FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Hugo Bronze Canelas de Brito Prata

INTELLIGENT DATA CONSOLIDATION METHODS FOR CROWD-SOURCED CONTRIBUTIONS

A contribution to FireLoc

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Prof. Alberto Cardoso and Prof. Jacinto Estima, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

January 2023

Abstract

It is of the most importance to tackle a Natural Disaster as early as possible, so as to mitigate its effects to the fullest. While preparation and prevention are important steps in dealing with Natural Disasters, the speed at which an organisation can respond to fluid events can also greatly influence the outcome of such disasters. Thanks to recent developments in automation, Artificial Intelligence (AI), and the widespread access to mobile technology, it is our belief that it is possible to leverage this new technological reality to contribute to the fight against Natural Disasters.

The present work aims to develop a smart system capable of refining individual events into clusters made up of related members, allowing its geo-location and the tracking of its evolution over time. To reach this goal, different techniques will be researched and subsequently chosen by taking into account their performance regarding the inputs that are expected to be necessary in the context of this project. This includes the research of several clustering techniques, novel and standard, as well as ways of dealing with inflows of heterogeneous data from multiple sources, such as Data Fusion. Finally, ways of mapping this information through a graphical interface were also researched.

This project aims to be a contribution to the FireLoc project, a a system designed for the identification, positioning and monitoring of forest fires with the aid of crowd-sourced data, while also attempting to become a general approach to dealing with Natural Disasters, by providing a smart system capable of monitoring inflows of information-rich geo-located events over time.

Keywords

FireLoc, Disaster Detection and Monitoring, Geo-location, Data Fusion, Smart Systems, Clustering

Contents

Acronyms

AI Artificial Intelligence.

BN Bayesian Networks.

CNN Convolutional Neural Networks.

DBSCAN Density-based Spatial Clustering of Applications with Noise.

FCT Foundation for Science and Technology.

GPS Global Positioning System.

GUI Graphical User Interface.

INES-C Institute for Systems Engineering and Computers at Coimbra.

LSTM Long Short-Term Memory Neural Networks.

ML Machine Learning.

MRF Markov Random Fields.

NN Neural Networks.

PCA Principal Component Analysis.

UC University of Coimbra.

List of Figures

List of Tables

Chapter 1

Introduction

In recent years we have been able to witness unprecedented developments both in the field of *Intelligent Systems* as well as *the Internet of Things*. Smart Algorithms and Artificial Intelligence (AI) have become part of our daily lives, and not only are they now a pillar of industry (Siemens, 2021), but also of society itself. From automating farming to dealing with train logistics, as well as suggesting new products to consumers through social media, *Smart Algorithms* are ever more present in our lives.

And with each passing year, technology as a whole evolves even further (Reock, 2020), becoming ever more intertwined both within itself as well as with its users. The average smartphone owner can now boast both an high-end 4k camera and a micro-computer that not only rivals the most advanced machines of yesteryear but that also has access to any place on Earth, thanks to technologies such as Global Positioning System (GPS) and the advancement of Social Media. All of this within the same package. Our smartphones are truly a powerful tool which, barring a lack of connection to the internet, can have unhinged access to virtually any other piece of technology, information or person, regardless of time or place.

But even in a relatively high-tech world, natural disasters still occur on a daily basis, and contemporary methodologies for preventing and combating these disasters are being overwhelmed. With the worsening of climate change in recent years (Mohanty, 2021), both the number of instances and severity of natural disasters have increased. Floods, fires, droughts. These events grow more and more common, and the developed world isn't an exception. Even considering the additional preventive measure taken in recent years, such as the increase in investment in organisations that counter these disasters, such as firefighters and forest management, the increasing trend of disasters has not subsided remarkably. It only makes sense to take advantage of these unprecedented technological advancements, along with the availability of immense amounts of data, to better complement any project intent on helping to mitigate these dire events.

1.1 The FireLoc Project

The FireLoc Project is one of the systems that aims to help in the fight against natural disasters by harnessing the power of , widespread technologies. With universal access to smartphone cameras and the internet, Fireloc aims to draw on crowd-sourced data to locate, pin-point and monitor forest fires, with the goal of assisting authorities in the early identification and geo-location of ignitions so that these may be tackled with as little delay as possible.

FireLoc [1] is a joint effort of several professors of University of Coimbra (UC) and members of Institute for Systems Engineering and Computers at Coimbra (INESC) (Alberto Cardoso et al., 2021), funded by the Foundation for Science and Technology (FCT), an organization within the Ministry of Science, Technology and Higher Education of Portugal. The FireLoc system uses data collected by citizens using a dedicated app that enables the automatic triangulation of observed fires from the few known observation points. On top of the geo-located coordinates, there is also an option to submit geographic coordinates, text and/or imagery on the app. By analysing this data, the software would then be able to submit a more detailed description of the event to the respective authorities.

FireLoc is yet to be completed, but is currently built on three main components:

- The data collection component (i.e., the FireLoc app) which was developed with mobile devices in mind, while also being complemented by other modules that allow the collection of data through additional sources;
- A data integration and processing component, which handles duties such as geolocating observed events with the available data, assess upload on an user basis, and estimate the risk of events;
- The data visualization component, which includes a multi-platform Graphical User Interface (GUI) meant to be used by the authorities and other end-users, as well as an administrative interface.

More specifically, the modules that this document will focus on are part of the second and third components, and will be elaborated on in the following subsection. But firstly, there is a need to understand what happens before the start of the standard use-case that the prototype module is expected to follow.

The **new prototype module** effectively receives events in real time, which, at the time of input, would've already been analysed by the previous FireLoc modules. These Fireloc modules search for text and visual information regarding both fires and other landmarks or objects that may affect the final risk evaluation, some examples being gas stations or chemical plants. The modules in question also have access to other sources of information such as open-source satellite imagery, meteorological data, or data from the OpenStreetMap[2] project, which complement the datasets with valuable information.

[1] More detailed information at `https://fireloc.org`

[2] More detailed information at `https://www.openstreetmap.org`

The analysis itself is done through AI, using **Deep Learning image recognition algorithms** to find signs of smoke and flames, as well as **Natural Language Processing** to understand if the contributed event holds any sort of textual information on the situation, terrain, or any landmarks close by. The analysis of the submitted data done by these modules allows for the creation of simple and accurate events that hold all the relevant information necessary to identify disasters, while also rejecting data that is deemed to be irrelevant by the algorithms. Some examples of irrelevant data would be submissions of pictures with the absence of any signs of fire, or malicious/spam submissions. Once the crowd-sourced data is treated, it is forwarded to the module that this research project aims to create.

1.2 Contribution and Goals of this Project

The goal of this research project is to elaborate and prototype on a new FireLoc module that would manage the processing, validation, and aggregation of data that resulted from crowd-sourced submissions and other FireLoc sources. It's expected that the Fireloc system will deliver several inputs, which will be used by this prototype module to produce concise and aggregated information about each identified event to be displayed within a GUI.

The Fireloc System will handle the early processing and validation of these volunteer contributions, while the prototype module will follow up and process the contributions into a standard event data-structure. For this, the Fireloc contributions need to be analysed by the prototype so as to know whether to associate these contributions to existing events, to create a new event, or even whether to disregard them or not. Should a contribution be associated to an existing event, or should it lead to the creation of a new event, both cases result in the attempt to identify the following pieces of information: geospatial location, the events' chronology (if there were any prior related events), and any element that is deemed to bring forth **new** relevant information about the event (if the event already exists within the database). It is important to note that this new module will be referred to as **"prototype"** or **"prototype module"** throughout this research project, while the Fireloc Project and it's other modules may also be referred to as **"system"** or **"Fireloc system"**.

By employing forms of AI and Machine Learning (ML) so as to autonomously geo-locate, monitor, and also classify the graveness of the events at hand, it's believed that this process would be completed in the most efficient manner attainable at the time of writing this document. This would also bring the additional benefits of limiting both the cost of maintenance, as well as lower the need for human supervision and/or intervention. Finally, by allowing this information to be showcased to the authorities through a visual interface, it is expected that the entirety of the decision-making process would become more streamlined. This would result in an optimal reduction in initial delays when it comes to mitigating natural disasters that can't always be predicted or prepared for. The introduction of this prototype to the process of countering disasters should also reduce the inherent delay that exists in answering ever-changing conditions away from the

front-lines, such as from a logistical hub.

The main goals established in the context of this research project have been set as the following:

- Research on possible methodologies to be used in the creation of our prototype;
- Development of a prototype capable of handling and displaying disaster data;
- Apply the developed prototype to initial field-testing.

These goals are expected to result in the development of a new prototype module to be integrated into the FireLoc system, as well as the publication of the following documentation:

- State of the Art in regards to the context of this work;
- Architecture of the Prototype module;
- Field-testing & Lab results.

By publicising the previous documents, we also hope to contribute to the overall State of the Art of the various fields present in this research project, as well as play a part in the effort to mitigate the effects of Natural Disasters world-wide.

1.3 Document Structure

The remainder of this proposal is organized as follows: Chapter 2 enumerates key concepts that are relevant to the work at hand, as well as the State of Art, which goes over contemporary works relevant to this project along with the technologies and methodologies employed by their authors.

In Chapter 3, the approach and methodologies chosen for the first iteration of research are outlined. This chapter also elaborates on the work-plans of both phases of development, as well as provides a summary of the work achieved during the first phase of development, which took part in the first semester.

Finally, Chapter 4, Conclusion, reviews the previous chapters, while also presenting a summary of the goals to be completed in the next phase of development, which will take place during the second semester.

Chapter 2

Relevant Concepts & State of the Art

The use of crowd-sourced data to attempt to solve problems involving event detection has recently started to become a common approach, at the time of writing this research document. Examples of works that take this approach are (Afyouni et al., 2022), which attempted to detect unusual events (both generic and specific) by utilizing social media. This chapter is reserved to further elaborate on the concepts and methodologies that are relevant both in the context of the **contribution of our prototype** towards the **FireLoc Project**, as well as to the development of a more generic approach to solving the issue of autonomously monitoring events such as **natural disasters** over a period of time. At an early stage, an attempt was done in dividing this prototype concept into coherent parts that represent a specific role or methodology within the prototype. This was done so as to better understand the requirements of the prototype, and to function as starting points in our research for relevant methodologies and other similar works. For each of these groups, a synopsis is presented, which addresses the different methods available for each role, examples of real-world applications, a resume of their inner workings, as well as their individual relevance to the challenges presented in this document.

So far, the prototype has been divided into three main groups of methodologies, which have been named the following:

- Methodologies for Data Correlation [1];
- Methodologies for the Intelligent Component;
- Methodologies for Data Visualisation.

The available technologies and methodologies that would allow each individual group to achieve its goals are numerous and diverse. In the following sections, we will attempt to elaborate on some of the methods that appear to be the most widely used when attempting to face similar technical challenges to those of this research project.

[1] Images and Text are processed beforehand by a different FireLoc module, which includes a Natural Language Processor, among other features

2.1 Methodologies for Data Correlation

According to the contribution and goals set in chapter 1, the prototype needs to be able to merge and correlate different types of data from different sources, ranging from data which only includes simple coordinates to data which may include images and/or text. This process must be done in a way that extracts all the meaningful data that is to be used by the remaining of the prototype module. One possible methodology to meet these requirements is a process called ***Data Fusion***.

Data fusion is the process of integrating diverse information from multiple sources (such as sensors and cameras) to produce comprehensive and unified data about a more complex entity, in a way that proves more desirable than using an individual source of data (i.e. more reliable or efficient) (Chatzichristos et al., 2022).

Data Fusion is a widely used technique across several fields of research when it comes to correlating data in a way that accurately describes the real world and its inner relations. This is due to the need to integrate data from different sources, as using multiple sources of data is a proven way of reducing uncertainties, imperfections, outliers or any other obstructions to meaningful data, as shown in works such as (Abdulhafiz and Khamis, 2013). Data Fusion is therefore commonly used in projects that handle heterogeneous inputs from multiple different sources.

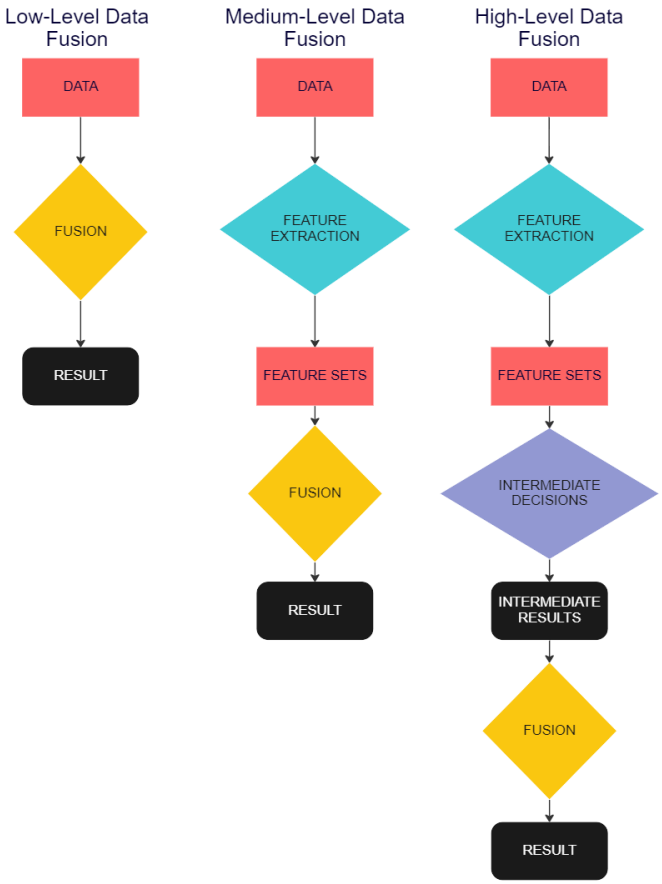

Figure 2.1: The three basic levels of Data Fusion.

There are several methodologies within the concept of Data Fusion (see fig. 2.1) that, in works such as (Kashinath et al., 2021b), (Schmitt and Zhu, 2016) and (Khaleghi et al., 2013), are commonly grouped in three levels: **low (data) level**, also called data association, **medium (feature) level**, also called state estimation, and **high (decision) level**, also called decision fusion. More recently, new types of data fusion are starting to emerge, such as Kernel Data Fusion (Smolinska et al., 2019). In this work, we will only elaborate on the three most basic and well-known types of Data Fusion.

Some examples of projects successfully applying Data Fusion can be seen through the several works done within the field of traffic monitoring, which elucidate on the use of the different levels of Data Fusion used to handle data from multiple sensors, (Kashinath et al., 2021a). (Zhang et al., 2016) delineates an approach on using Data Fusion to achieve real-time urban traffic state estimation using Global Positioning System (GPS) and loop detectors.

Within the context of traffic sensing, a Data Fusion approach would seek to reconstruct data from these sensors with the goal of finding correlations that result in better data, and therefore better decisions and results. The use of fluid GPS coordinates to monitor vehicles is something akin to the aim of our own research project, and is thus a major point of interest.

2.1.1 Low Level Data Fusion

Low Level Data Fusion is considered to be the simplest fusion method to achieve a combination of inputs (Smolinska et al., 2019). In this case, the data is rearranged into a new data matrix, where the different variables are placed one after the other (even if redundant). The resulting matrix will be the sum of the previously separated data.

The goal of low level data fusion is to improve raw data quality at early stages of data processing. It combines several sources of raw data and seeks to produce new raw data, (Floudas et al., 2007). This should in theory result in the newly fused data becoming more informative, synthetic and easier to handle than the original data. At this level, increases in data quality are achieved using techniques such as filters that perform data cleaning and de-noising, as well as estimation of missing values, or removal of outliers. Computationally-wise, this level of Data Fusion is considered to be the lightest.

An example of low-level Data Fusion can be observed through the application of Kalman Filters, which were used in the works of (Wang et al., 2014) to fuse heterogeneous traffic data, along with Gaussian mixture models. Other techniques employ the use of estimation to compensate for missing information and to reduce sampling, with (Saffari et al., 2022) showing an application of Data Fusion in large-scale urban networks, or the use of weighted averages to improve the precision of sensor measurements by correcting these same measurements with Data Fusion, as seen in (Yang et al., 2019) where it was applied to object speed measuring.

2.1.2 Medium Level Data Fusion

Medium Level Data Fusion is based on feature extraction which maintains relevant variables while eliminating undesirable variables from the datasets. This can be done with a multitude of algorithms that have been developed for this purpose (Smolinska et al., 2019), (Floudas et al., 2007). The use of these algorithms often requires thorough research of the features to develop an efficient solution, however.

The goal of medium level data fusion is to merge and filter data so as to extract meaningful features from different sources. Unlike low-level data fusion, there's and additional step prior to the final fusion step that is common to all Data Fusion techniques. It is within this step that fusion of features extracted from the original data occurs. In other words, data reduction is applied through one of several available feature reduction methods. One way of achieving this data reduction is through Principal Component Analysis (PCA). This step is where the main difficulty of medium-level fusion lies, because of the necessity to understand which features are relevant and need to be kept, and which are not and are therefore undesirable, as well as the innumerable ways of achieving this with varying levels of performance.

(Huang and Narayanan, 2016) applied Medium Level Data Fusion within the field of healthcare, by researching on the detection of falls of senior-citizens. These falls were monitored through several different sensors in different axis, such as wearable sensors and cameras, and the data gathered by these sensors was then correlated with feature-level data fusion and support vector classification. This resulted in a higher detection rate and lower false alarm rate. Other applications of this kind of data fusion can be explored in the works by (Zhu et al., 2017), where GPS data was fused with data from users phones to estimate the time until a bus reached a designated bus stop, and (Gao et al., 2019) where video was used as the main target of the fusion process. Interestingly, on the work by (Zhu et al., 2017), it was proven that fusion does not guarantee a desirable result, and that correlation structure needs to be taken into account when elaborating a fusion algorithm.

2.1.3 High Level Data Fusion

High Level Data Fusion works on a decision level, and makes use of selection, inference and state estimation. This type of fusion also falls under the notion of distributed detection systems, which uses multiple sensors and estimation to identify objects, (Floudas et al., 2007). The first step of high level fusion is to fit supervised models to each data matrix. These models are regression models which provide continuous responses for the inputted data, deciding on the data class membership using high-level inference. These decisions are later combined into a complex final model. This can be surmised into a process of selecting one out of a multitude of hypothesis, while taking into account both the decisions of a given number of sensors as well as the effect which noise and interference have on these same sensors (Smolinska et al., 2019).

(Soua et al., 2016b) showcased an application of high-level data fusion by utilizing it to estimate the final destination of ongoing traffic in their works. Another work of interest within the topic of traffic monitoring is (Soua et al., 2016a), where data fusion was used along with fuzzy logic, the latter used to simulate human reasoning rather than binary logic in state estimation, with the goal of making the most efficient transitions of traffic light state within a large urban junction by utilizing state estimating.

Due to the inherent traits of High Level Data Fusion, it is widely used in several other fields outside of traffic monitoring. Classification problems such as industrial quality control and maintenance are one interesting application of decision-level data fusion, with the works of (Wei et al., 2021) as a particularly interesting example in the field of aircraft manufacturing. In the context of this research project, the most interesting work found during our research into the available methodologies for this chapter was (Texier et al., 2019), which used decision-level fusion for disease outbreak detection and surveillance. This work proved that data fusion based approaches were at least equivalent to all contemporary standard algorithms, and oftentimes even yielded a great efficiency gain.

2.2 Methodologies for the Intelligent Component

The next component is the **Intelligent System**, which receives the now fused data from the **Data Handling component**. As stated in chapter 1, this component is meant to be an intelligent component that utilizes forms of Artificial Intelligence (AI) and Machine Learning (ML) to model the fused data into something that can then be displayed by the the **Data Visualisation Component**.

As a **brief introduction to Intelligent Systems**, AI is a tool which enables a machine to simulate human behaviors. ML on the other hand is a subset of AI, which allows a machine to automatically learn from past data without programming explicitly for a single goal. This usually means taking data and looking for underlying trends within it through the use of a wide ranging choice of algorithms. AI are designed to make decisions, often using real-time data. Using sensors, digital data, or remote inputs, they are capable of combining huge amounts of information from a variety of different sources, and act on the insights derived from this data, in some circumstances even without any form of human insight or supervision. This allows the users of AI to access automation capabilities that are next to unlimited.

In the context of this contribution, it is seen as of great importance to be capable of dealing with an huge influx of data in real-time in a most efficient manner. The Intelligent System Component is required to handle a dataset made up of fused data from the previous component, the bulk of which are events made up of simple geo-located coordinates. It would then proceed to analyse and mold these data so that they can be used by the following component. More precisely, the Intelligent System Component needs to meet the following requirements within acceptable time limits for real world use:

- Receive a dataset of events from the other components of the Fireloc System;
- Organise events within the dataset by their similarity to each other;
- Prioritise certain events depending on their content and who submitted them;
- Leverage redundancy and handle duplicate events and submissions;
- Allow for a Sequence/Progression of events.

There are several contemporary AI techniques that are capable of fulfilling these requirements, and not all of these requirements call for the use of ML to achieve the most efficient solutions, since ML can easily result in additional computational loads. The following sub-chapters elaborate on these requirements based on the methodologies we found to be most commonly employed to solve similar issues. The expected uses that ML is meant to have in this research project are, firstly, in the field of Clustering data for visualization of similarities between events and density in geo-location, and secondly, the use of deep-learning in the assignment of priority to events, as well as the management of redundancy and duplicate events. Finally, we also expect to use forms of Data Fusion and Data aggregation to handle incoming inputs from FireLoc. Due to the necessary emphasis on automation, following trends and leveraging data, it was decided to narrow down the relevant techniques to be researched and evaluated to the ML techniques that are elaborated in the following sub-chapters. These were found to be decisively relevant candidates for handling event geo-locating and monitoring problems such as the one described in this document, having already been used to solve similar problems such as those described in (Song et al., 2010), (Haldi Widianto et al., 2020) , or (Afyouni et al., 2022).

2.2.1 Novel Approaches to Clustering

This work defines a clustering approach as novel in the case it employs the use of algorithms which weren't created, or aren't commonly used strictly for the purpose of clustering data. Examples of these are Markov Random Fields (MRF) and Bayesian Networks (BN), which see widespread usage across several research fields with roles unrelated to clustering.

Markov Random Fields

MRF, also known as Markov Networks, are a form of representing dependencies. They describe a system by local interaction and denote features of a system by using terms representing their spatial or contextual dependencies. Markov networks are also undirected and can be cyclic, which allows this model to represent infinite loops in its dependencies. A benefit of MRF is that these types of networks are designed based on both statistical and structural information that standard clustering methods tend to neglect (Wang et al., 2013). In the case of our

research project, this would allow for more specific grouping when using MRF for clustering.

An example of this benefit would be the following scenario: if, although point A is closer to B, and as such standard clustering methods such as K-means group them together, there could be other more relevant ways of grouping these points such as A with C, when taking into account topology and elevation. We refer to this as the Elevation Problem, which is exemplified in figure 2.2, with the expected K-means grouping in red, and a custom MRF in green.

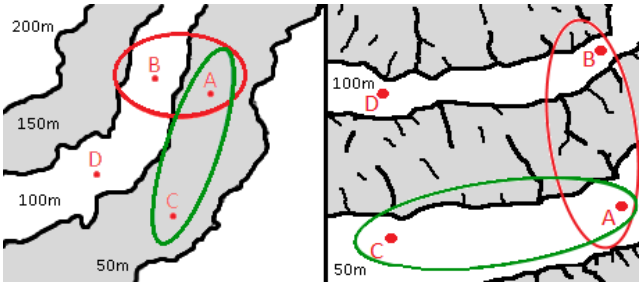

Figure 2.2: Examples of the elevation problem, from two perspectives.

Within this context, it would make more sense to group point D with point B and point C with point A, this being due to the nature and accessibility of the terrain. MRF are also widely used in the field of terrain mapping due to this characteristic, as seen in works such as (Tse et al., 2015).

MRF also face certain setbacks, however. They are computationally demanding, and while there are many algorithms that are capable of optimising these networks, one of which is *Iterated Conditional Modes*, but this then adds another layer of complexity to the solution. MRF are often used in image processing and computer vision. Using these networks for clustering is a novel approach which is well documented in (Song et al., 2010). Other interesting works that use this type of technique for clustering purposes are often found in the field of medicine, thanks again to the higher accuracy of this technique. Examples of these works are (Suliga et al., 2008) where the goal of MRF is to find clusters of pixels that represent cancerous masses. Another more relatable research project which used MRF for clustering purposes is (Li et al., 2021), however. In this specific work, MRF were used to cluster social events for organisers in the context of event-based social networks, with the goal of event management. This specific work proves us that MRF could effectively be used in the context of our research project. Overall, it can be concluded that MRF are more robust and detailed than standard Clustering methods, at the cost of computational performance, proven by the results shown in the previous works. This means that MRF are the better option when accuracy takes precedence over any other metric when evaluating an algorithm.

Bayesian Networks

BN are probabilistic models that share similarities with MRF. One difference, however, is that Bayesian networks are directed and acyclic, so they can induce

dependencies between events, making these models ideal for predicting the contributing factor of an event, or in other words, patterns. Unlike MRF however, they can't handle infinite loops within them (cycles), and so if the data was generated from a model where several variables correlate to each other then BN wont be able to model this relationship (Ben-Gal, 2008).

A relevant example would be to predict relationships between diseases and their symptoms, one of many examples being in cancer research. Such a case was extensively described in (Zhao et al., 2021), where BN were used with the goal of revealing molecular structures of tumours so that these could then be further researched. In the previous research article, the performance of BN proved to be better compared to the other algorithms it was up against, mainly due to the inherent pattern-finding traits of this particular type of network. BN, not unlike MRF, are also considered computationally demanding. All branches must be calculated in order to calculate the probability of any one branch. Not only that, there is no commonly accepted way for creating BN from data, all while they are considerably difficult to create, requiring "*a priori*" knowledge for achieving the most efficient solutions. In works such as (Pham and Ruz, 2009), it was demonstrated that, at the cost of extra computation power, BN prevailed over standard clustering techniques such as K-means, which were shown to be 10% less accurate.

Another interesting work that uses this type of technique for clustering purposes is (Marek et al., 2014). In this specific work, spatial analysis was applied to medical datasets with the goal of mapping disease events through clustering. According to this work, Baysean algorithms are already widely used to smooth data so that become easier to spot. This work created clusters both in the spatial and space-time planes, with the clusters depicting the risk of health anomalies in a density map.

2.2.2 Standard Clustering Techniques

There are many clustering techniques described in contemporary literature, such as Probabilistic, Partitional, Spectral or Grid based Clustering, amongst others. The two main Clustering techniques considered for this contribution are Centroid-based and Density-based Clustering, which are the most well-documented and commonly used techniques.

Centroid-based clustering

Centroid-based clustering organizes the data into non-hierarchical clusters.
Centroid-based algorithms are simple, efficient and scale well to large datasets. Examples of this technique include **K-means**, which is the most widely-used centroid-based clustering algorithm. K-means aims to partition data into clusters in a way where each individual observation belongs to the cluster with the nearest mean to a cluster centroid. The initial k centroids are randomized, and as the cluster grows the center is recalculated (LEDU, 2018), (GeeksforGeeks, 2023).

An example of the employment of K-means over a dataset of points can be seen in figure 2.3. This technique is often used in Market and Image Segmentation, but is far from limited to these fields.

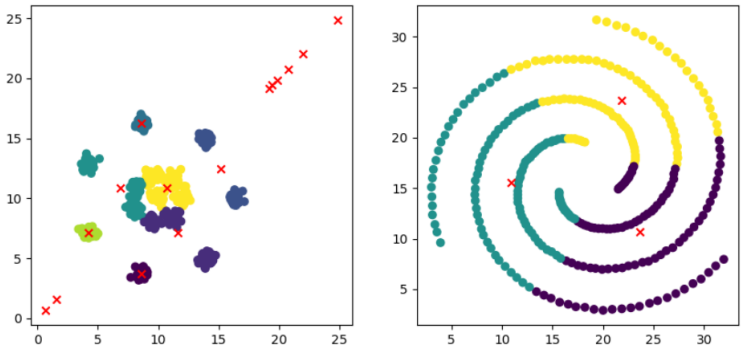

Figure 2.3: An example of K-means applied to two clustering problems.

Some of the issues with K-means include sensitivity to initial conditions and outliers, requiring manual setting of an optimal K-value and distance metric, and the inherent randomness to K-means, which can result in less efficient results on the short-term. These variables influence the shape of the clusters. K-means also has difficulty handling clusters of varying density and/or arbitrary shapes.

When it comes to event detection, use-cases of K-means can be seen in works such as (Oladimeji et al., 2015), where K-means was applied with the goal of event detection within systems such as fire alarms. This work achieved the goal of event detection by performing data aggregation on the nodes created by K-means, which clustered the data around two possible labels (the two possible outputs of the system) and then followed with pattern recognition utilizing Convolutional Neural Networks (CNN). This work concluded that utilizing this combination of methodologies significantly improve fire detection performance when compared with what were defined as standard approaches: Feed-Forward Neural Networks (NN) or Naive Bayes Classifiers.

Density-based clustering

Density-based clustering connects areas of high density into clusters. Examples of this technique include **Density-based Spatial Clustering of Applications with Noise (DBSCAN)**, which is mainly used to find relevant associations and structures within data. This is a very simple to implement technique that only requires two initial inputs, the minimum size of a cluster, and the maximum distance between its members. Density-based clustering allows for arbitrary-shaped distributions as long as dense areas are present in the dataset (Dey, 2023). By also having a notion of noise, density-based clustering is, by design, more robust to outliers since it does not assign them to clusters. Unfortunately, these algorithms still have difficulty with data clusters of varying densities, much like K-means. The previous dataset of points utilized for the K-means example can now be seen in figure 2.4 being clustered with DBSCAN.

The main theory behind this algorithm is quite simple: a point belongs to a cluster

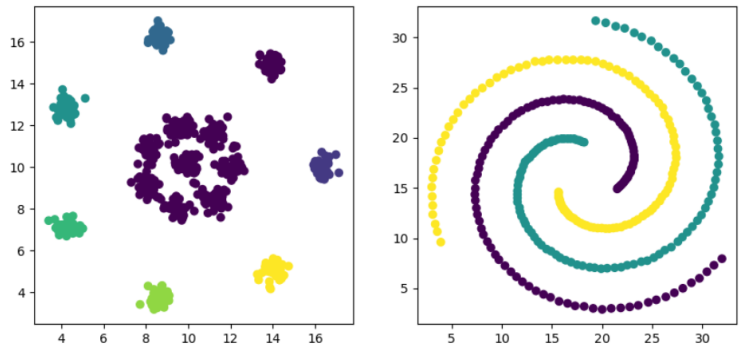

Figure 2.4: An example of DBSCAN applied to two clustering problems.

if said point is close to other points from that same cluster. This proves to be a challenge when the dataset is sparsely populated.

DBSCAN has in fact been used in works similar to this contribution before, as described in (Haldi Widianto et al., 2020), which elaborates on the use of DBSCAN with the goal of identifying disasters through Twitter. DBSCAN has also been used in both (Karanja, 2016) (Anwar et al., 2019), where DBSCAN was employed to identify areas with wildfire risk, utilizing historical data of wildfire hot-spots as occurrence points, and then calculating risk based on cluster density. This makes this technique particularly relevant in the context of this documents research topic. Another interesting work in the context of using clustering to observe trends is (Cerezo-Costas et al., 2018), which used geo-located social media posts to help detect and understand unexpected behaviors in urban areas in real time, some examples being abnormal patterns and contrasting location densities which could signify a multitude of activities. This was achieved using several social media platforms such as Twitter and Instagram, and was put to test within large urban areas such as New York. The methodologies used along with DBSCAN included Natural Language Processing to automate the understanding of the social media posts posts, and thread-based data aggregation techniques. Another interesting work in the context of geo-located events is (Huang et al., 2018), where DBSCAN was once again used with the goal of detecting events and their textual content which could then be used for a multitude of research purposes, such as marketing or geo-social studies. This work focused on Twitter posts as a source of data, and utilized DBSCAN to cluster tweets according to their spatial and temporal characteristics. Afterwards, these tweets were subjected to analysis by a text processor. The results proved to be promising, as using data collected from four college cities over the span of two years resulted in the identification of several events that occurred within that time-frame.

These previous works which utilized DBSCAN show several common traits with the goals of our own research project, and thus we found them to be very enlightening in the context of handling event detection in both the spatial and temporal spaces. Being a standard clustering technique however, DBSCAN suffers from some inherent problems when it comes to accuracy. The earlier elevation problem that MRF solved still stands when using DBSCAN, so caution is needed when dealing with three-dimensional data, or two-dimensional data that uses elevation data, such as with topographic maps. Depending on the accuracy needed, either

in geo-location coordinates or local terrain characteristics, DBSCAN may not be a viable methodology for certain works.

2.2.3 Approaches to Data Prioritisation

This specific part of the Intelligent Component is required to be able to decide on which events take precedence over others, and as such this can be surmised to a simple decision problem that can be tackled with deep learning. Within the field of ML alone there are already several methodologies capable of solving these types of problems, and in the following subsections we lay down the ones we found most common in the works that were researched in the context of this project.

Convolutional Neural Networks

CNN (see fig. 2.5) are a type of Artificial Neural Network, often used in image processing, but not limited to this field. Neural Networks, in the context of computing, are systems which attempt to mimic the workings of biological neural networks and how they process information. Artificial Neural Network make part of the concept of Deep Learning, a branch of Machine Learning that focuses on Neural Networks. These networks are multi-layered (hence "deep" learning), and can have a multitude of different layers. CNN are feed-forward NN that utilize convolution blocks within a computational operation that filters an input matrix. Neurons move along an input matrix, convoluting with filters, and resulting in feature maps. These feature maps are then merged through several convolution operations, resulting in a layer output (Mishra, 2020).

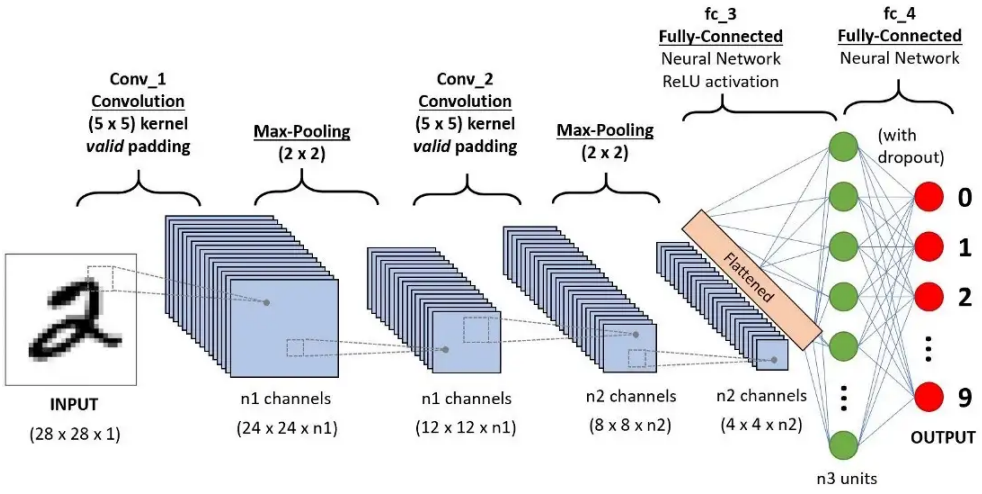

Figure 2.5: An example of the architecture of a CNN, by (Saha, 2022).

One issue CNN can have is their proneness to over-fitting, due to the fact that layers of neurons can be completely connected to every individual neuron of the previous layer. A benefit of CNN is, however, their adaptability. One can utilize pre-made, pre-trained networks, and adapt these same networks to solve

problems that are unrelated to the networks initial goals. This is done through **Transfer Learning**. They are also automated, unlike other networks. CNN adapt and optimize the used filters.

Within the works already mentioned, those who had similar requirements to our own utilized methodologies involving NN, the most common of which was found to be CNN, or derivatives of this same type of NN. One of these cases was (Afyouni et al., 2022), which utilised both CNN and Long Short-Term Memory Neural Networks (LSTM) in the Event Classification step, which sought to detect and classify social media posts of interest. In a sense, this step is similar to the step laid out for data prioritisation within our prototype, as both cases can be considered classification problems. Another work of interest which utilizes CNN for event monitoring is (Dabiri and Heaslip, 2018), which utilized these networks in the context of detecting and monitoring traffic accidents through social media posts. These networks were used both as a way to analyse and convert the tweets into data, as well as to classify them into one of the three possible labels in the context of that work (non-traffic, incident, condition).

Due to the widespread use, as well as the efficiency and ease of application of CNN, these networks are proven to be a good candidate for fulfilling the requirement set in the context of data prioritisation within our research project.

Long Short-Term Memory Neural Networks

The second method that appears to be used in similar works to our own are LSTM networks (see fig. 2.6). LSTM are another type of Artificial Neural Network. A major difference between LSTM and CNN is that the former has feed-back connections. In fact, LSTM are a type of recurrent neural network, a network that is capable of forming cycles within its connections, and thus allowing for dynamic behaviour (Dolphin, 2020). For example, this behaviour allows these networks to compute both single data points as well as sequences of data, while CNN's only allow for points of data. In fact, LSTM were created for the purpose of solving the **"vanishing gradient problem"**, which is inherent to other recurrent neural networks, and can be summed up to the gradient used by these types of networks becoming so small that propagation and training become impossible (Wang, 2019).

LSTM are used for a wide variety of roles, but they excel at roles that need to handle time series data. This includes classification and prediction problems, natural language, and sensor data. The reason these specific networks excel at these roles is because they can handle empty space within a time series. "Long Short Term Memory" Networks are capable of learning long-term dependencies within sequences by memorizing short-term, moment-to-moment changes. In other words, the LSTM architecture allows them to choose whether to retain information within its short-term memory or to discard it. This allows for "Longer" Short term memory, which is used to remember dependencies.

LSTM networks are not without faults, however. They are more complex than CNNs, and require manual setting of several more initial parameters when com-

pared to CNNs. Another of the drawbacks of these networks is also their computational weight. They are harder and take longer to train. They are easy to over-fit, and use up more memory. These networks are also sensitive to random initialization. And finally, they can't handle infinite sequences, unlike standard RNNs (aditianu1998, 2021).

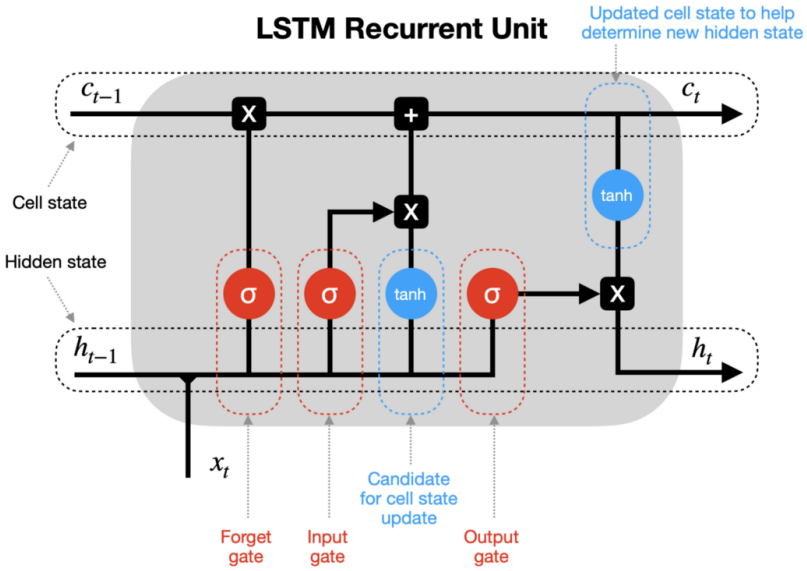

Figure 2.6: An example of the architecture of a LSTM, by (Dobilas, 2022).

Taking into account figure 2.6, we can see the main components of an LSTM network:

- Forget Gate - decides what old information from previous inputs to forget;
- Input Gate - decides what new information to remember;
- Output Gate - decides which cell state to output.

While these networks didn't seem to be used in previous works as commonly as CNN, we found them to be employed in works researched in the context of this project non-the-less. As mentioned in the previous sub-chapter, (Afyouni et al., 2022) utilized this type of neural network, along with CNN, in similar roles, within the project of Deep-eware. Another interesting work that utilized this type of networks was (Zhang and Pan, 2019), where LSTM were utilized to identify and classify protest events within Chinese social media. These networks were also put to use in the works of (Zhang et al., 2018), a work that attempts to detect traffic accidents. The goal of this work was to utilize deep-learning as a way of identifying traffic accidents and other abnormal events through social media posts and crowd-sourcing.

2.3 Methodologies for Data Visualisation

Clustering, by itself, already provides a similar presentation to density maps, and thus shows by default which areas seem to be experiencing more activity. This can be seen in chapter 2.2.2 - Standard Clustering Techniques, in figure 2.3. Keeping this in mind, the Data Visualisation component of this project will also need to handle the display of information of individual events, such as risk, landmarks or event coordinates, as well as background information that is inherent to the terrain where the events are happening. This requires the reader to be able to select individual events, and to be able to overlap the clusters with maps depicting different types of terrain information.

All of this information would also need to be presented in a way that allows for the display of temporal sequences of all event clusters, and allow for visual cues on what is happening within the area of events, as well. A valid technique to display these developments could be heat-maps (see fig. 2.7).

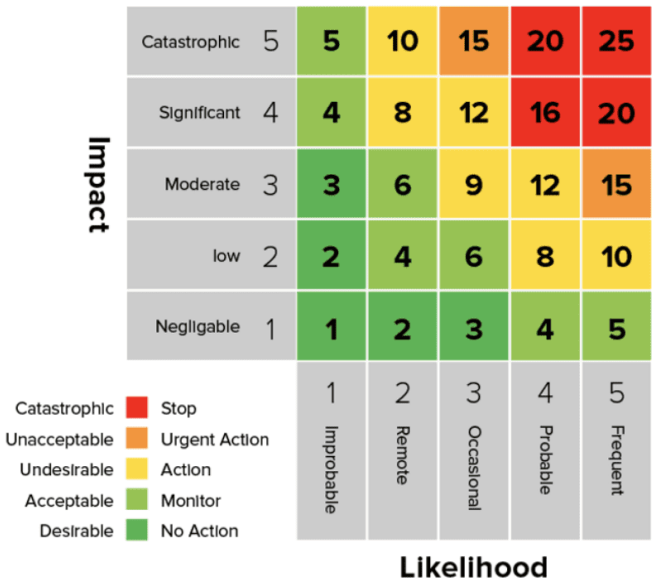

Figure 2.7: An example of a risk-based heat-map from (Balbix, 2022).

This technique shows the magnitude of events through color-coding, with the variation giving obvious visual cues about the specifics of the events: how they are clustered, how they vary over space and time, or the importance of each event on a certain scale. This last option is particularly relevant to this project because not only can it be used to represent "risk", but heat-maps themselves can be used in conjunction with density-maps to provide a very visually intuitive map interface.

Chapter 3

Proposed Approach and Work Plan

The goal of this chapter is to elaborate on the work that was done during the first semester, as well as to plan the approach to be used in the development of the **Prototype Module**. This prototype is a contribution to the FireLoc System, and is meant to handle crowd-sourced data from FireLocs' modules so as to create relevant geo-located events that can then be visualized within an interface, in a way that aids the authorities in their efforts to combat natural disasters, with a focus on forest fires from our part.

3.1 The Current Work

The first step in the development of our prototype was the study of the state of the art and previous works that attempted to answer similar problems involving geo-locating and monitoring events not limited to disasters or fire detection. Several works regarding the monitoring of geo-located events were researched, from traffic monitoring to multi-sensor health systems, from social media crowd-sourcing to urban development.

The following step was to define a list of tasks that need to be completed by the prototype, so as to better understand the full requirements that need to be met. This list was created by taking into account the information gathered within the two previous chapters of this document. We found that to achieve the goals defined in this research project, the prototype needs to meet the following list of requirements:

- Receive a dataset of events from the other components of the Fireloc System;
- Aggregate and/or fuse data within the received inputs;
- Prioritise certain events depending on their content, age, or who submitted them;
- Organise events within the dataset by their similarity to each other;
- Leverage redundancy and handle duplicate events and submissions;

- Allow for a Sequence/Progression of events;
- Assign a lifespan for each event;
- Visualise individual event logs;
- Visually display the events in a relevant and informative way.

With the goal of meeting these requirements, several methodologies which were researched and elaborated upon in the previous chapter were considered. To better understand the role of these methodologies, as well as the stage at which each one of them is supposed to take action within the prototype module, the following flowchart present in figure 3.1 was created:

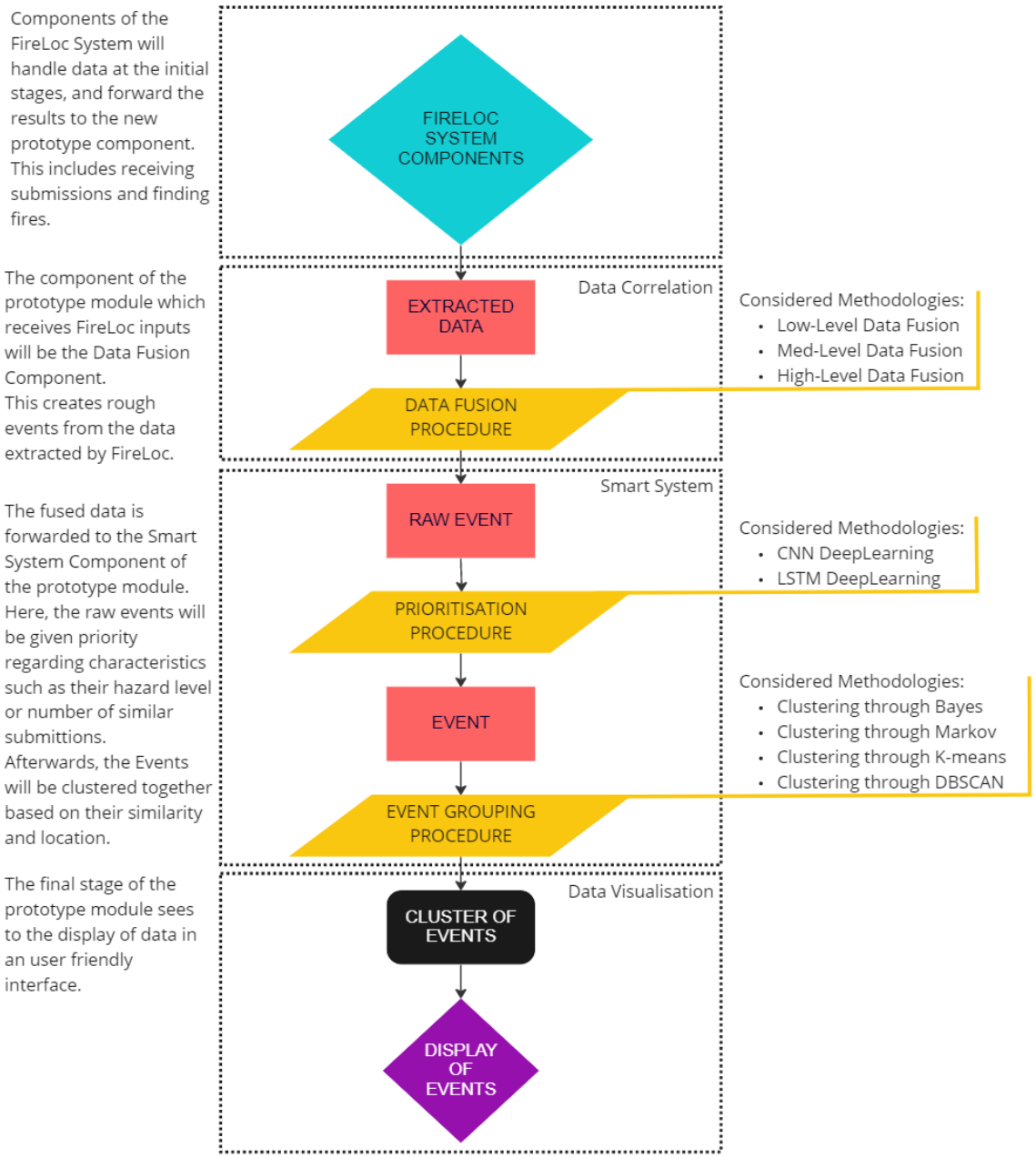

Figure 3.1: Flowchart of the Components that make up the Prototype Module.

Some degree of testing will be necessary to effectively choose the most apt methodologies for each component of the prototype. This will be done on top of the testing done to strain-test the prototypes' reliability. The prototype needs to be used in real-world conditions and be able to face highly volatile volumes of submissions, in real-time and within acceptable accuracy standards. An important factor in testing is that the prototype will deal with forest fires, which are disasters that put lives and property at risk, and thus a higher degree of accuracy is demanded.

Some additional difficulties include the technical demands of utilizing different methodologies within the same module, the several roles the prototype is meant to have within the FireLoc System, and the need to handle real-time, dynamic inputs. Due to the novelty of this research project, there is also a lack of standards for what truly makes a representative dataset for the training and testing of this specific prototype, when applicable. The datasets used for testing need to be representative of all possible situations the module will face on the field so as to provide as much accuracy as possible. There can also risks due to the inherent traits of each of the methodologies used. For example, the components involving CNN and LSTM networks require deep-learning, which require long periods of training and result in high computational loads.

3.2 The Approach

Having defined the requisites, available methodologies, and after understanding the possible constraints to be faced, both by the prototype and its creators, our proposed approach for the development of the actual prototype module consists of the following steps:

- An iterative development of each individual component of the prototype module, followed by individual testing for fulfillment of the requisites of each component;
- Bridging of each component, into a single working entity, followed by testing for fulfillment of the requisites of the prototype as a whole;
- Strain-testing, performance-testing, and field-testing of the prototype, followed by iterations of optimization and bug-fixing.

The main objective being the creation of a fully functional prototype, capable of meeting all the set requisites. Following the flowchart, for each component of the prototype, a methodology needs to be chosen and optimized for the problem, so as to obtain consistent and accurate results, both in real-time and on the field.

At the current stage of research, and based on the state of the art chapter, we believe that the methodologies that should be used are, in the order of the previous flowchart, High-Level Data Fusion, CNN Deep-Learning, and finally, DBSCAN. Before committing to this, however, early testing will be done to confirm the viability of each methodology that is under consideration. Early testing would occur

during the initial month of the second semester, at the same time as the improvements to the State of the Art are expected to be done. This would allow the testing of the methodologies' viability without committing too much time to a single solution at the early stages of development.

3.3 The Work Plan

So as to better organise and manage the workload through both semesters, as well as to facilitate development by stating multiple fixed milestones, the following Gantt charts (figures 3.2 and 3.3) were created:

The First Semester

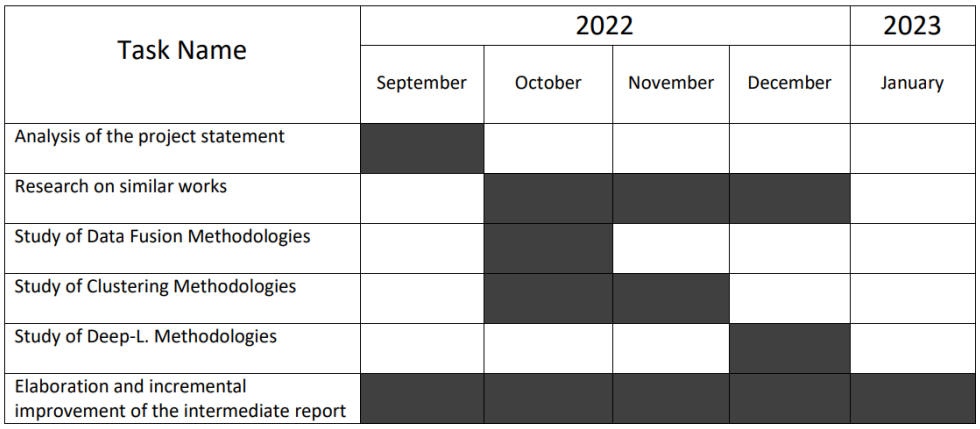

Figure 3.2: Gantt chart of the work plan for the first semester.

As stated in section 3.1, the first semester was used to understand what the FireLoc System required of the prototype module, as well as to research the state of the art and similar works, along with works that employed methodologies of interest to this research project in a relevant matter, but that were not necessarily related to our project.

The Second Semester

As stated in section 3.2, the second semester will focus on the development and testing of the prototype module. Once the components of the prototype are operational, an additional effort will be made to optimize each component as much as possible without compromising reliability. The final version of the deliverable will also be written in parallel with the prototype development iterations.

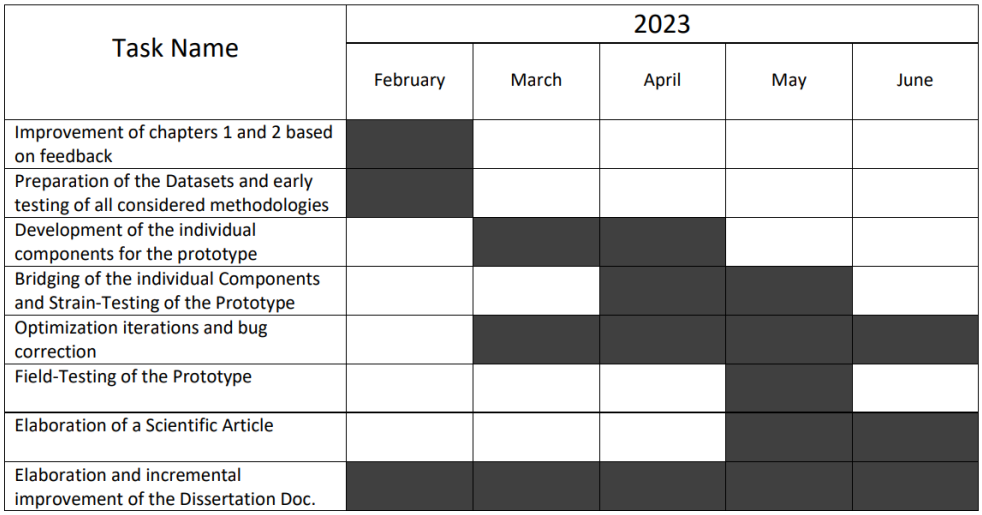

Figure 3.3: Gantt chart of the work plan for the second semester.

3.4 Risk Analysis

Some possible risks were briefly mentioned at the end of section 3.1. Taking this into account, it is important to understand and mitigate risks to the development of a project as early as possible in the projects' lifespan. For this, a number of risks will be evaluated based on a risk scale (table 3.1). The Severity and Probability factors will be assigned values within a scale, which will then be used through a risk matrix to understand the true weight of these risks on the project.

Table 3.1: Risk Scale table.

Value	Probability	Severity
1	Low	Negligible
2	Med	Marginal
3	High	Critical
4	V. High	Catastrophic

This table helps define weights for the risks. For both Probability and Severity, a numeric value is assigned. Probability represents the chance of a certain risk happening during development, while Severity represents how badly that risk would influence the development process.

Risk Enumeration

Table 3.2: Risk Enumeration table.

Risk ID	Context	Description	Prob.	Sev.
R1	Technical	There are several methodologies which need to work in sync	2	3
R2	Technical	The chosen methodologies may end up being nonviable or unnecessary	1	4
R3	Time	Early testing is needed, and could quickly spiral into causing delays	1	2
R4	Time	Unexpected events can occur during development and cause delays	1	3
R5	Time	Training of CNN/LSTM can become extremely time-consuming	1	2
R6	Inherent	Optimisation attempts can cause bugs and prove to be fruitless	2	1
R7	Inherent	Bug-hunting and fixing features may cause delays in, or block milestones	4	2
R8	Data	Data may not always be available, or be in a state that makes it difficult to use	2	4
R9	Technical	Since this project is a contribution to FireLoc, it is also highly dependent on it	3	4

Table 3.2 helps enumerate the risks and their respective context, and also assign the previous risk weights to individual risks. Probability and Severity are assigned based on the expected effort that would need to be employed to solve the issues caused by the risk happening, as well as the expected chance of said risk happening at any stage of development.

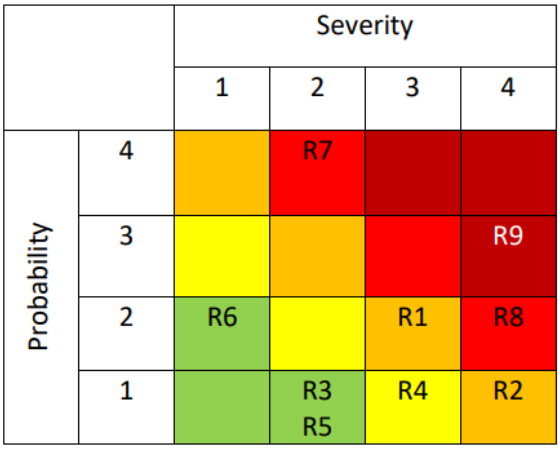

Figure 3.4: Risk matrix depicting the true weight of each risk.

The resulting risk matrix (see fig.3.4) allows us to see the weight of each risk in a visually intuitive manner. Risk weight is represented in a gradient going from green to red, where green represents lower risks, and red represents higher risks.

Risk Mitigation

There are many tools and methodologies which help in reducing risk. For each risk, we found the following actions to be a viable source of risk-reduction:

- R1 - Good coding practices and modularity help mitigate this issue;
- R2 - Early testing and research mitigate this issue, but cause issues of their own;
- R3 - Assigning time limits to tasks solves this issue;
- R4 - This risk is random and to an extent, unavoidable;
- R5 - The use of various tools, libraries and frameworks helps in mitigating this issue;
- R6 - All that can be done is to limit the time used in this task;
- R7 - Good coding practices help mitigate this issue;
- R8 - Use of open-source data, as well as creating **synthetic** data for testing may mitigate this issue;
- R9 - Close collaboration with FireLoc colleagues will help in solving this issue, as well as good documentation, standardisation, system modularity and system backups.

Chapter 4

Conclusion

The **First Phase** of development of this project involved the research of the state of the art, as well as the study of similar and relevant works. A distinction was made between "similar" and "relevant" works because it was seen as important to widen the research to not only works with similar goals, but works that applied similar techniques to other problems that were not necessarily similar to our own research project (e.g. Medicine, Traffic Management).

This initial research culminated on a number of methodologies and concepts being selected and studied, which in turn helped in finding more works of interest. Parallel to this, a basic list of requisites was created, so as to better guide the research process. This research was followed by the synthesizing of the work performed in the first semester, as well as the planning of the approach to follow on the second phase of development.

The **Second Phase** of development, which will take place during the second semester, is based on the results of previous works which were found to be relevant to this project. Unfortunately, it was not possible to carry out early testing of the methodologies which are under consideration during the first semester, so this was postponed to the first month of the second semester.

The second phase will focus on the actual development of the prototype module which is to be used alongside the FireLoc System. A work-plan was set, along with several milestones. To showcase this, Gantt models were created, one for each phase of development. Possible difficulties for the second phase were also pondered, so as to understand how to avoid or mitigate them as early as possible.

References

Waleed Abdulhafiz and Alaa Khamis. Handling data uncertainty and inconsistency using multisensor data fusion. *Advances in Artificial Intelligence*, 2013, 01 2013. doi: 10.1155/2013/241260.

aditianu1998. Understanding of LSTM Networks - GeeksforGeeks — geeksforgeeks.org. `https://www.geeksforgeeks.org/understanding-of-lstm-networks/`, 2021. [Accessed 05-Jan-2023].

Imad Afyouni, Aamir Khan, and Zaher Al Aghbari. Deep-eware: spatio-temporal social event detection using a hybrid learning model. *Journal of Big Data*, 9, 06 2022. doi: 10.1186/s40537-022-00636-w.

Jacinto Estima Alberto Cardoso, Cidália Fonte, José Paulo de Almeida, and Joaquim Patriarca. The fireloc project: Identification, positioning and monitoring forest fires with crowdsourced data. In *The FireLoc Project: Identification, Positioning and Monitoring Forest Fires with Crowdsourced Data*, 2021.

Muchamad Anwar, Wiwien Hadikurniawati, Edy Winarno, and Aji Supriyanto. Wildfire risk map based on dbscan clustering and cluster density evaluation. *Advance Sustainable Science, Engineering and Technology*, 1, 11 2019. doi: 10.26877/asset.v1i1.4876.

Balbix. Risk heat map – a powerful visualization tool, Jan 2022. URL `https://www.balbix.com/insights/cyber-risk-heat-map/`. [Accessed 11-Jan-2023].

Irad Ben-Gal. *Bayesian Networks*. John Wiley & Sons, Ltd, 2008. ISBN 9780470061572. doi: https://doi.org/10.1002/9780470061572.eqr089.

Héctor Cerezo-Costas, Ana Fernández-Vilas, Manuela Martín-Vicente, and Rebeca P. Díaz-Redondo. Discovering geo-dependent stories by combining density-based clustering and thread-based aggregation techniques. *Expert Systems with Applications*, 95:32–42, 2018. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2017.11.019.

Christos Chatzichristos, Simon Van Eyndhoven, Eleftherios Kofidis, and Sabine Van Huffel. Chapter 10 - coupled tensor decompositions for data fusion. In Yipeng Liu, editor, *Tensors for Data Processing*, pages 341–370. Academic Press, 2022. ISBN 978-0-12-824447-0. doi: https://doi.org/10.1016/B978-0-12-824447-0.00016-9. URL `https://www.sciencedirect.com/science/article/pii/B9780128244470000169`.

Sina Dabiri and Kevin Heaslip. Developing a twitter-based traffic event detection model using deep learning architectures. *Expert Systems with Applications*, 118, 10 2018. doi: 10.1016/j.eswa.2018.10.017.

Debomit Dey. Dbscan clustering in ml: Density based clustering, Jan 2023. URL `https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/`. [Accessed 08-Jan-2023].

Saul Dobilas. Lstm recurrent neural networks how to teach a network to remember the past. `https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e`, 2022. [Accessed 04-Jan-2023].

Rian Dolphin. LSTM Networks | A Detailed Explanation — towardsdatascience.com. `https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9`, 2020. [Accessed 05-Jan-2023].

Nikos Floudas, Aris Polychronopoulos, Olivier Aycard, Julien Burlet, and Malte Ahrholdt. High level sensor data fusion approaches for object recognition in road environment. In *2007 IEEE Intelligent Vehicles Symposium*, pages 136–141, 2007. doi: 10.1109/IVS.2007.4290104.

Jun Gao, Yi Murphey, and Honghui Zhu. Personalized detection of lane changing behavior using multisensor data fusion. *Computing*, 101, 12 2019. doi: 10.1007/s00607-019-00712-9.

GeeksforGeeks. K means clustering - introduction, Jan 2023. URL `https://www.geeksforgeeks.org/k-means-clustering-introduction/`. [Accessed 08-Jan-2023].

Mochammad Haldi Widianto, Ivan Sudirman, and Muhammad Awaluddin. Application of density based clustering of disaster location in realtime social media. *TEM Journal*, pages 929–936, 08 2020. doi: 10.18421/TEM93-13.

Che-Wei Huang and Shrikanth Narayanan. Comparison of feature-level and kernel-level data fusion methods in multi-sensory fall detection. In *2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, 2016. doi: 10.1109/MMSP.2016.7813383.

Yuqian Huang, Yue Li, and Jie Shan. Spatial-temporal event detection from geo-tagged tweets. *ISPRS International Journal of Geo-Information*, 7(4), 2018. ISSN 2220-9964. doi: 10.3390/ijgi7040150.

Stephen Karanja. Density-based cluster analysis of fire hot spots in kenya's wildlife protected areas. In *Density-based Cluster Analysis Of Fire Hot Spots In Kenya's Wildlife Protected Areas*, 2016.

Shafiza Ariffin Kashinath, Salama A. Mostafa, Aida Mustapha, Hairulnizam Mahdin, David Lim, Moamin A. Mahmoud, Mazin Abed Mohammed, Bander Ali Saleh Al-Rimy, Mohd Farhan Md Fudzee, and Tan Jhon Yang. Review of data fusion methods for real-time and multi-sensor traffic flow analysis. *IEEE Access*, 9:51258–51276, 2021a. doi: 10.1109/ACCESS.2021.3069770.

Shafiza Ariffin Kashinath, Salama A. Mostafa, Aida Mustapha, Hairulnizam Mahdin, David Lim, Moamin A. Mahmoud, Mazin Abed Mohammed, Bander Ali Saleh Al-Rimy, Mohd Farhan Md Fudzee, and Tan Jhon Yang. Review of data fusion methods for real-time and multi-sensor traffic flow analysis. *IEEE Access*, 9:51258–51276, 2021b. doi: 10.1109/ACCESS.2021.3069770.

Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14 (1):28–44, 2013. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2011.08.001.

Education Ecosystem LEDU. Understanding k-means clustering in machine learning, Sep 2018. URL `https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1`. [Accessed 08-Jan-2023].

Xiao Li, Yashas Malur Saidutta, and Faramarz Fekri. Social event planning using hybrid pairwise markov random fields. *International Journal of Intelligent Systems*, 36, 07 2021. doi: 10.1002/int.22569.

Lukas Marek, Vít Pászto, Pavel Tucek, and Jiří Dvorský. Spatial clustering of disease events using bayesian methods. In *Spatial Clustering of Disease Events Using Bayesian Methods*, volume 1139, 04 2014.

Mayank Mishra. Convolutional Neural Networks, Explained — towardsdatascience.com. `https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939`, 2020. [Accessed 05-Jan-2023].

Arup Mohanty. Impacts of climate change on human health and agriculture in recent years. In *2021 IEEE Region 10 Symposium (TENSYMP)*, 2021.

Muyiwa O. Oladimeji, Mikdam Turkey, Mohammad Ghavami, and Sandra Dudley. A new approach for event detection using k-means clustering and neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5, 2015. doi: 10.1109/IJCNN.2015.7280752.

D. Pham and Gonzalo Ruz. Unsupervised training of bayesian networks for data clustering. *Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences*, 465:2927–2948, 09 2009. doi: 10.1098/rspa.2009.0065.

Eran KinsbrunerandJustin Reock. The evolution of smartphones - and web technology development, Jan 2020. URL `https://www.perfecto.io/blog/evolution-of-smartphones-web`.

Elham Saffari, Mehmet Yildirimoglu, and Mark Hickman. Data fusion for estimating macroscopic fundamental diagram in large-scale urban networks. *Transportation Research Part C: Emerging Technologies*, 137:103555, 2022. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2022.103555.

Sumit Saha. A comprehensive guide to convolutional neural networks the eli5 way. *Medium*, Nov 2022. URL `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`. [Accessed 05-Jan-2023].

Michael Schmitt and Xiao Zhu. Data fusion and remote sensing – an ever-growing relationship. *IEEE Geoscience and Remote Sensing Magazine*, 4:6–23, 12 2016. doi: 10.1109/MGRS.2016.2561021.

Siemens. https://new.siemens.com/global/en/company/stories/industry/ai-in-industries.html. In *AI and Industry 4.0*, 2021. [Accessed 05-Jan-2023].

Agnieszka Smolinska, Jasper Engel, Ewa Szymanska, Lutgarde Buydens, and Lionel Blanchet. Chapter 3 - general framing of low-, mid-, and high-level data fusion with examples in the life sciences. In Marina Cocchi, editor, *Data Fusion Methodology and Applications*, volume 31 of *Data Handling in Science and Technology*, pages 51–79. Elsevier, 2019. doi: https://doi.org/10.1016/B978-0-444-63984-4.00003-X.

Ran Song, Yonghuai Liu, Ralph Martin, and Paul Rosin. Markov random field-based clustering for the integration of multi-view range images. In *Markov Random Field-Based Clustering for the Integration of Multi-view Range Images*, pages 644–653, 11 2010. ISBN 978-3-642-17288-5. doi: 10.1007/978-3-642-17289-2_62.

Ridha Soua, Arief Koesdwiady, and Fakhri Karray. Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3195–3202, 2016a. doi: 10.1109/IJCNN.2016.7727607.

Ridha Soua, Arief Koesdwiady, and Fakhri Karray. Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3195–3202, 2016b. doi: 10.1109/IJCNN.2016.7727607.

M. Suliga, R. Deklerck, and E. Nyssen. Markov random field-based clustering applied to the segmentation of masses in digital mammograms. *Computerized Medical Imaging and Graphics*, 32(6):502–512, 2008. ISSN 0895-6111. doi: https://doi.org/10.1016/j.compmedimag.2008.05.004.

Gaëtan Texier, Rodrigue S. Allodji, Loty Diop, Jean-Baptiste Meynard, Liliane Pellegrin, and Hervé Chaudet. Using decision fusion methods to improve outbreak detection in disease surveillance. *BMC Medical Informatics and Decision Making*, 19(1):38, Mar 2019. doi: 10.1186/s12911-019-0774-3.

Rina Tse, Nisar Ahmed, and Mark Campbell. Unified terrain mapping model with markov random fields. *Robotics, IEEE Transactions on*, 31:290–306, 04 2015. doi: 10.1109/TRO.2015.2400654.

Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov Random Field Modeling, Inference & Learning in Computer Vision & Image Understanding: A Survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013. doi: 10.1016/j.cviu.2013.07.004.

Chi-Feng Wang. The Vanishing Gradient Problem — towardsdatascience.com. `https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484`, 2019. [Accessed 05-Jan-2023].

Chunhui Wang, Qianqian Zhu, Zhenyu Shan, Yingjie Xia, and Yuncai Liu. Fusing heterogeneous traffic data by kalman filters and gaussian mixture models. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 276–281, 2014. doi: 10.1109/ITSC.2014.6957704.

Yupeng Wei, Dazhong Wu, and Janis Terpenny. Decision-level data fusion in quality control and predictive maintenance. *IEEE Transactions on Automation Science and Engineering*, 18(1):184–194, 2021. doi: 10.1109/TASE.2020.2964998.

Junjia Yang, Shijun Wang, Xuezhu Na, and Yang Yang. A weighted data fusion method in distributed multi-sensors measurement and control system. In *2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, pages 654–658, 2019. doi: 10.1109/ICISCE48695.2019.00135.

Han Zhang and Jennifer Pan. Casm: A deep-learning approach for identifying collective action events with text and image data from social media. *Sociological Methodology*, 49:008117501986024, 07 2019. doi: 10.1177/0081175019860244.

Pan Zhang, Lanlan Rui, Xuesong Qiu, and Ruichang Shi. A new fusion structure model for real-time urban traffic state estimation by multisource traffic data fusion. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–6, 2016. doi: 10.1109/APNOMS.2016.7737227.

Zhenhua Zhang, Qing He, Jing Gao, and Ming Ni. A deep learning approach for detecting traffic accidents from social media data. *Transportation Research Part C: Emerging Technologies*, 86:580–596, 2018. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2017.11.027.

Yize Zhao, Changgee Chang, Margaret Hannum, Jasme Lee, and Ronglai Shen. Bayesian network-driven clustering analysis with feature selection for high-dimensional multi-modal molecular data. *Scientific Reports*, 11, 03 2021. doi: 10.1038/s41598-021-84514-0.

Lin Zhu, Fangce Guo, John Polak, and Rajesh Krishnan. Multisensor fusion based on data from bus gps, mobile phone, and loop detectors in travel time estimation. *Computing*, 01 2017.