on white space as the separation character. From the words found, in the initial step all words beginning with “http” are removed, as these are likely to be hyperlinks and web crawling was not in the scope of this implementation. Stopwords were then removed using the NLTK package (stopwords are a list of words used commonly, but which do not provide much informational value in the context of text analysis such as articles or connectives). Additionally, words less than three letters long were also removed as these are likely to be words that have limited value for this implementation. Then words that appeared frequently in these Tweets were added to the list of words that events will be used to detect. It was discovered that the names of opposition teams may provide some informational value (these were mentioned in around 1790 Tweets). Once a list of words had been created it was hardcoded as a list of strings. When obtaining the activity of each word, the body is converted to a lowercase format to remove case sensitivity.

Once the words had been selected, the activity pattern was determined so that it could be provided to the neural network. This task was performed by the function “load_nn_tweets” which creates a list for each word with each element specifying the number of times that word is used on each day. The lists are then combined to form a nested list. The function then iterates over each nested list and normalises them individually, making use of some of NumPy's mathematical operators. After normalisation, the residual was calculated. Finally, the function selects the values for each nested list and outputs them.

4.6. Graphing

During the development, sets of functions were produced to help visualise the data and the outcomes of stages of the approach. However, these will not be described in detail as they have no impact on the performance of the approach. Two main libraries were used for graphing: the pyplot sub-library of matplotlib and kepler.gl. All graphs were saved as PNG files for graphs using pyplot and HTML files for those using kepler.gl, since these two file types can be viewed on most modern operating systems.

4.7. Training The Neural Networks

The mobile phone data neural network is created and trained using the "cell_network" function from the "dataneuralnetworks.py" file, which loads the activity for each day of one of the cells containing the San Siro stadium. Next, the neural network is initialised to the configuration shown in figure 2 and trained for 10,000 iterations with a learning rate of 0.1. The final network is subsequently saved to a pickle file using the function "save_network_cell".

Similarly, the Twitter neural network is created and trained using the function "tweet_network" from the same file. This function takes the word list as a parameter. The function loads the activity of the words on some predefined days. The network is then initialised to the configuration shown in figure 3. Again, the network is trained, but this time the number of iterations is 1,000 with the same learning rate. A lower number of iterations was chosen, as the training data sample size was smaller and therefore the chance of overfitting was much higher. Once again, the network is saved to a pickle file using the function "save_network_tweet".

The networks were saved to pickle files as opposed to other storage methods as it was believed that it would be difficult to store the weights of the layers effectively in a CSV or JSON file, whereas, with a serialized pickle file, very little effort was needed to save and load the networks.

4.8. Event Detection

The first step before performing the event detection is to load the trained networks using the functions "load_network_cell" and "load_network_twitter". The event detection takes place within the "detect_events_double" function of the "dataneuralnetworks.py" file, which takes the two neural networks and the word list as parameters and a parameter called "start_date". The "start_date" parameter is an integer representing the time in days since the data epoch to facilitate the resumption of processing when the execution of the function has been interrupted. The first step is to load the activity pattern of the words and normalise these patterns. Subsequently, the function iterates through each day of the time frame, loading the mobile phone data for all the cells on that day from the relevant CSV file. On each day the activity values of the words are loaded and passed into the Twitter neural network to detect any events. Additionally, for each day, the function then iterates through each cell, normalising the activity pattern and calculating the residual as was done in the "load_nn_cells" function. In the next step, the mobile phone data neural network then detects whether there is an event on that day and the result is saved. If both neural networks detect events, the day is added to the cell's event list.

5. Testing

This section will go into detail about how the code in the implementation was tested to demonstrate that it is working as expected, thus removing implementation errors and increasing confidence in the accuracy of the results that were produced.

5.1. Neural Network Framework

The framework for the neural network was tested in a file called "neural network test.py" by training a network to perform an XOR operation. XOR was chosen for the test as it is a simple logic operator that does not have a linear classification boundary and therefore is exactly the category of problem that a neural network is suited to solving. The test was performed by training a network with all four possible inputs for XOR and checking whether each output is the correct one for that input.

5.2. Opening Mobile Phone Data

Opening the mobile phone data and the merging of the separate country code values were tested by selecting a random set of cells from the files and checking if the merged results were the same when totalled manually. A similar operation was performed for testing the conversion of the data to an hourly

resolution. During this stage of testing, it was noticed that there was missing data for two of the cells, 5239 and 5339, for the entire month of December. This was kept in mind when moving on to the evaluation stage of the approach, but it had no material effect on the outcome. The calculation of the residual was tested similarly.

5.3. Twitter API

Assessing the ability to obtain Tweets from the Twitter API was done in three stages, the first being to identify whether the script was accessing the API correctly. This was achieved by performing some trial runs and monitoring the response codes provided by the API. The next stage was to identify whether the Tweets were being obtained in the format and volume expected, by checking whether the expected number of Tweets was being produced as indicated in the summary data and whether a random selection of them was in the correct language, proving that they were from the correct region. Additionally, the time stamps were monitored to ensure that the Tweets were being pulled from the correct time frame. The final stage was to monitor whether the Tweets were being saved correctly, by verifying whether all the Tweets in some test runs were in the saved files. A unit test framework was utilised with mock responses to test behaviour when the rate limit was reached or an invalid authentication token was used.

6. Evaluation of Final Product

After having established certainty that there are no implementation errors the effectiveness of the approach is assessed. For the purposes of evaluation the word list used consisted of “ac milan”, “inter”, the names of the teams they played against (see Appendix B), “san siro” and “giuseppe meazza”. These words were chosen as it was believed that their usage was representative of any ongoing footballing events, while not introducing too much noise.

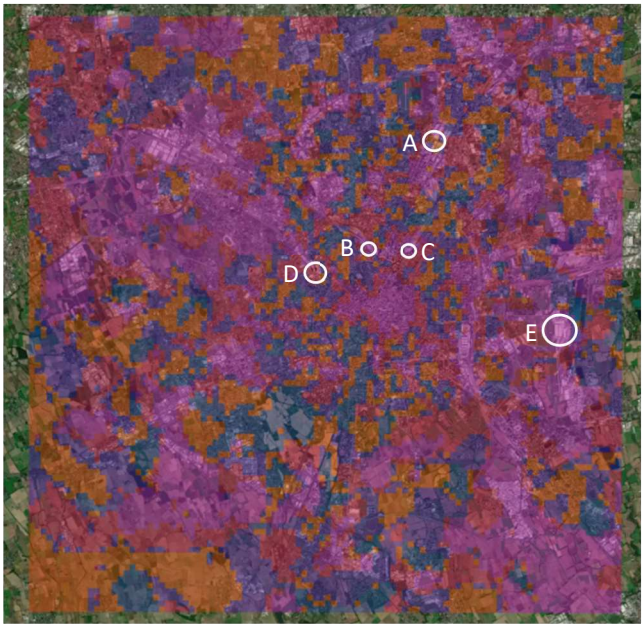
6.1. Land Use Classification

Firstly, the result of the land use classification will be explained. This consists of two parts, the first being the analysis of the map and the activity patterns, with the second being an insight based on Google Street View of what each land use may be. As per the literature [7], five clusters were used which classified cells as follows:

1. The first cluster is green on both charts. From the satellite view, this land-use appears to cover some fields, as well as some of the more built-up areas in the suburbs of the city. Given the high level of activity in the mornings, which then decreases and picks up again in the evening on weekdays, it is likely that the major use of this classification is residential. It is also possible that there may be some clubs and restaurants in this classification, producing the increased activity in the evening.
2. The second cluster is marked as purple on the charts. This land-use peaks during the evening and is less busy during the working day but has the least variance between the two time periods. It is therefore highly likely that these areas are a mix of many different land-uses. As it seems the cells that have been assigned to this land use are not completely inactive during the day, this classification probably contains a high proportion of shopping and commercial land use. This is expected to be the case as shops may open after most people get to their place of work and stay open until shortly after workers go home, enabling purchases on the commute home.
3. The third cluster is pink. Satellite imagery from Google Maps indicates that this land-use contains the city centre as well as a lot of the industrial parks. The indication from the activity pattern is that these areas are mostly places where people work, since on weekdays the activity is generally highest during working hours. Typically, offices are found in the centre of a city, as this allows for efficient access for the largest proportion of the workforce of a city. The inclusion of the city centre in this land use has probably influenced the activity pattern, as the city centre also contains most of Milan’s tourist

attractions, which are likely to be visited in similar amounts on weekends and weekdays. The spike on weekend afternoons indicates that there may be some leisure areas in this classification too.

4. The fourth cluster is red. This land-use forms a ring around the city centre and seems to cover a lot of the peripheral towns and villages of Milan. On weekdays, these areas appear to be busiest during working hours, which indicates a high likelihood that the dominant land use attracts people to work there.
5. The fifth cluster is orange. This land-use follows an activity pattern similar to cluster one, but with a slightly lower activity level on weekday evenings. This would imply that it is also residential land use, but it may contain some workplaces or shopping areas as well which may attract people during working hours. Interestingly, land use one, two, four and five all follow similar land uses at the weekend, with a dip in activity around midday.



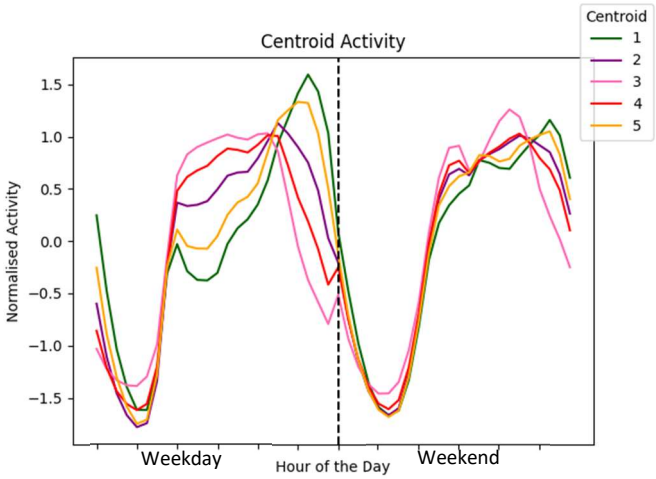


Figure 11 (Left): Map of the land use classification for Milan, with Biccoca shopping village (A), the two main train stations (B, C), the CityLife development (D) and Milan Linate Airport (E) marked.
Figure 12 (Right): The activity pattern of each centroid (non-residual). The dashed line separates weekdays (Left) and weekends (Right).

The analysis of the land use classification using Google Maps was performed by exploring the classification of several notable points of interest:

- Milan Porta Garibaldi and Milano Centrale Railway Stations (B and C on the map) are both in cluster 3 cells. This is not surprising as it is likely that the number of people travelling to and from the city will remain consistent during the day, with a slight increase during rush hours as people commute to and from the city.
- Milan Linate Airport (E) also falls under cluster 3. It is important to note that this is not the main airport for Milan, which is located outside the area included in the dataset.
- The CityLife development (D), is a major redevelopment project similar to London's Canary Wharf, which is mainly classified as land-use class four. Initially, this was surprising as it was believed this development would be a major office hub and thus the activity should reflect this, but further research showed that only some of the residential buildings had been completed during the data time frame, with the major office areas still being built. As this area was still under construction, it explains why this predominantly residential area did not have a peak in activity in the evenings, assuming the builders produced a significant amount of the mobile phone traffic in that area during the day.

- Biccoca Shopping Village (A) is a shopping centre classified as a mixture of clusters two and five. This makes sense since some parts of the shopping centre may become busier in the evening, with people visiting after work, but generally, the centre will be busy during the entire working day and on weekends.

Overall, the impact of these results was to give confidence in the land use classification component while allowing some insight into the structure of the city and where unexpected events may take place.

6.2. Mobile Phone Data Event Detection

The evaluation of the neural network will be performed by investigating the events that are detected by mobile phone data and their accuracy in two areas: the first containing the San Siro stadium, and the second containing the Fiera Milano exhibition centre. These two venues were selected as the San Siro stadium hosted events which had been discovered manually before the implementation of the project (See Appendix B) whereas the Fiera Milano exhibition centre did not.

Figure 13 demonstrates the events detected around the San Siro, where each dot in the mapping of the detected events represents a single event, with the height representing the number of detected events. The cells directly containing the stadium showed detection of all the events, with no false positives in this area. Some of the cells slightly further away from the actual stadium detected fewer events, for example, the two cells to the left of the stadium that only detected one event. This is not that surprising, as it is likely that the further a cell is from the event holding venue, the less of an impact it feels. The event that was most detectable by peripheral cells (which are around 500 metres away) was the Milan derby. It is not surprising that this was most detectable due to having the largest increase in activity at the San Siro, as already mentioned in the data analysis. The second most detected event across the cells in this area was the AC Milan vs Fiorentina match. While some of the cells were used in the training data, these figures demonstrate that the neural network is suitably able to detect an event and the radius of its impacts.

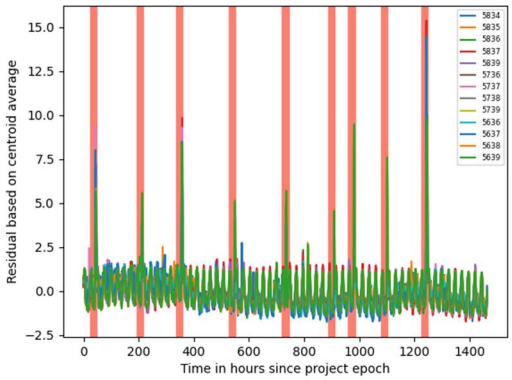


Figure 13 (Left): Events detected in the vicinity of the San Siro, with the stadium highlighted.
Figure 14 (Right): The activity pattern for the cells containing the stadium, with detected events highlighted.

The second area that was examined is the Fiera Milano exhibition centre, which was chosen as it was not included in the events listed before the model was created (Appendix B). However, during the data's time frame, it did hold some events. For example, between the 30th of November and the 12th of December the "Artigiano in Fiera" exhibition was held. This is reflected in the events detected in that area, with some cells detecting events on all days that this exhibition was held. However, in some cells, the exhibition was not detected on the 1st and 2nd of December. These two dates were a Sunday and a Monday, so it is possible that there was a lower attendance on these two dates for some reason such as the exhibition being closed. There are also events detected on the 29th of December, however, despite extensive research, it does not

appear that any events were held on that day. Furthermore, the activity pattern does not show a spike on that day either. There are also some days where it appears that there is a spike in activity on the activity pattern, but no event was detected on those days, particularly towards the early parts of the time frame. Further investigation could be made into these exceptions in further developments of the approach. Different cells seemed to have spikes on different days, which suggests that the venue can be split into different sections and thus host different events. Figure 16 demonstrates that some potential events which are represented by spikes are not detected, this is believed to mostly be due to a lack of training data. However, the fact that "Artigiano in Fiera" was mostly detected demonstrates that identification of the locality of unexpected events is possible.

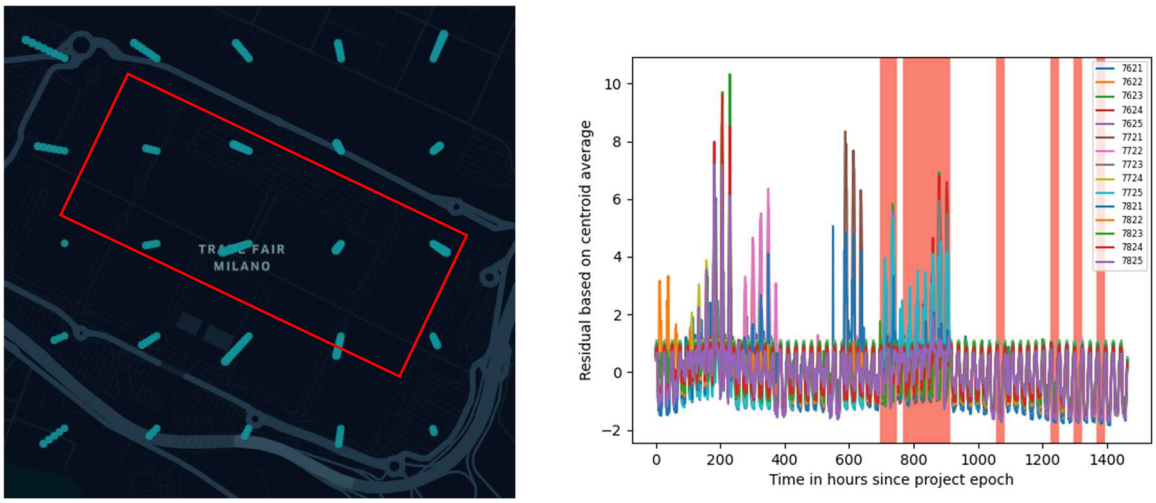


Figure 15 (Left): Events detected in the vicinity of the Fiera Milano with the exhibition centre highlighted. Figure 16 (Right): The activity pattern for the cells containing the exhibition centre, with the days detected events took place highlighted.

Overall, the lack of training data has placed some limitations on the effectiveness of detecting social events using only mobile phone data, especially as the events used to train the network are likely to have occurred at similar times of the day due to broadcast scheduling. However, ultimately the goal of achieving locality of detected social events was fulfilled and the approach was demonstrated to be able to measure the area of impact of an event as the footballing events were detected and a radius of their impact was estimated to be 500 metres. As previously mentioned, the project focuses on footballing events as these have the largest footprint in the mobile phone data, as well as having a significant footprint in the Tweets.

6.3. Twitter Data Event Detection

To evaluate the events detected from the Twitter data, a confusion matrix has been produced. The precision is 69.2% and the recall 90%., which gives an F1 score of 78.2%. This indicates that the neural network is effective in identifying social events from Tweets. The high recall shows that a high proportion of the events that occurred during the time frame of the project were detected. While this could be improved, it is still a good level of precision for the detection of events at this stage, demonstrating that there are not too many false positives as is further evidenced by a false positive rate of only 7.84%.

		Actual	
		Positive	Negative
Predicted	Positive	9	4
	Negative	1	47

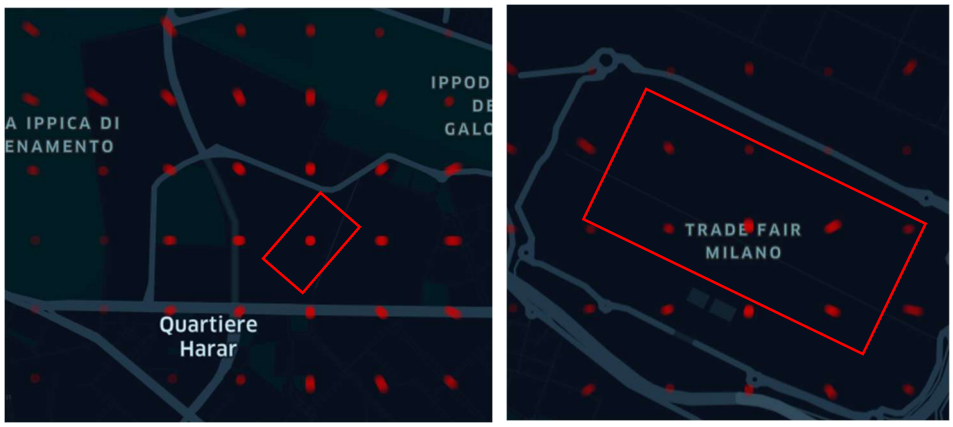
Figure 17: Confusion matrix For the Twitter detected events.

The results provide sufficient evidence to demonstrate that the techniques used for detecting social events from social media are adequate for a reliable automated system. However, if more training data were to become available, the effectiveness would probably be improved in a similar way to the mobile phone network events.

6.4. Combined Event Detection

The expectation is that the combined event detection identifies events which are likely to be caused by football, with some other events that happened on the same days also being detected.

In the region containing the San Siro stadium, all the football-related events were detected, with no false positives and with a similar density of detected events as identified by the mobile phone detection step. It also appears that the combined event detection does not lose the ability to measure how widespread the effects of an event are. As the word list mainly focuses on football, the system can successfully detect events which are related to football. Conversely, the exhibition centre has some of the non-football events detected removed and therefore the detected events are less dense than those purely from the mobile phone detection, which is to be expected given that the events held there are unlikely to be related to football. Still, some events were detected which may be filtered out if there were to be a switch to an hourly resolution of the Twitter data. This demonstrates that the approach successfully creates a link between causality and locality. Overall, key evidence in the results shown here that this approach is successful. Once again in the figures below, each red point indicates a single event, with the intensity and height of each stack showing the number of events in the figures.



Figures 18 San Siro (Left), Figure 19 Fiera Milano (Right): Detected events after the combination of the mobile phone and social media detected events. The regions are the San Siro stadium and the Fiera Milano exhibition centre respectively.

7. Critical Assessment of Project as a Whole

Overall, I believe the project can be deemed a success. The key stages of the project have been designed and developed effectively in accordance with the specification. Although some issues have been discovered in the results, especially concerning the mobile phone data, these do not negate the fact that the approach has achieved its main aim: to be a proof of concept that mobile phone and social media data can be used in tandem to detect social events.

7.1. Functionality

The final implementation can obtain and access both types of data, successfully retrieving data from the Twitter API and opening the mobile phone data. The Twitter word usage can be obtained, normalised and the residual found. Similarly, any given mobile cell's daily activity can be normalised and prepared for further processing. The final product successfully incorporates a method of land use classification. An

automated approach for detecting social events has been implemented using neural networks to detect events from the two sets of data, along with a method of correlating the events detected.

As required in the specification there is a high degree of portability, as all the plots are in a format that can be widely accessed such as HTML or PNG files. This was also enabled by using virtual environments which allow the code to be run across different environments without much configuration being required.

7.2. Approach To the Problem

I deem the overall approach to the problem to be successful as all the objectives laid out in the specification have been achieved, all be it with varying degrees of success. The concept successfully implements ideas that have been suggested in previous literature, and the approach for the selection of correct implementation methods was satisfactory. Leveraging supervised and unsupervised machine learning techniques for different stages proved successful as it demonstrated this method for detecting social events and identification of causality and locality was effective.

However, there were some shortcomings in the approach of the project. The main one was the initial underestimation of the minimum size of event that was expected to be visible from the mobile phone data activity. This was a large issue as it drastically reduced the events that could be used to experiment with and train the model. However, this could be solved in the future as more data becomes available. Furthermore, the lack of prior research in this field made it extremely difficult to draw on previous experience. Therefore, the concept had to rely on bringing together abstractions of ideas from other fields such as land use classification and adapting them for use in this context. There was very little information which would set the expectation of the accuracy of the model and what the outcome of the experiments of the approach would be.

7.3. Process

I believe that the design and development of the approach were successful. Taking inspiration from a microservice style architecture proved extremely useful as it enabled a splitting of the approach into two main tracks. This allowed abstraction during development which was beneficial as I could focus on researching and developing a given stage without impacting other stages as long as the output remained the same. Technically, this could have meant the development of the approach in a completely random order as long as the inputs and outputs were defined, however, development followed the sequence of each track to maintain an overview of the implementation and limited the amount of test data needing to be created. The design was successfully developed, with the prior research aiding the final implementation, as exemplified by the successful use of land use classification data which has been discussed in the evaluation section. The aim of producing a highly portable implementation was achieved by using Python, virtual environments, and file types which can be opened in many different programs. The flexibility of custom implementations of both K-Means and neural networks proved to be beneficial as it improved the performance of the implementation of the approach.

The use of a Gantt chart successfully provided an overview of progress through the development, while the Kanban board maximised efficiency during a single stage of development of the implementation. While making use of the full Agile methodology was attempted, performing development on my own prevented the use of the key features, such as daily stand-ups.

7.4. Limitations

As mentioned in section 6, there are some limitations of the final model. It fails to detect events on certain days, but also produces false positives on other days. However, this is to be expected, as no model will be able to predict events with 100% accuracy. Nevertheless, the number of false positives is slightly higher than anticipated, which is expected to be largely linked to the lack of training data. If more data were available, the model would probably perform much better.

A less significant limitation is that the land use classification step can be computationally expensive. This could pose a problem if the data set is continuously changing, however, it is believed that in the real world the makeup of a city does not change rapidly, and therefore the land use classification could be run infrequently, perhaps even requiring to be only run once a year or less.

8. Future Work and Conclusion

Having evaluated the results of the approach, which was mostly a proof of concept as opposed to a fully productionised system, there are many directions this work could be taken in the future:

1. To apply the approach to some sort of real-time data stream. This would further develop the work performed in this approach in the direction of the original motivation as a real-time social event detection method that would allow for the real-time allocation of resources in response to social events. This could involve utilising recurrent neural networks. Recurrent neural networks take information from the previous time step's detection to influence the next time steps prediction. This would help with areas that are busy temporarily for some reason such as additional traffic due to road closures or similar.
2. The performance of the approach should also be investigated with a higher volume of training data and more varied training data. Which events can be detected may vary across different cultures and nations, and this should be evaluated. Furthermore, this implementation could only be tested on events in a relatively brief 61-day time frame.
3. It may be beneficial to assess the impact of alternative land use classification techniques on the ability to detect events. Using supervised versus unsupervised machine learning techniques should be assessed to determine whether these would facilitate the detection of different events.
4. Investigation of the threshold at which events can be detected as the initial data analysis demonstrated there must be one, but it was not investigated where this boundary may lie or what drives it.
5. To attempt to automatically assign tags to the events detected in a cell. An example of this would be all the events in the San Siro being labelled with the teams that are playing. The key hurdle to overcome this issue would be the need to switch the Twitter data processing to an hourly resolution, as several events may occur on the same day and thus a greater resolution may be needed to allow differentiation between the different events.
6. To explore the use of more auxiliary data to detect events. While this approach used some auxiliary data to help in the detection of events in the form of census data, it may be feasible that other data streams could aid the detection of events. For example, the number of people entering and exiting a public transport network could show the occurrence of a social event as there may be more people in the public transport network just before a football match or concert than expected. The consequence of adding more data streams may be increased certainty with which events are detected.

In conclusion, the concept presented in this report demonstrates that locality using mobile phone data and causality from Tweets can be used in the detection of social events, providing evidence that the three main hypotheses hold. There are several directions that this approach could move in the future as described above that would prove to be an exciting challenge for those who choose to pursue this work further.

9. References

[1] F. Richter, "Infographic: World Cup Final Sparks 32 Million Tweets," *Statista Infographics*. Statista, Jul. 2014. [Online]. (2021, Nov. 11). Available: https://www.statista.com/chart/2454/most-tweeted-world-cup-matches/.

[2] Z. Pi and F. Khan, “System design and network architecture for a millimeter-wave mobile broadband (MMB) system,” in *34th IEEE Sarnoff Symposium*, 2011, pp. 1–6. doi: 10.1109/SARNOF.2011.5876444.

[3] S. A. Munir, B. Ren, W. Jiao, B. Wang, D. Xie, and J. Ma, “Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing,” in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW’07)*, 2007, vol. 2, pp. 113–120. doi: 10.1109/AINAW.2007.257.

[4] T. Pei, S. Sobolevsky, C. Ratti, S.-L. Shaw, T. Li, and C. Zhou, “A new insight into land use classification based on aggregated mobile phone data,” *International Journal of Geographical Information Science*, vol. 28, no. 9, pp. 1988–2007, 2014.

[5] X. Wang *et al.*, “Spatio-temporal analysis and prediction of cellular traffic in metropolis,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2190–2202, 2018.

[6] F. Girardin, A. Vaccari, A. Gerber, A. Biderman, and C. Ratti, “Towards estimating the presence of visitors from the aggregate mobile phone network activity they generate,” 2009.

[7] V. Soto and E. Frías-Martínez, “Automated land use identification using cell-phone records,” in *Proceedings of the 3rd ACM international workshop on MobiArch*, 2011, pp. 17–22.

[8] G. Sagl, E. Delmelle, and E. Delmelle, “Mapping collective human activity in an urban environment based on mobile phone data,” *Cartography and Geographic Information Science*, vol. 41, no. 3, pp. 272–285, 2014.

[9] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, “Tedas: A twitter-based event detection and analysis system,” in *2012 IEEE 28th International Conference on Data Engineering*, 2012, pp. 1273–1276.

[10] G. Gautam and D. Yadav, “Sentiment analysis of twitter data using machine learning approaches and semantic analysis,” in *2014 Seventh International Conference on Contemporary Computing (IC3)*, 2014, pp. 437–442.

[11] “Structure - San Siro Stadium.” [Online]. (2022, Apr. 4). Available: https://www.sansirostadium.com/en/stadium/Structure.

[12] G. Barlacchi *et al.*, “A multi-source dataset of urban life in the city of Milan and the Province of Trentino,” *Scientific Data*, vol. 2, no. 1, p. 150055, 2015, doi: 10.1038/sdata.2015.55.

[13] J. L. Toole, M. Ulm, M. C. González, and D. Bauer, “Inferring Land Use from Mobile Phone Activity,” in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, 2012, pp. 1–8. doi: 10.1145/2346496.2346498.

[14] “Tweets lookup introduction | docs | twitter developer platform,” *Twitter*. (2022 Apr. 21). https://developer.twitter.com/en/docs/twitter-api/tweets/lookup/introduction.

[15] J. Heaton, *Introduction to neural networks with Java*, 2nd ed. Heaton Research, 2009.

[16] G. Panchal, A. Ganatra, Y. P. Kosta, and D. Panchal, “Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers,” *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, pp. 332–337, 2011.

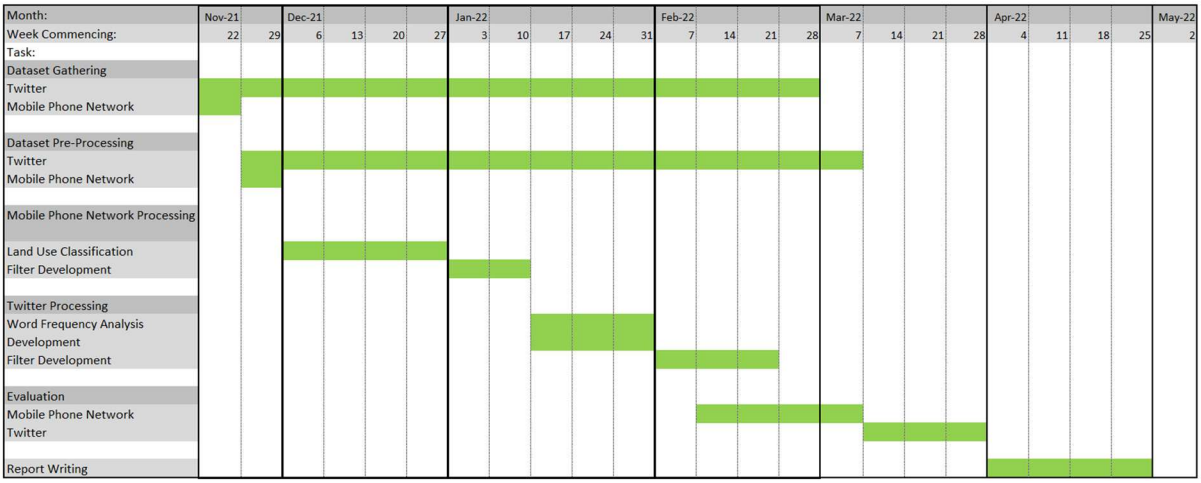
[17] “Alcatraz, Milan,” *cityseeker*. [Online]. (2022, Apr. 5). Available: https://cityseeker.com/milan/20092-alcatraz#:~:text=With%20a%20capacity%20of%20about,ambiance%20and%20well%2Dfurnished%20spaces.&text=%22Musical%20Times%22-,Alcatraz%20is%20one%20of%20the%20most%20popular%20and%20well,event%20venues%20in%20the%20city.

[18] F. Botta, H. S. Moat, and T. Preis, “Quantifying crowd size with mobile phone and Twitter data,” *R Soc Open Sci*, vol. 2, no. 5, p. 150162, 2015.

[19] O. Aflak, “Neural network from scratch in Python,” *Medium*. Towards Data Science, May 2021. [Online]. Available: https://towardsdatascience.com/math-neural-network-from-scratch-in-python-d6da9f29ce65

10. Appendices

A. GANTT CHART FOR THE PROJECT



B. LIST OF EVENTS IN MILAN

Name	Venue	Date
Bob Dylan	Teatro Degli Arcimboldi	03/11/2013
Pixies	Alcatraz	04/11/2013
Bob Dylan	Teatro Degli Arcimboldi	05/11/2013
Frank Sent Us	Centro Sociale Cantiere	09/11/2013
Fabri Fibra	Alcatraz	12/11/2013
Arctic Monkeys	Mediolanum Forum	13/11/2013
The Naked and Famous	Circolo Magnolia	16/11/2013
Buckcherry	Alcatraz	17/11/2013
Max 20 Live Tour	Mediolanum Forum	18/11/2013
Tom Odell	The Factory	19/11/2013
Skrillex	Magazzini Generali	19/11/2013
John Newman	La Salumeria Della Musica	20/11/2013
Primal Scream	Alcatraz	20/11/2013
Bastille	Alcatraz	23/11/2013
Bring Me The Horizon	Alcatraz	25/11/2013
The Wonder Years	Tunnel Club	26/11/2013
Nick Cave and The Bad Seeds	Alcatraz	28/11/2013
Negrita	Teatro Nuovo	02/12/2013
Jake Bugg	Alcatraz	04/12/2013
Kodaline	Tunnel Club	07/12/2013
The Fratellis	Magazzini Generali	08/12/2013
Hanson	Magazzini Generali	17/12/2013
Pusha T	Limelight	19/12/2013
Italy vs Germany	San Siro	15/11/2013
Inter vs Livorno	San Siro	09/11/2013
Inter vs Sampdoria	San Siro	01/12/2013
Inter vs Parma	San Siro	08/12/2013
Inter vs Milan	San Siro	22/12/2013
Milan vs Genoa	San Siro	23/11/2013
Milan vs Fiorentina	San Siro	02/11/2013
Milan vs Roma	San Siro	16/12/2013
Milan vs Ajax	San Siro	11/12/2013