Learning based event model for knowledge extraction and prediction system in the context of Smart City

Olivera Kotevska

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Olivera KOTEVSKA

Thèse dirigée par **Ahmed LBATH**

préparée au sein du **Laboratoire Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Un modèle de gestion d'évènements base sur l'apprentissage pour un système d'extraction et de prédiction dans le contexte de Ville Intelligente

Learning based event model for knowledge extraction and prediction system in the context of Smart City

Thèse soutenue publiquement le **30 janvier 2018**,
devant le jury composé de :

Monsieur AHMED LBATH
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Directeur de thèse
Monsieur KOKOU YETONGNON
PROFESSEUR, UNIVERSITE DE BOURGOGNE, Rapporteur
Monsieur YACINE OUZROUT
PROFESSEUR, UNIVERSITE LYON 2, Rapporteur
Monsieur ABDELLA BATTOU
CHEF DE DIVISION, NIST - ETATS-UNIS, Examinateur
Monsieur HERVE MARTIN
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Président
Monsieur ABDELTAWB M. HENDAWI
CHERCHEUR, UNIVERSITE DE VIRGINIE ETATS-UNIS, Examinateur
Monsieur DANIEL SAMAROV
CHERCHEUR, NIST - ETATS-UNIS, Examinateur
Monsieur SAID OUALIBOUSH
PRESIDENT DIRECTEUR GENERAL, SOCIETE REVARTIS A NEUCHATEL - SUISSE, Examinateur

Contents

Abstract in French

Des milliards de «choses» connectées à l'internet constituent les réseaux symbiotiques de périphériques de communication (par exemple, les téléphones, les tablettes, les ordinateurs portables), les appareils intelligents, les objets (par exemple, la maison intelligente, le réfrigérateur, etc.) et des réseaux de personnes comme les réseaux sociaux. La notion de réseaux traditionnels se développe et, à l'avenir, elle ira au-delà, y compris plus d'entités et d'informations. Ces réseaux et ces dispositifs détectent, surveillent et génèrent constamment une grande quantité de données sur tous les aspects de la vie humaine. L'un des principaux défis dans ce domaine est que le réseau se compose de «choses» qui sont hétérogènes à bien des égards, les deux autres, c'est qu'ils changent au fil du temps, et il y a tellement d'entités dans le réseau qui sont essentielles pour identifier le lien entre eux.

Dans cette recherche, nous abordons ces problèmes en combinant la théorie et les algorithmes du traitement des événements avec les domaines d'apprentissage par machine. Notre objectif est de proposer une solution possible pour mieux utiliser les informations générées par ces réseaux. Cela aidera à créer des systèmes qui détectent et répondent rapidement aux situations qui se produisent dans la vie urbaine afin qu'une décision intelligente puisse être prise pour les citoyens, les organisations, les entreprises et les administrations municipales.

Les médias sociaux sont considérés comme une source d'information sur les situations et les faits liés aux utilisateurs et à leur environnement social. Au début, nous abordons le problème de l'identification de l'opinion publique pour une période donnée (année, mois) afin de mieux comprendre la dynamique de la ville. Pour résoudre ce problème, nous avons proposé un nouvel algorithme pour analyser des données textuelles complexes et bruyantes telles que Twitter-messages-tweets. Cet algorithme permet de catégoriser automatiquement et d'identifier la similarité entre les sujets d'événement en utilisant les techniques de regroupement.

Le deuxième défi est de combiner les données du réseau avec diverses propriétés et caractéristiques en format commun qui faciliteront le partage des données entre les services. Pour le résoudre, nous avons créé un modèle d'événement commun qui réduit la complexité de la représentation tout en conservant la quantité maximale d'informations. Ce modèle comporte deux ajouts majeurs : la sémantiques et l'évolutivité. La partie sémantique signifie que notre modèle est souligné avec une ontologie de niveau supérieur qui ajoute des capacités d'interopérabilité. Bien que la partie d'évolutivité signifie que la structure du modèle proposé est flexible, ce qui ajoute des fonctionnalités d'extensibilité. Nous avons validé ce modèle en utilisant des modèles d'événements complexes et des techniques d'analyse prédictive.

Pour faire face à l'environnement dynamique et aux changements inattendus, nous avons créé un modèle de réseau dynamique et résilient. Il choisit toujours le modèle optimal pour les analyses et s'adapte automatiquement aux modifications en sélectionnant le meilleur modèle. Nous avons utilisé une approche qualitative et quantitative pour une sélection évolutive de flux d'événements, qui réduit la solution pour l'analyse des liens, l'optimale et l'alternative du meilleur modèle.

Par conséquent, nous avons mis en œuvre ces techniques dans FNEDAP (Framework for Network Event Detection Analysis and Prediction), un outil d'analyse développé au cours de cette thèse. Il est conçu pour être capable d'analyser les données complexes provenant de diverses sources et types afin de fournir une analyse proactive et prédictive. Il propose également une analyse efficace des relations entre les flux de données comme la corrélation, la

causalité, la similitude pour identifier les sources de données pertinentes qui peuvent servir de source de données alternative ou compléter le processus d'analyse. Les techniques de visualisation sont utilisées pour faciliter le processus décisionnel.

Nous évaluons les avantages de l'outil proposé sur différents scénarios d'applications de villes intelligentes impliquant des réseaux complexes comme le trafic, la criminalité et les réseaux sociaux. Les données utilisées dans ces expériences sont basées sur des données du monde réel recueillies à partir de Montgomery Country-Maryland. En répondant aux exigences des scénarios réels, le prototype démontre la validité et la faisabilité de l'outil.

Abstract in English

Billions of “things” connected to the Internet constitute the symbiotic networks of communication devices (e.g., phones, tablets, and laptops), smart appliances, objects (e.g., smart home, fridge and so forth) and networks of people (e.g., social networks). So, the concept of traditional networks (e.g., computer networks) is expanding and in future will go beyond it, including more entities and information. These networks and devices are constantly sensing, monitoring and generating a vast amount of data on all aspects of human life. One of the main challenges in this area is that the network consists of “things” which are heterogeneous in many ways, the other is that their state of the interconnected objects is changing over time, and there are so many entities in the network which is crucial to identify their interdependency in order to better monitor and predict the network behavior.

In this research, we address these problems by combining the theory and algorithms of event processing with machine learning domains. Our goal is to propose a possible solution to better use the information generated by these networks. It will help to create systems that detect and respond promptly to situations occurring in urban life so that smart decision can be made for citizens, organizations, companies and city administrations.

Social media is treated as a source of information about situations and facts related to the users and their social environment. At first, we tackle the problem of identifying the public opinion for a given period (year, month) to get a better understanding of city dynamics. To solve this problem, we proposed a new algorithm to analyze complex and noisy textual data such as Twitter messages-tweets. This algorithm permits an automatic categorization and similarity identification between event topics by using clustering techniques.

The second challenge is combing network data with various properties and characteristics in common format that will facilitate data sharing among services. To solve it we created common event model that reduces the representation complexity while keeping the maximum amount of information. This model has two major additions: semantic and scalability. The semantic part means that our model is underlined with an upper-level ontology that adds interoperability capabilities. While the scalability part means that the structure of the proposed model is flexible in adding new entries and features. We validated this model by using complex event patterns and predictive analytics techniques.

To deal with the dynamic environment and unexpected changes we created dynamic, resilient network model. It always chooses the optimal model for analytics and automatically adapts to the changes by selecting the next best model. We used qualitative and quantitative approach for scalable event stream selection, that narrows down the solution for link analysis, optimal and alternative best model.

Therefore, we have designed a Framework for Network Event Detection Analysis and Prediction (FNEDAP), an analysis tool developed during this dissertation where we implemented these techniques. It is designed to analyze complex data from various sources and types and to provide proactive, predictive analysis. It also proposes afficient relationship

analysis between data strams such as correlation, causality, similarity to identify relevant data sources that can act as an alternative data source or complement the analytics process. Visualization techniques are used to help the decision-making process.

Different experimentations ware performed in order to evaluate the benefits of the proposed tool over different smart city application involving complex networks such as traffic, crime, and social media. Performance metrics for measuring the accuracy of the models are based on minimum prediction error and confusion matrix. The data used in these experiments is based on real-world data collected from the Montgomery County, Maryland. By addressing the requirements in real-world scenarios, the prototype demonstrates the validity and feasibility of the tool.

Keywords

Event Processing, Analysis, and mining of complex data, Knowledge extraction, and representation, Machine learning, Dynamic model, Smart City.

Acknowledgments

At first, I would like to thank my supervisor Prof. Lbath for accepting me as a Ph.D. student, and for all his advises during my program. I gratefully acknowledge the funding source that made my Ph.D. work possible. I would like to thank the National Institute of Standards and Technology and Dr. Abdella Battou that provided the necessary financial support of this work. My sincere appreciation is extended to Dr. Gilad Kusne, Dr. Samarov, and Dr. Gelernter, for their feedback, discussions and give me the privilege to collaborate with them.

Special thanks to my friends and coworkers for their emotional support, valuable friendship and for helping me get through the difficult times. I really enjoyed the discussions that we had and the time that we spent together while we were working in the laboratory or enjoying a cup of coffee somewhere outside the research campus.

I would also like to extend my most profound recognition to the people that help me with editing the dissertation; they are Tara Brown, Katie Rapp, Kathryn Miller, and Stephen Nightingale.

Additionally, I wish to thank my family for all the moral support they have given me over those years.

List of tables

List of figures

List of equations

Abbreviations

AL	Architectural Layers
AR	Autoregression
ARIMA	Autoregressive Integrated Moving Average
BN	Bayesian Networks
CA	Combined Architectures
CEP	Complex Event Processing
CPD	Conditional Probability Distributions
DAG	Directed Acyclic Graph
DTN	Delay Tolerant Networks
ED	Event Driven
EP	Event Processing
IoT	Internet of Things
LAN	Local Area Networks
ML	Machine Learning
MN	Markov Networks
MRF	Markov Random Fields
NB	Naive Bayes
NBD-PWD	NIST Big Data Public Working Group
NSF	National Science Foundation
NIST	National Institute of Standards and Technology
PA	Predictive analytics
PGM	Probabilistic Grpahical Model
PR	Poisson Regression
QoI	Quality of Information
RF	Random Forest
SC	Smart City
SOA	Service Oriented Architectures
SVM	Support Vector Machines
UAN	Urban Automation Networks
U.S.	United States
VAR	Vector Autoregression

Chapter 1

Introduction

"Measuring science is about determining what data to capture and under what conditions so that we can make sense of it. "
– NIST

1.1 Overview

Today's cities (and the vision of the future city) represent an environment of millions of connected and distributed devices that envision a complex network of interlinked people, processes, sensors, and data, created by social and sensor networks, or by process management. The real-world events are now being observed by multiple networked streams, where each is complementing the other with his or her characteristics, features, and perspectives. Many of these networked data streams are becoming digitized, and some are available in public formats and available for sense-making. Over time, the number of these distributed networks increase which produces by increasing the volume and variety of generated data, and this trend will continue. Because of the increased flow of data, information systems are frequently required to deal with vast amounts of data that can be in various structures from heterogeneous sources and type. In some cases, it contains raw and noisy data to build high-level abstractions to be later analyzed and organized for delivering useful functionalities to end consumers.

The networked data streams provide an opportunity for their link identification, similarity, and time dynamics to recognize the evolving patterns in the inter-intra-city. The information delivered can help us to understand better how cities work and detect events and patterns that can help to remediate a broad range of issues affecting the everyday lives of citizens and the efficiency of the city. Providing the tools that can make this process easy and accessible to city stakeholders has the potential to improve traffic, event management, disaster management systems, health monitoring systems, air quality, and city planning.

However, there has been progressing in the field of generating action and situation recognition from various data streams [7] [112] [119] [153], but there are still open issues. This dissertation addresses and tackles some of the critical challenges related to making sense of collected data resiliently. We focus on the issues related to detecting event types from unstructured data, especially Twitter. Finding similarity between topics, creating the semantic scalable event model structure that provides unification of complex data from various data

sources and types. Relationship analysis between data streams to identifying alternative data sources that can be used in case of data loss, creating a dynamic network model that adapts to environmental data stream changes like adding new data stream (source) or removing, as well as recognizing the right parameters for the optimal prediction model.

We motivate this work based on the smart city applications and real-world types of problems that are becoming increasingly relevant. We have case studies for improving crime prediction, identifying safe zip zones for pedestrians and using social media knowledge to improve local city services.

1.2 Motivating example

Real-world Events and their Multimodal Appearances

Real-world events are observed by multiple observers including machine sensors and citizens' sensors. For example, there may be sensors monitoring a road for some vehicles passing over the road and people observing events in the city and reporting them on social media. As shown in Figure 1, the intelligent system connects and collects a variety of data streams related to multiple human functions like weather, video car speed, traffic lights, health symptoms, and social media networks. The real-time streams originate either from the traditional sensor device based sources, such as weather, traffic sensors, satellite images, or the increasingly common social reporting mechanisms, such as Twitter, Foursquare updates. The event recognition system accepts input streams and continually seeks to identify shared patterns. The output is a stream or multiple streams of recognized patterns, prediction and forecasting for future behavior.

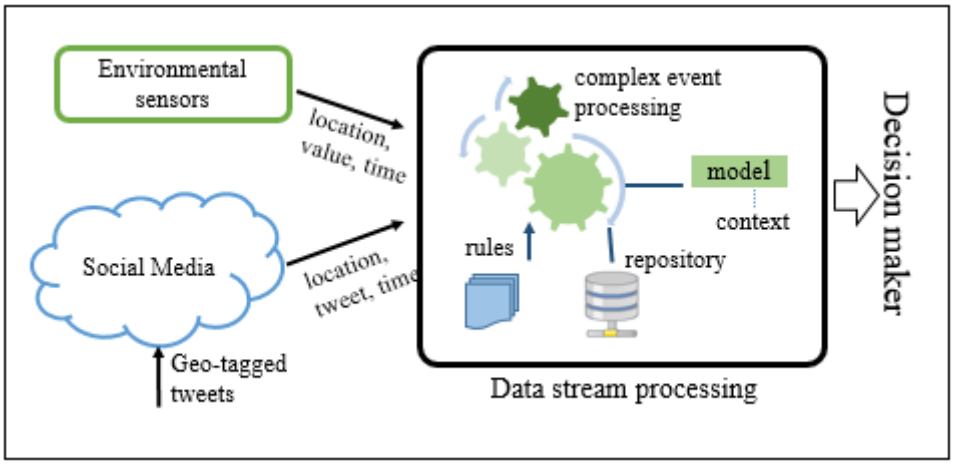


Figure 1: High-level view (in abstract form) of an event processing system

The nature of the physical world can be hard to understand by just observing it piecemeal. Social sensors can play multiple roles, they are the humanized interpretation and can describe different aspects of a situation, many of which are not yet measurable by any hardware sensors. Millions of users are already active on those social networks and are expressing their thoughts for different actions on a daily basis.

Most real-world events exhibit close interactions between physical, cyber, and the social worlds as illustrated in Figure 1, by traffic analytics, health, wellbeing applications, and power grid maintenance. The models of data allow us to infer theories of the physical world through observations.

In this dissertation, we try to infer the models of the physical world from observational data. We firmly believe that accurate models need to be complemented with data from the real-world for a realistic understanding of the physical world. We describe a general use case as defined by the NIST template in Appendix C. The main content of the construction of smart and connected community applications is Smart Public Service and Construction of Social Management. This entails collecting and analyzing data in urban areas, providing more accurate service to the city's decision-making processes. We experiment with a number of studies, each with a different focus.

Let us consider one typical use case scenario, illustrated in Figure 2, that of analyzing the places that people visit. We use police records of traffic incidents related to pedestrian safety, community events at specific areas and weather. We can find the relations between places characterized by shopping and restaurant events and increased number of traffic incidents involving pedestrians. However, we have not found significant change between these type of events in different seasons or weather changes.

These observations contain valuable nuggets of information for decision makers such as city authorities and planners, doctors, and patients. For example, if city authorities know the reason for slow-moving traffic, they can mobilize appropriate units to mitigate the problems and reduce the impact of people's mobility. Alternatively, if they know crime trends at particular locations in future, they can allocate more resources to those areas.

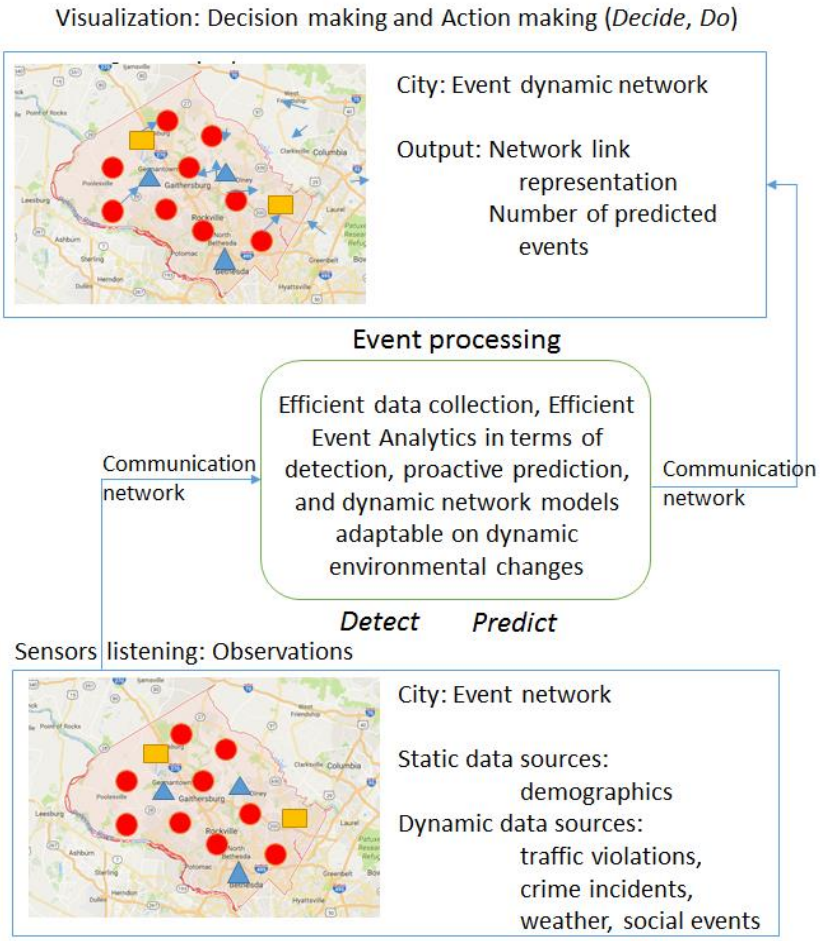


Figure 2: Use case scenario for predicting future event traffic and crime events

Atomistic approach to analytics that can help decision-makers to transform observations into actions is the one proposed by Etzion et al. [48], it is following the cycle of Detect, Predict, Decide, Do. Detect the events, event types or complex events of the physical environment. Predict future event occurrence, when is the likelihood of occurrence. Decide using a model of action for concrete event situation, and Do through recommendation engine or another form of proactive actions. Like, this approach is the one proposed by Boyd [80] called OODA (Observe, Orient, Decide, and Act) loop, he created this method to help an individual or an organization for making intelligent decisions. In this work, we made contributions to Detect and Predict steps, while in Decide step we provide a visualization that can help policymakers, and in Do step, we do not make any contribution that is for the future fork.

Therefore, understanding real-world events in the physical world utilizing observational data is a challenging problem. We highlight some of the challenges in understanding real-world events utilizing observational data.

i) *Heterogeneity*: Presented scenario has heterogeneity in observations of a single event manifesting in multiple modalities requiring techniques to integrate and process different comments.
ii) *Qualitative vs. Quantitative*: Sensor data is qualitative providing precise view of a quantity of interest, e.g., 50 F. Data from people are usually qualitative providing a high-level description of an event, e.g., cold weather.
iii) *Incompleteness*: Application processing events that have to arrive from sources such as sensors and social media that have inherent incompleteness and uncertainties associated with them.
iv) *Dynamic environmental changes*: Datastream failures can occur for various reasons; also, the new data stream can be added at any time.

This dissertation demonstrates the benefit of using event processing and predictive analysis in dealing with some of these challenges. We present techniques to integrate multimodal observations such as numerical sensor data and textual, social data to address the challenges of incompleteness and heterogeneity. We experiment with probabilistic and count model to deal with uncertainty and accurate event prediction. We used similarity metrics and graphical structure to model interactions and relationships between real-world incidents. We formulate time series based models to capture the dynamism of real-world events. We illustrate the characteristics of the observations related to smart city events by considering the examples in the real world.

1.3 Research challenges and methodology

Our goal is to fill the gaps derived in the related work and provide solutions in some of the problems arising from the application domain.

Research challenges

We determine the challenges listed below, separated in three categories: scientific challenges, engineering challenges (R&D) and challenges related to standards.

I. Scientific challenges

1) How to extract knowledge from collected data from sensors (physical and social)?
   - How to efficiently preprocess the text data streams from social sensors?
   - How to extract the knowledge from text data streams?
   - How to identify trends based on extracted knowledge?
   - How to graphically present the event trends for decision makers from city representatives?

To solve this problem, we need to proceed with different steps, and each step is related to existing work in the literature. We begin each phase by making a survey of existing work, comparing them, and choosing the more appropriate in respect of precision and performance measures. We create a global approach to addressing this problem and develop a fully automated algorithm for extracting knowledge from social sensors.

2) How to frame complex data streams from different data sources and types?
   - What is schema structure or event model is more appropriate for event streams?
   - What is the most suitable model to represent events?
   - What event model to propose to tackle this problem?
   - How to automatically identify relevant data streams?
   - How to integrate incomplete event streams?
   - How to consider event semantics analysis?
   - How to handle scalability in the event model regarding event attributes and event data streams?

   To solve this problem, we start with a literature survey and make a comparison between existing solutions for existing event models. We go on to create our own event model that supports scalability, uncertainty, semantics, and automatic event identification.

3) How to create a predictive model based on knowledge?
   - How to efficiently identify relationship links between data streams?
   - What is an appropriate prediction model?
   - How to adapt the prediction model?
   - How to integrate scalability for the model to choose?
   - How to graphically present the network dynamics and relationship between data streams for decision makers from city representatives?

   To solve this problem we choose two empirical research approaches, one is quantitative using statistical methods, and the other is qualitative using multi-dimensional visualization method. We presented the advantages and disadvantages of using each of these.

II. R & D challenges

1) What are the challenges regarding Smart City applications/use cases?
2) What is the appropriate use case within the Smart City application domain to validate our experimentation?

3) What framework to design and develop that integrates event detection and prediction models?

III. Standards challenges

1) How to assess the proposed systems regarding accuracy, performance and other metrics?
2) Can we comply with ongoing NIST standard for Big Data Use Cases and Smart City use cases?

Methodology

The methodology we follow consists of the following steps:

1) Review the literature and related work of the relevant research areas like event processing detection and event data models, prediction methods for multisensory learning, and Smart City applications. Understand their advantages, limitations, and applicability. Define a problem of multisensory event detection and data stream fusion.

2) Analyze the challenges that need to be answered and formulate the main research questions of knowledge extraction, the efficient data format for multisensory data fusion, and predictive modeling based on knowledge.

3) Create an abstract design for the problem formulation, proposed development solution steps for testing. In this phase for different problems as explained previously, we choose different research approaches like comparative, qualitative, and quantitative.

4) Develop a prototype as a proof of concept implementation for the particular research question.

5) Finally, evaluate the solutions using relevant performance metrics on real-world data sets with the regards to application specifics.

6) Assess the disadvantages and advantages of the proposed approach.

7) Define and design how specific solutions fit together in one framework.

1.4 Contributions

Following the research methodology, the research presented in this dissertation results in contributions in event modeling and prediction, implementation and adaptation to provide an easy-to-use, on-demand event processing capability in application domains such as a Smart City. Per the research questions, we summarize in this section all input of this dissertation. They cover a research challenge and integration framework to give the big picture description of this dissertation.

- Fully automated event processing system and event detection algorithm
  - Table that shows advantages and disadvantages of the most used event detection algorithms
  - Demonstrated social network sensing model that will feed in addition to sentiment context and provide visualization that helps decision making city representatives

- Generic approach to incorporate different data streams
  - Reviewing the most used event definitions
  - Comparison between various models for data fusion
  - Generic approach integrating several ontologies for event semantic analysis
  - incompleteness

- Multisensory predictive model adaptable to data stream changes
  - Capable of considering based on the data coming to choose the best model based on context and link analysis
  - Adaptable to data stream changes
  - Evaluate the theoretical prediction models based on performance metrics and data characteristics

- Experimental evaluation and validation using real-world data sets (traffic, crime, weather, demographics, distance in miles, community events, microblogs) on several use cases analysis, applicability to network assessment, congestion
  - Identify, design, develop and demonstrate use cases for decision makers for city representatives

- Identification of the key metrics and measurements dedicated to the event detection, multisensory analytical methods, and network assessment

- Visualization method to show the effectiveness of designed methods and visualization of complex data promptly

- A formal approach implemented within a framework for experimentation
- Formulating the use cases using existing NIST standards

Nevertheless, note that each one of the contributions presented is valuable by themselves and can be utilized separately from the others. Therefore, either by considering these contributions in isolation or together, this research significantly advances the event processing state of the art and provides tools and methodologies that can be applied in the context of event processing research and development.

1.5 Organization

The design of a smart and sustainable city is faced with various challenging issues such as event management, reliable systems and efficient decision making. In this dissertation, we aim to study these issues using data analytics approaches. The rest of this dissertation is dedicated to the above issues, and organized as follows:

Chapter 2 presents the theoretical concepts for the event processing and prediction methods deployed, and also the challenges related to data analytics in the domain of Smart Cities.

Chapter 3 surveys state of the art in the fields of event detection, data event models and prediction methods with the focus on smart city application. We show the difficulties of old solutions compared with ours.

Chapter 4 presents the proposed solutions for automatic event processing, semantic scalable event model, and dynamically adaptable network model.

Chapter 5 presents and describe the data used to validate our solutions, and demonstrates the experiments and evaluation for each use case, with the discussion of the obtained results.

Chapter 6 gives conclusions and discusses a variety of possible directions for future work. We envision how the models developed in this dissertation have potential to be used in critical applications such as federated cloud, and we discuss the role these models might play as the number of sensors become denser in the future.

We also have Appendix section that presents selected event definitions (Appendix A), the comparison between event models based on different criteria (Appendix B), use case standard form based NIST Big Data Framework (Appendix C), and details about the development environment and packages.

In conclusion, in this chapter, we present the idea and motivation for the work developed in the rest of the dissertation. In the following, we describe the theoretical concepts that were used and the latest research achievements in those areas.

Chapter 2

Background

"It is the theory that decides what can be observed."
- Albert Einstein

Summary. This chapter introduces some of the theory that is fundamental to the work described in this dissertation. We discuss concepts relevant to Event Processing and Analytical Techniques for multisensory learning, as well as the context of Smart and Connected Communities applications which we used as a use case and for experimentation. The definition of an event will be explained first. Then, the theoretical concepts for event processing such as detection will be discussed. Finally, complex event patterns, analytical methods for multisensory event learning and requirements characteristics of smart city case studies are presented.

2.1. Event processing

2.1.1. Definition of Event

The notion of "events" is broadly understood across different research fields (computational linguistics, artificial intelligence, information retrieval, information extraction, automatic summarization, natural language processing). There are many definitions for an event in various disciplines; some of the most commonly used definitions in literature are presented in Appendix A. For a concise explanation we selected a highly-cited definition of "event" by Etzion et al. [47]:

> "An *event* is an occurrence within a particular system or domain; it is something that has happened, or is contemplated as having occurred in that field. The word *event* is also used to mean a programming entity that represents such an occurrence in a computing system."

This definition presents two meanings of events, the first one refers to something that happens in the real world or some other defined system, while the other one refers to programming entity.

Real world events are events that happen in our daily lives such as, phone call, ordering food, or a bus arriving at a bus stop. In addition, events can also be simulations occurring in a

defined system or virtual reality environment. In the context of a programming entity, events take the form of a database transaction, transmitting messages between systems or the structure of the programming language.

This dissertation is the primary concern with computing the corresponding events generated and described from daily living. For example, a real world everyday life scenario computing to work in an electric car. One event is when the electric car detects that it needs to be charged. A second event is when it calculates its travel route. A third event could be when a security camera detects construction on the road, informing the car's travel controls of a route change. Traffic sensors noting the car and changing their timing is another event. Figure 3 shows a visual representation of this use case.

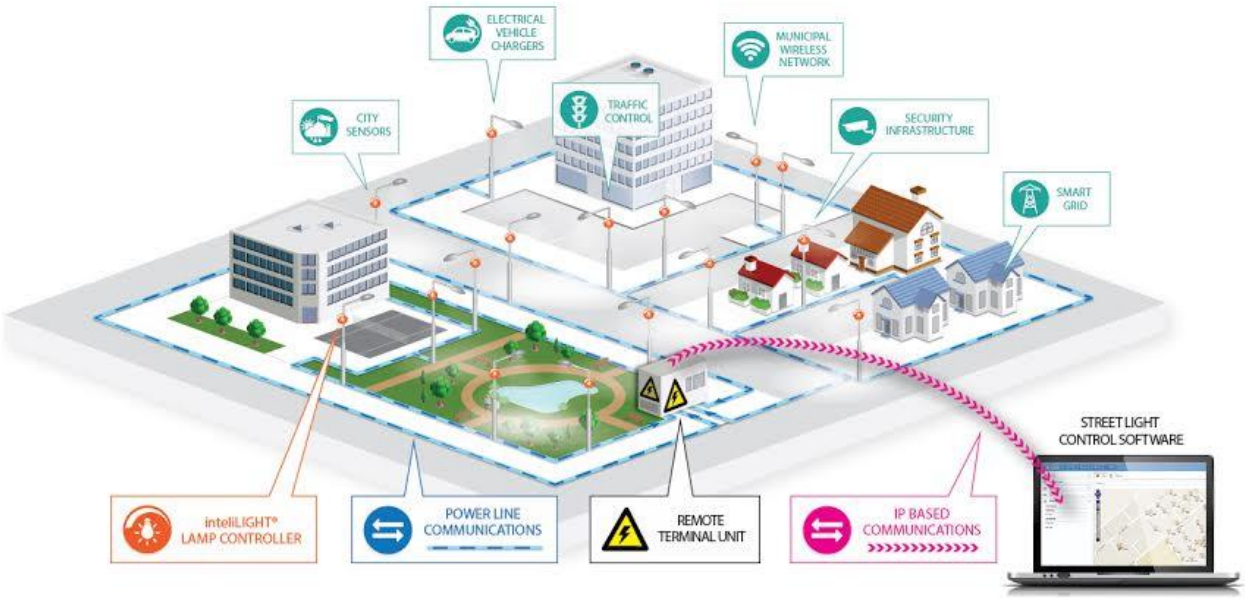


Figure 3: Real word event generated in daily life [1]

Events are classified by devices, context, and categories. Low-level events (LLE) come from devices such as GPS, accelerometers, internal thermometers, microphones, and internal cameras. High-level events (HLE) are produced by context: for example, punctuality, passenger and driver safety, passenger and driver comfort, passenger satisfaction, traffic congestions, and so forth [47]. Categories can also be events as burglary incident detected on social media and a thievery incident reported by police officials. Also, the single event occurrence can be represented by other entities, and a given event object might capture only some of the facts of an event occurrence [47].

High-level events are of most considerable interest for this dissertation. They can appear in multiple formats originated by sensor networks, and test streams from social and web media. These events may be occurrences across the various layers of the system. Alternatively, they

[1] https://www.ipi-singapore.org/technology-offers/street-lighting-management-system

may be news items, text messages, social media posts, stock market feeds, traffic reports, weather reports, or other kinds of data.

2.1.2. Definition of Event Processing and Complex Event Processing

The event is the key concept of event processing as we defined and explained previously. Luckham et al. [91] define event processing (EP) as :

"*A method of tracking and analyzing the streams of information about the things that happen* ."

EP is a research discipline with many antecedents, including active databases [57], temporal databases, data stream management systems, inference rules, discrete event simulation and distributed computing [48].

A common characteristic of event processing applications is to continuously receive events from different event sources, such as sensors, social media, mobile devices, and so forth. Examples of application areas include social media monitoring [93], traffic control [149], [136] or environment monitoring using wireless sensor network [150]. Some earlier research projects on this topic include Rapide in Stanford [92], Infospheres in Cal Tech [8], Apama in Cambridge University [33] Amit in IBM Haifa Research Lab [18], and a few streaming projects such as Stanford Stream project [16] and Aurora [28].

Complex Event Processing (CEP) is event processing build out of lots of event instance from multiple sources to infer events or patterns that suggest more complicated circumstances. The goal of CEP [91] is to identify meaningful events or situational knowledge from massive amounts of events and respond to them as quickly as possible or as close to real-time as possible, see Figure 4. The primary role of a CEP is to detect the occurrence of an activity pattern on the incoming streams of data. CEP is used to deal with patterns among events and process large volumes of messages with little latency. CEP may detect the logical and statistical relationship contained in the event stream by matching the pattern.

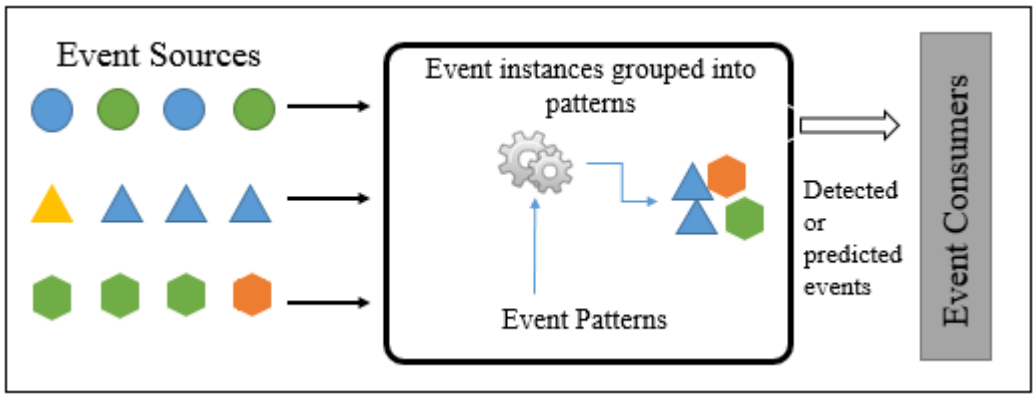


Figure 4: High-level overview of event processing

CEP technology provides new solutions to the field of multiple pattern identifications and real-time data processing, which can be used for improving the performances in smart and connected environments. The efficient processing of events is fundamental to the quality-of-service requirements of event processing systems [35].

2.1.3. Event detection

Some events can be observed very quickly, for example, things we see and hear during our daily activities. Some require us to do something first, for example, subscribing to newsgroups, or reading a newspaper [24]. In other cases, we need to do some work to detect the event that happened, as all we can observe are its effects.

The first scenario as presented in Figure 5 can be summarized as follows: When a Receiver detects a new event, it is sent to the CEP engine. The CEP engine can perform some preprocessing, send information to the next processing phase, and combine events from multiple sources. Per Cupola et al. [41], a CEP engine is divided into two components: Decider and Producer. They treat the events per predefined rules. Rules define the condition, and Decider checks it at the beginning of events. After the event is detected, it is sent to Producer which generated the corresponding action, such as notification, alarm or a new compound event.

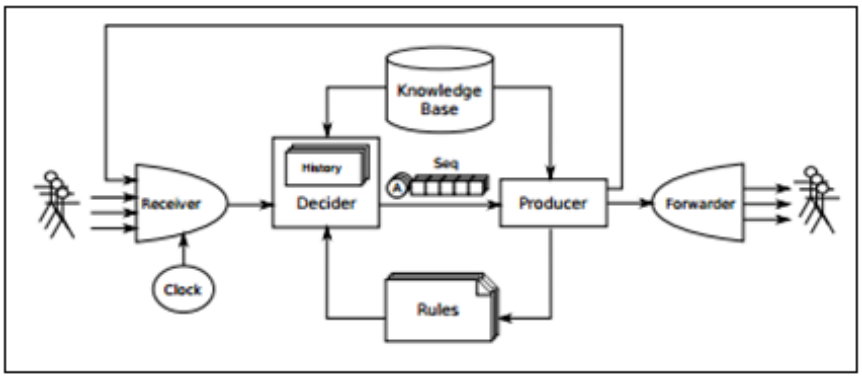


Figure 5: The functional requirement of a CEP system [41]

The first challenge is to detect and apply the events that are relevant to the decisions for the problems that we are trying to solve. There are a few techniques for event detection: hand-written rules, sequence models which are usually used in a text processing domain such as natural language processing and information retrieval, and machine learning classification methods like Naïve Bayes algorithm.

CEP supports a few detection paradigms, like using deterministic finite automata (Ode), Petri-nets (SAMOS), logical rules (ETALIS) or graphs (Apache Storm, Apache Flink, Apache Samza). Ode and SAMOS support the semantics of event operators. ETALIS is a logic programming system that uses background knowledge for reasoning. There are also many

software vendors like Esper, Tibco, StreamBase, Coral8/Aleri, Progress Apama that provide a development environment, with the support of different event processing languages like SQL, XML or vendor specific languages.

2.1.4. Event design patterns

Detecting complex patterns of events, consist of events that are widely distributed in time and location of occurrence. Pattern detection is one of the essential functions of event processing; it is a combination of the role within the context. As an example, in transportation, CEP is used to track the individual events and trigger some actions when an exception is found. CEP systems usually use Event-Condition-Action (ECA) rules for event processing, and event algebra expression for its construction, parameters, and monitoring intervals. Composite events are defined using logic operators.

Design patterns represent generic solutions for particular data streams problems. Alves et al. [9] presented ten basic design patterns: filtering, aggregation, correlation, joins, time-based patterns of events across multiple streams, hierarchical events (processing, analyzing, composing), in-memory, database lookups, database writes, and dynamic queries. Furthermore, for efficient event processing, individual event patterns need to be applied, such as event filtering (based on type or context), partitioning, enrichment, aggregation, relationship, application time, missing event detection, modeling behavior, and hierarchical events [9], [104], [147]. Figure 6 represents sliding window pattern and relationship patterns, such as aggregation, correlation, and causality.

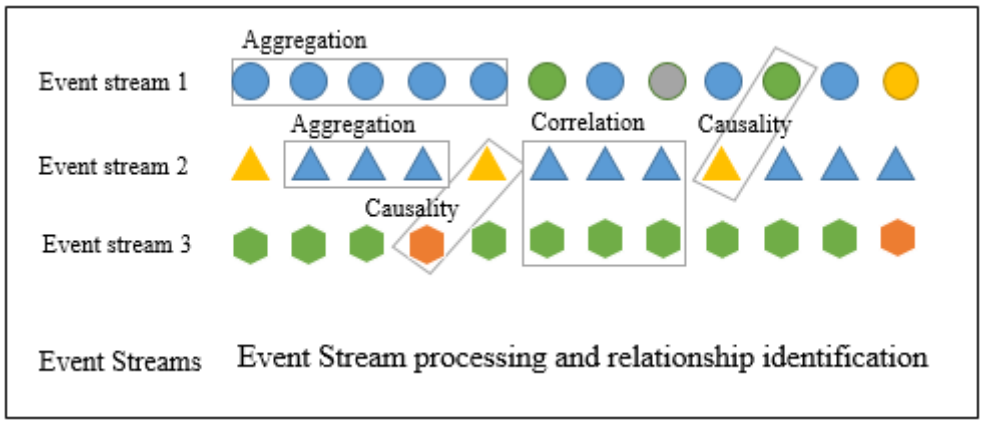


Figure 6: Representation of relationships between event streams

Events can be related because of the network topology or other factors which are not explicitly encoded in the event's data. A key to understanding events is knowing what caused them and having that knowledge at the time the events happen [91]. The event is flowing from one location and creating events to another. For instance, when the weather changes the

consistency in traffic also changs. The causal relationship between events can be both horizontal and vertical as illustrated in figure 7. Horizontal causality emphasizes that the caused and causing events happen on the same conceptual level. For example, snowing created a slippery road, which then caused a car accident. On the other hand, vertical causality is discovering relationship across layers. Between low and high strata, for instance, broken network link can produce incomplete results and missed event detection on the application layer. It is essential to trace causal relationships between events in real-time, both horizontally within a level of system activity and vertically between high and low levels of network activity.

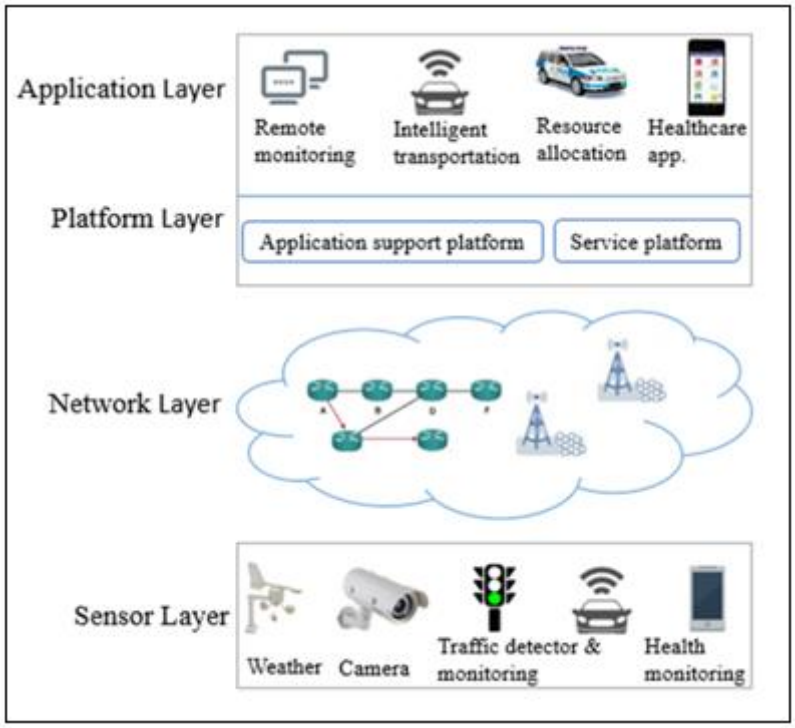


Figure 7: Typical layers in information system

Events occurring in distributed, heterogeneous sources and applications are linked together to form a so-called event cloud [151]. As we mentioned, an event can be related to other events by time, causality, and aggregation. By the use of CEP engines, low-level events can be aggregated in high-level events. CEP collects LLE metadata such as ID, location and time processing them to create new significant events called complex events and then forwarding them to the application layer. Complex events are the valuable new events arising after processing atomic events according to the specified rules [108][109]. A complex event can be achieved with known event patterns.

Event processing must process events coming from various sources, for example, sensors, social media, or the Internet. The quality of design patterns depends on errors in measurements, noise in the environment, and granularity of the observations [27]. Depending on the domain and application specifics different requirements need to be respected and

satisfied, such as low bandwidth, energy restrictions, a large volume of data, a variety of data types, velocity and dynamicity. For instance, in-network fusion techniques or dimension reduction techniques are used to overcome the problem of the large volume of transmitted data. To increase the latency and accuracy combination of data from multiple resources can be utilized, and more trustworthy data sources. Events can arrive in unexpected orders and timing; those are some of the challenges that design pattern engine needs to consider.

Event patterns are implemented using event pattern languages (EPL), like SQL, XML or a vendor-specific language. Paschke et al. [109] discuss in details the criteria for a successful event processing language design which allows an easier generation of new CEP applications. According to Luckham et al. [92], an EPL needs to meet the following properties:

- Power of expression: it must provide relational operations to describe the relationships between events
- Notational simplicity: It must have a simple notation to write patterns succinctly
- Precise semantics: It must give a mathematically precise concept of matching
- Scalable pattern matching: It must have an efficient pattern matcher in order to handle large amounts of events in real-time.

Examples of EPL include Rapide, Borealis, RuleCore, SASE+, Cayuga and RAPIDE-EPL, STRAW-EPL, StreamSQL and there are still ongoing research efforts.

CEP systems are usually evaluated for their performance regarding throughput, measured as how many events are processed per second and latency, the average time required to process an event. Less often, the memory footprint is reported. Standard benchmarks have not yet been established, although some work towards this direction has begun according to Mendes et al. [98] and Zámečníková et al. [158].

2.1.5. Uncertainty in event-based processing

As mentioned in subsection 2.1.4, heterogeneous events that arrive from various sources like sensors and social media have inherent uncertainties associated with them. The data streams have inherent risks associated with them, for instance, incomplete data flows, unreliable data sources, and networks. Having a mechanism for handling uncertainties gives higher confidence level to event processing as a base for the quality of data streams. Uncertainty can be in the input data source, change in the definition of events and event patterns. Authors from [55] create the taxonomy of event uncertainty where they categorize them in two dimensions': element and origin uncertainty. The first dimension refers to event occurrence and event attributes, while the second one refers to uncertainty associated with a feature or event source and uncertainty resulting from event inference. Another group of authors [14] defines the types of uncertainty that event processing systems have to handle, such as:

- Incomplete event streams: Such as failure to detect an event due to a power outage

- Insufficient event dictionary: Detection of some types requires the detection of some other activities
- Erroneous event recognition: The time of delivering the results, or in this case, a report, can produce mistaken event
- Inconsistent event annotation: The pattern recognition algorithms, rules defined by an expert, or training data used by machine learning algorithm
- Imprecise event pattern: In some case, it may not be possible to identify all conditions in which a particular event happens precisely.

Alternatively, Artikis et al. [14] summarized them in three groups:

- Uncertainty in the event input
- Certainty in the event input and uncertainty in the composite event pattern
- Uncertainty in both input and pattern.

Some of the existing approaches are logic-based models, probabilistic graphical models, and fuzzy set theory. Logic-based models are very expressive with formal declarative semantics; they directly exploit background knowledge and have trouble with uncertainty. Probabilistic graphical models can handle uncertainty, have a lack of a formal representation language, and are difficult to model complex events and to integrate background knowledge. Fuzzy set theory handles uncertainty by assigning uncertainties to the rules where detection is a reported asset structure. There are efforts to combine logic-based approaches that incorporate statistical methods like Statistical Relational Learning, and probabilistic approaches that learn logic-based models like Probabilistic Inductive Logic Programming. For implementation, the program ProbLog with Markov Logic Networks (MLN) algorithm can be used, or a combination of MLN with Markov Chain Monte Carlo, and event calculus in MLN.

Also, another approach is ontology-driven modeling to handle uncertainty [89] [72] [142]. Alternatively, per Skarlatidis et al. [135] event recognition techniques can handle uncertainty to some extent by using: automata-based, logic-based programs (MLNs, Bayesian networks, and Probabilistic logic), Petri-nets and context-free grammars. The same group of authors defines the operators that should be supported by a CEP engine. They are sequence, disjunction, iteration, negation, selection, production, and windowing. In addition, other functional characteristics such as support for background knowledge, probabilistic properties of each method (independence assumption, data uncertainty support, pattern uncertainty support, hard constraints), and inference capabilities can be used.

Primary areas of interest are using background knowledge to reason and using statistical knowledge to match and detect the pattern of interest, to identify relationships between them and make a prediction for a future event occurrence.

The next section gives a theoretical explanation about prediction algorithms that were used in this dissertation.

2.2. Analytical methods for multisensory learning

Event though there have been many studies on the processing of data streams [23], in recent years, with advancements in online technologies, data stream mining has attracted researchers' attention with a focus on classification and predictive analytics. The aim of predictive analytics is to anticipate the outcome of future events by answering the following three questions: What happened? What is going on? What will happen? Predictive analytics is comprised of predictive modeling, which aims to address the who, when, and why questions regarding the current behaviors, and forecasting, which concerns their future behavioral patterns. Predictive analytics is used by corporations to foresee trends in customer behavior, product usage, and the likelihood of purchases. However, predictive analytics is also being used for unconventional purposes, such as predicting traffic congestion, manufacturing supply chain problems, and the spread of infections. The essential function of predictive analytics technology is to identify patterns among raw, historical data via complex event processing to forecast and assess potential risk. However, there are a variety of analytic prediction methods that are used. Predictive models can be classified as univariate and multivariate. Univariate models are the simplest, and operating with one variable, while multivariate models predict outcomes of situations affected by more than one variable.

Predictive and forecasting modeling are classified as qualitative prediction and quantitative prediction. Qualitative is a subjective (human) judgment based on experience, expertise, and intuition. This approach is usually used when there is no available historical data, or for any reason, a mathematical model cannot be created. Some of these methods are Delphi method, Jury of Expert Opinion, Scenario Analysis, Sales Force Composite, and Market Survey. Quantitative methods are based on mathematical modeling. They can be classified as causal models which investigate how the forecasted variable is determined by factors, non-causal models that make predictions by extracting patterns from the past, and a combination of both causal and non-causal models. We will focus on quantitative methods considering causation and the importance of historical data points.

Predictive analytics is categorized as prediction and forecasting. The following sections (Bayesian network, Poisson process, and Time-series model) provide a detailed description for each of the algorithms.

2.2.1. Bayesian network

To model an event we would deal with a lot of random variables. Random variables are the variables whose value is not known until they are observed and whose domain is the set of core events. An event is an outcome or a union of results when the outcomes are the occurrences over which we can assign probabilities. Probabilistic Graphical Models (PGMs) provide a

framework for modeling a large number of random variables *(X_1, X_2, …, X_n)*. PGM uses a graph-theoretic representation *G(V, E)* where nodes correspond to random variables V and edges E correspond to relationships between them. When the edges are directed, they are known as Bayesian networks (BNs). PGMs with undirected edges are known as Markov networks (MNs) or Markov Random Fields (MRFs).

BN provide modeling of probability distributions over several variables X and using a directed graph G called Directed Acyclic Graph (DAG). Random variables represent nodes on the chart, while edges typically account for the dependency between variables represented as nodes. More precisely, various combinations of events represented as the value of random variables that are assigned a probability. The joint probability distribution *P (X_1, X_2, …, X_n)* overall random variables *(X_1, X_2, …, X_n)* can be represented as follows:

$$P(X_1, X_2, \dots, X_n) = \sum_{i=1}^{n} P(x_1 \mid Pa\,(x_i))$$

Equation 1: Joint probability distribution over random variables

The formula presents the joint distribution represented as a product of conditional distributions. BN is characterized by local models, independencies, and causality [51].

Conditional Probability Distributions (CPD) represent the local conditional distributions P(Xi|Pa(Xi)), given the value of the parent node Pa. Also, the BN graph implicitly encodes a set of conditional independence assumptions.

Each independence is of the form ($X_1 \perp X_2$), which lead to X_1 is independent of X_2. If P is a probability distribution with independencies I(P), then G is an I-map of P if I(G) ⊆ I(P). If P factorizes according to G then G is an I-map of P. This is the key property to allowing a compact representation, and crucial for understanding network behavior.

While a BN captures conditional independences in distribution, the causal structure is not necessarily meaningful, e.g., the directionality can even be intertemporal. In a BN structure, an edge X → Y means that X causes Y, directly or indirectly. BNs with causal structure is considerate to be more natural and sparser. While two graphs X → Y and Y → X are equivalent probabilistic models, they represent different causal models.

Take for example, if we need to observe various combinations of events represented as the value of random variables and assign a probability. Formally, it needs to specify the joint probability distribution defined by *P* (*A, B, C, D, E, F*). Each arrow indicates a dependency, in this case, D depends on A, D depends on B, E depends on C, F depends on E and D. A, B and C are independent, and D, E, and F have a conditional dependency. Also, A ⊥ B, B ⊥ C, A ⊥ C, and D ⊥ E.

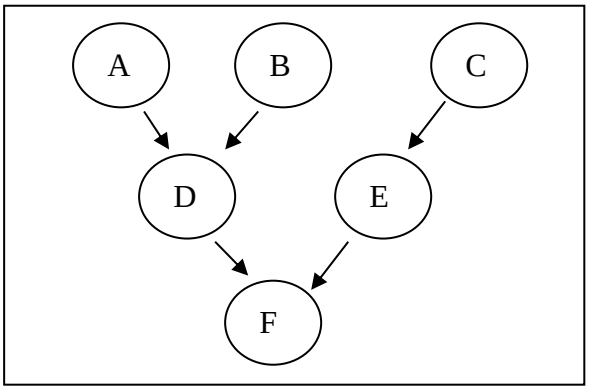


Figure 8: An example of a Bayesian network

The joint conditional function is

$$P(A, B, C, D, E, F) = P(A) * P(B) * P(C) * P(D \mid A, B) * P(E \mid C) * P(F \mid D, E)$$

Equation 2. Example of joint conditional function

Also, graphical models can make computation more straightforward and more intuitive. Computational properties of inference and learning can be determined by viewing the structure of the graph, (See Figure 8). For instance, to illustrate how the graphical model structure can be used to simplify inference, in Figure 9 the value of some of the variables are known, and the values of other variables are not known. Suppose we wish to compute the probability distribution of D given the evidence (i.e., P (D | A, B)). To calculate this directly from the joint distribution, we could use the law of total probability by summing over all the values of the remaining unobserved variables. Also, for a given reason, we can make a prediction using an appropriate combination of A, B, C, D, E, F.

A directed graph G = (E, N) assigns a contemporaneous causal flow to a set of variables based on a correlation relationship [86]. The relationship between each pair of variables characterizes the causal relationship between them. No edge (E) means independence between two, whereas an undirected edge variables (X - Y) signifies a correlation with no causation. Direct Edge means (X -> Y) means X causes Y, but X does not cause Y. And bidirected edge indicates bidirectional causality (X <-> Y).

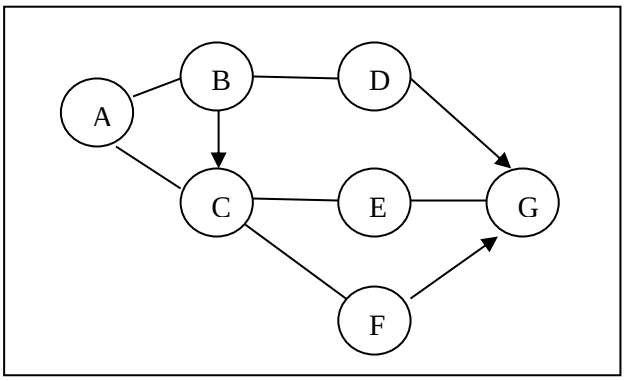


Figure 9: Mixed graph, directional and unidirectional nodes

Domain knowledge plays a significant role in specifying independence among various random variables resulting in a significant reduction in the number of parameters to be determined. Probabilistic graphical models utilize probability to deal with uncertainty, missing values, and structure to deal with complexity.

BNs are used to answer queries of interest, such as the likelihood of an assignment of the values of all the variables. Other questions of concern are conditional probability of latent variables given values of observable variables, maximum a posteriori likelihood of variables of interest, the probability of an outcome (predictive modeling) when a causal variable is set to a value, and so forth.

2.2.2. Poisson regression

The Poisson process is a counter process represented as $\{N(t), t \geq 0\}$, where N(t) is the number of event observations prior to time t and where N (0) = 0. The counter tells the number of events that have occurred in the interval (0, t), figure 9, or more generally in the interval (t_1, t_2).

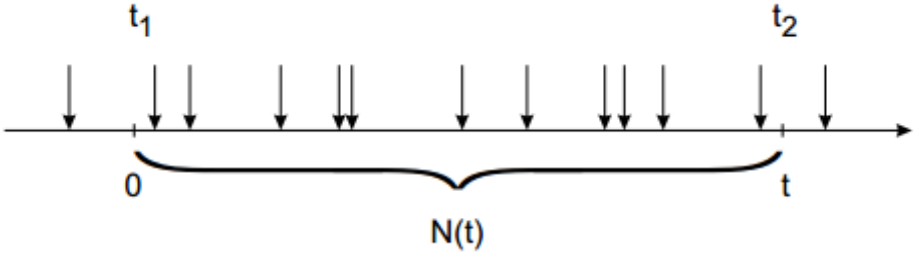


Figure 10: Number of events in the interval (0, t)

If the counting process is a Poisson process, then the probability of observing d counts over a period of length m is

$$P\{N(t+m) - N(t) = d\} = Poisson(d; \lambda m) = e^{-\lambda m}\frac{(\lambda m)^d}{d!}, d = 0, 1, \ldots$$

Equation 3: Poisson process

So, the probability of observing d counts over a period of length m for a homogeneous Poisson process was a function of the homogeneous rate parameter multiplied by the length, Poisson (d; λm). To find the probability of observing d counts between two points in time in a not homogenous Poisson process requires integrating over the rate parameter function λ(t). So, the probability of observing d counts between time a and time b in is

$$P(N(a) - N(b) = d) = Poisson(d; \int_a^b \lambda(d)dt)$$

Equation 4: Probability of observing between time a and time b

For example, if the process is not homogeneous, we can take advantage of their periodicity. For example, although vehicle traffic flow may fluctuate throughout the day, the traffic flow patterns at the same day and time of the week, for instance, Mondays at 3 pm are typically similar. A simple method for implementing this non-homogeneous Poisson process that is characterized by periodic behavior is to segment the week into equal-sized time intervals and model each interval with a different Poisson rate parameter. The rate-setting function λ(t) then becomes a piecewise constant operate of time as illustrated in X, where at each discrete time interval there is a homogeneous Poisson process with constant rate λ(t). This method is appropriate for sensors that measure human activity commonly reported as an aggregate count measurement across a fixed time segment.

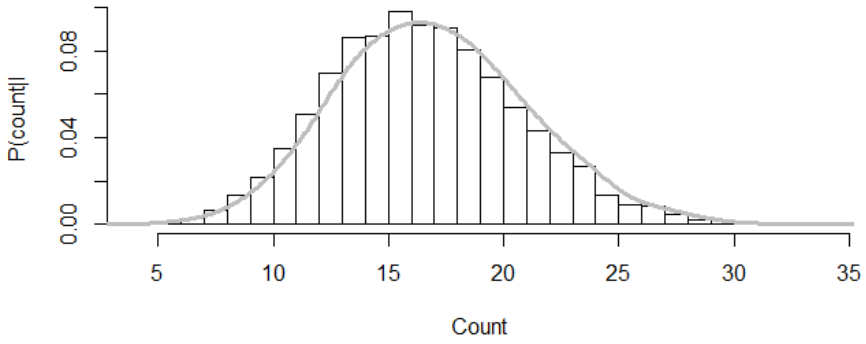


Figure 11: Poisson distribution function, l = 17

Using this piecewise constant model, the total count measure reported at time t follows a Poisson distribution. The Poisson distribution, P(d; λ(t)) is a discrete probability distribution, where the probability of observing d counts during a fixed window of time is presented on the formula below, and the rate parameter λ(t) is the expected number during the particular period window at time t.

$$P\big(d;\lambda(t)\big) = e^{-\lambda(t)}\frac{\lambda(t)^d}{d!}, d = 0,1,\dots$$

Equation 5: Poisson distribution

Figure 10 shows the probability mass function (pmf) for a Poisson distribution with rate parameter λ = 1.5 (blue) and the pmf for a Poisson distribution with rate parameter λ = 17 (red).

A history of count measurements at the same time and day shares some common characteristics with the Poisson distribution. The observed number cannot be negative, and the counting process is discrete. Another feature of the Poisson distribution is that the variance of the distribution is equal to the mean of the distribution.

2.2.3. Time-series model

A time series is a sequence of data points collected over time, and they are uniquely suited to capture the time dependence of these variables. Time series analysis techniques have been used for (i) forecasting, (ii) the determination of the temporal ordering of some variables through Granger causality tests, and (iii) the determination of the over-time impact of the variables or specific discrete events [70]. Here we use all of them, with the focus on continuing multivariate time series. Vector autoregression (VAR) describes the evolution of a set of variables over the same period as a linear function of their past values.

Let $X_t = (X_{1t}, X_{2t}, X_{3t})$ with n1, n2, n3 dimenasion respectively, the VAR model is:

$$X_t = J(B)X_{t-1} + u_t = \sum_{i=1}^{l} J_i X_{t-i} + u_t$$

Equation 6: Vector autoregression model

where u_t is a noise process, J_i is a vector of intercept variables, and $X_t \epsilon R^{d\times l}$ for t ϵ 1, …, T be *d dimensional* multivariate time series. The null hypothesis of X_3 does not cause X_1 can be formulated as:

$$H_0: J_{1,13} = J_{2,13} = \cdots = J_{k,13} = 0$$

Just replacing the vectors and matrices with scalars will produce the definition of an autoregression (AR). VAR is the mechanism that is used to link multiple stationary time-series

variables together. It is characterized with stationarity, unit roots, and cointegration, for which a different type of analysis is needed.

A stochastic process whose distribution does not change when shifted over time is considered a stationary process. For many statistical procedures in time series analysis stationarity is the underlying assumption, and often non-stationary data is transformed to become stationary. When the underlying processes *Xt* is stationary, possible causal structure grows as some variables increase. The pairwise causal structure might change when different conditioning variables are added.

Time series yt defined as $A_p(B)y_t = C(B)\epsilon_t$ has a unit root if $A_p(1) = 0, C(1) \neq 0$. For yt, the existence of unit root implies that a shock in ϵ_t has permanent impacts on yt. If yt has a unit root, then the traditional asymptotic normality results usually no longer apply. We need different asymptotic theorems. When a linear combination of two I(1) processes become an I(0) process, then these two series are cointegrated. Cointegration implies the existence of long run equilibrium and, a common stochastic trend and restrictions on the parameters: proper accounting of these limitations could improve estimation efficiency. With integration, we can separate short and long run relationship among variables. It can be used to improve long-run forecast accuracy.

Let Yt be k-dimensional VAR(p) series with r cointegration vector $\beta(\, p \times r)$.

$$A_p(B)Y_t = U_t$$

$$\Delta Y_t = \Pi Y_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta Y_{t-1} + \Phi D_t + U_t$$

Equation 7: Cointegration

Cointegration introduces additional causal channel (error correction term) used by one variable to affect the other variables. Ignoring this other channel will lead to invalid causal analysis.

Forecasting is one of the primary objectives of multivariate time series analysis. Prediction from a VAR model is like a prediction from a univariate AR model. Consider first the problem of forecasting future values of Yt when the parameters J of the VAR process is assumed to be known

$$\hat{y}_t = J_1\hat{y}_{t-1} + J_2\hat{y}_{t-2} + + J_l\hat{y}_{t-l} + \epsilon_t$$

VAR model provides information about forecasting abilities of a variable's or a group of variables'. The following intuitive notion of a variable's forecasting ability was developed by Granger 1969 [62]. If one variable (or group of variables) y1 is found to be helpful for predicting another variable (or group of variables) y2, then y1 is said that y1 Granger-cause y2; otherwise, it is said to fail to y1 Granger-cause y2.

Testing causality is one of the most important and challenging issues. Experiments can be performed where all other causes are kept fixed except for the factor that is under investigation. By repeating the process for each possible variable, we can determine the causal relationship among factors or variables. Time series analysis looks for forecasting from the unique unidirectional property of time arrow: cause precedes effect. Based upon this concept, Granger proposed the following working definition of causality.
Xt is said not to Granger cause Yt if for all h > 0

$$F(Y_{t+h}|\Omega_t) = F(y_{t+h} - X_t)$$
Equation 8: Granger causality

where F denotes the condition distribution and $\Omega_t - X_t$ is all the information in the universe except series *Xt.* In plain words, *Xt* is said to not Granger cause *Yt* if X cannot help predict future Y. It is defined for all h > 0 and not only for h = 1. Causality at different h does not imply each other. Ω_t Contains all the information in the universe up to time t that excludes the potential ignored common factors problem. Formally, y1 fails to Granger-cause y2 if for all s > 0 the MSE of a forecast of y2, t+s based on (y2, t, y2, t−1, ...) is the same as the MSE of a forecast of y2,t+s based on (y2,t, y2,t−1,...) and (y1,t, y1,t−1,...). Clearly, the notion of Granger causality does not imply true causality, it only implies forecasting ability.

When data points are autocorrelated with each other, then simple classifiers would not work well. For those use cases using time series classification techniques is more adequate. Use cases where time series prediction is used are power load forecasting, demand prediction for retail stores, revenue forecasts, and yield and crop forecasting. In our analysis, we applied VAR methodology on a crime dataset.

2.3. Context of Smart City: smart interconnected communities

Cities are complex and dynamic systems that function and interact within multiple coincident spatiotemporal scales: from changing the traffic light to the seasonal hum of power stations, meeting increased energy demands [34]. The term ”city” constitute not only a geographical area characterized by physical and environmental features, but also includes a multi-layered construct containing multiple dimensions of social, technological and human-related components and services [34] [125]. To better describe this evolving urban environment, a variety of terms have been used, including “network city,” “digital city,” ”cyber city,” and “global city” [145]. The different metrics of urban smartness are reviewed by various groups of researchers ~~[1]~~[12] [40] and show the necessity for a standard definition of what constitutes a smart city, including the features, and how it performs compared to traditional cities.

Up to today, there is not an official definition of smart city accepted by academics, government, and business. The term was first used in the 1990s by the California Institute [40], whose focus was on creating modern infrastructures within cities, how communities could become smart and how a city could be designed to implement information technologies. Moreover, after 2010, research in this domain increased dramatically, and the usage of the terminology consequently changed [40]. One of the most-cited definitions of smart city is the following [67],

> “*A city that monitors and integrates conditions of all of its critical infrastructures, including roads, bridges, tunnels, rails, subways, airports, seaports, communications, water, power, even major buildings, can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services to its citizens.*”

In industry, the most commonly used definition is from IBM [70],

> “*Smart city is connecting the physical infrastructure, the IT infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city*.”

This initiative has an international context and has been supported by programs in many countries around the world, including the European Union’s Seventh Framework Program, the United States with Smart America Program (a $40 million grant to turn Columbus, Ohio into a smart city), Australia’s Csiro Program, and programs in China, India, and South America. Support has come from government institutions such as National Institute of Standards and Technologies (NIST) for developing Framework for Smart City Architectures and National Science Foundation (NSF) for leading the program Smart & Connected Communities. Furthermore, it has attracted significant vendors from the ICT industry including IBM (Smart Planet, 2008), Cisco (2011), Oracle, Intel, Siements, and Fujitsu (2014).

The Smart City nowadays is an essential strategy to improve the quality of life of billions of people all over the world [136]. It refers to the capability of a city to understand events that characterize its internal and external dynamics (e.g., demographic changes, road traffic, transportation issues, and so forth). Per Boyd Cohen, the critical components to making a city smart are smart {government, economy, people, living, mobility, and environment}. Concepts within the smart environment and a possible domain for applications spanning from government services, public transport, crisis management and smart grids, to health care, travel planning, smart home, and museum. Figure 12 illustrates some exemplifying applications in smart cities.

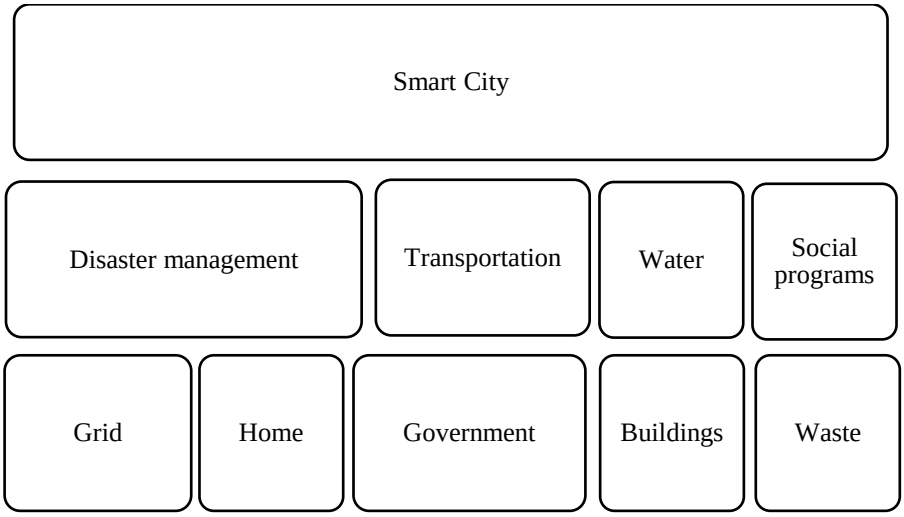


Figure 12: Overview of smart city applications

The heart of these smart environments is sensing from the various interconnected sensors that collect and send data to the information center where wise decisions can be made. Based on sensing characteristics the technical architecture of smart city applications includes three layers:

- Perception layer - identifies the objects and collect information through 2D barcode, RFID reader, camera, GPS, and so forth
- Communication layer - makes transmission and processing of information obtained in the perception layer through the integrated grid of communication, network management center, data center and intelligent processing unit
- Application layer - analyzes and processes massive data and information through cloud computing, and other smart technologies [136]

Sensors and devices from the perception layer depend on the requirements of the target application and are very tightly-related to the communication infrastructure.

The communication infrastructure also depends on the requirements of the target application, as well as different types of networks. For instance, Home Area Networks, Wide

Are Networks, Field Area Networks [69], and Urban Automation Networks (UAN) [60] supported by some of the communication standards, like Dash7, ZigBee, WiFi, LTE, 3G, NFC [69]. Existing UAN's are Low Rate Wireless Personal Area Network (LRWPAN), Wireless LAN, Mobile Network Operator, Simless Operator, and Delay Tolerant Networks (DTN). Moreover, only DTN cannot support event-based application [60], so their usage is not recommended in event-based types of applications.

The application layer can be implemented with any of the existing software architectures that participate in Smart City (SC), including Architectural Layers (AL), Service Oriented Architectures (SOA), Event Driven (ED), Internet of Things (IoT) and Combined Architectures (CA) [83]. This current state-of-the-art results [83] shows that CA gives better results, and the most common combination is IoT with AL, which allows researchers to add additional technologies that enhance system capabilities. There are attempts to combine three and even all of these in order to empower SC with the advantages of each of them. However, the intention is to create a typical pattern based on IoT architecture. Although network architectures support software architectures, it also depends on application requirements and another constraint. For instance, a very flexible solution was proposed by Presser et al. [114] : a multi-tiered hierarchical structured mobile-cloud architecture for scalable collaboration.

In general, sensing can be categorized as remote, in-situ sensing and collective sensing [34], mobile sensing, and social sensor sensing (social media as a sensor). Sensing data is characterized by external context associated with the environment and internal context associated with an individual level [125]. Spatiotemporal context involves more than just location, it incorporates scientific and human observations, and implications can be unique, personal, global and social. In general, contextual sensing allows context that has not been previously considered. Incoming sensed data can then be processed to detect relevant events like air pollution, humidity, wind speed, pollution, traffic congestion, and cultural events (such as concert, art exhibition, gallery presentation) and provide timely support.

Building a smart city requires providing both hardware and software. For hardware that collects information for the events happening within the urban environment, and software that utilizes the gathered information and helps decision makings in urban life, see figure 13.

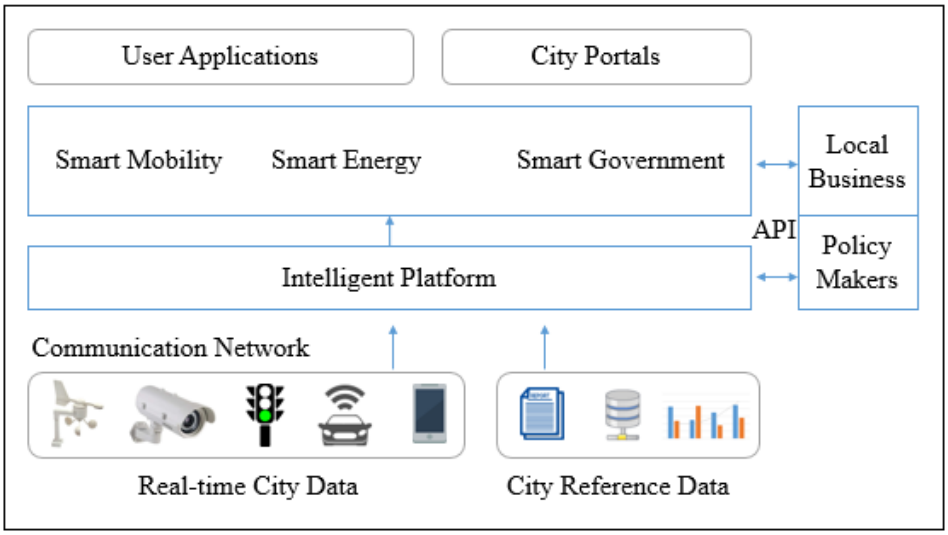


Figure 13: Sensing Architecture in Smart Cities applications

2.3.1. Basic requirements

The smart city application or service must be able to identify three main components of an event: (i) to analyze, (ii) to provide location and contextual sense of it and (iii) to react appropriately and on time. To achieve this, the cities must improve their spatial data infrastructure based on corporate, local, state/provincial, national, and global levels. Proposals for an innovative way of sensing public places use an aggregation of individual sensors (spatially enabled citizens, geosocial networks) and devices sensors (cameras as sensors) [119]. Because of this requirement, the initiative for open data was created in the U.S.A., and Europe (more details are provided in the next subchapter 2.3.2), and social networks were used for analyzing city pulse and support smart cities' operations. Monitoring provides a large variety of data for detecting events and situations in the appropriate context that might signify a potential point of interest (safety issues, topic interest).

The data processing procedure in smart city applications has three phases: (i) data gathering, (ii) data analysis and (iii) result delivery. Thus, raw urban data gathered for smart city applications may arrive in different formats, e.g., traffic information, parking spaces, bus timetables, and so forth, as well as from various interfaces, e.g., APIs, websites, and web services. Due to the data and interface heterogeneity, the data aggregation or abstraction from public data sources is typically carried out manually, resulting in static or outdated information. The analysis results should be context-aware and knowledge-based to provide insights into the current situations [112]. The vision of the data processing pipeline in smart city applications is illustrated in Figure 14.

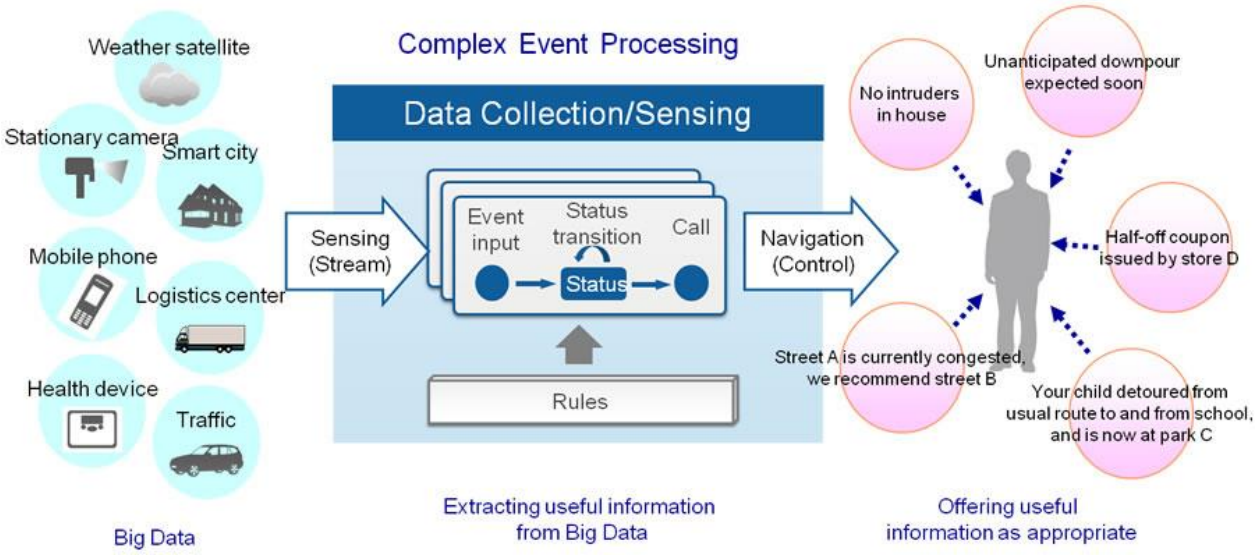


Figure 14: General data processing architecture in Smart City applications [55]

Figure 14 shows general computer architecture for a smart city application; sensors are at the lowest level of the architecture collecting data and transfer it to a gateway, which in turn sends the data to a processing system. The gateway chooses either to or not to summarize or preprocess the data. The Connection between sensors and gateway is via some of the previously mentioned communication protocols. The relationship from the gateway to analytic servers can be via the Internet, LAN, or Wi-Fi connection, and it can use a higher level protocol such as Message Queue Telemetry Transport or Constrained Application Protocol. Data is coming mostly from computer-based systems (e.g., transaction logs, system logs, social networks, and mobile phones) and sensors.

A standard feature and significant problem that sensing data shares is that each involves: diversity of data formats and mobility, information analysis and integration, optimization of large amounts of data coming from various smart appliances in diverse formats, real-time responses to situations happening in the city, and adaptation to the environment.

In the following work, Event Processing (EP), and Predictive Analytics (PA) techniques namely are discussed to fulfill these requirements. For our validation experiments, we address the needs of public safety information services and urban management. A description of the general use case is presented in Appendix C, while individual case studies are explained in Chapter 5. The output of the demonstrated solutions can help decision making authorities and policymakers manage city resources and future urban development better.

2.3.2. Open Data

The diversity of data helps in creating better models that describe and identify cities needs. The benefit of integrating different data was a reason for launching the open-data initiative in April 2010 [144]. The goal of the program is to make some of the city data available online for everyone to use to build an application that will help citizens. This initiative was followed by the United States, Europe, and India, and web-based open data repositories were created. Experiments used safety data sets from the U.S. Government's open data [2].

In this work for the experimentation were used cases studies extracted from the Smart America initiative[3] to improve urban management and public safety for people in the towns, for instance, by identifying safe zones and investigating where government resources need to be allocated. More focus is given to Mongomery County, Maryland, U.S. using global and real-world event records obtained from the open data initiative. The datasets description and pattern characteristics used for experiments are explained in Section 5, while the experiments are explained and revisited in details later in Section 4 and 5.

[22] www.data.gov
[3] Smart City Challenge is the challenge to prepare all American cities for the future

2.4. Conclusion

Smart Cities brings technology together and provides an intracity assessment of everything. Their focus is on improving public infrastructure and services that improve urban living. Some of the high-level use cases are waste management, smart parking, smart buildings and bridges, traffic management, air and water quality, smart road tax, and urban planning.

The research in creative environments like smart cities tackles the problem of developing the extensible and stable applications that satisfy some user's needs. It incorporates works from several disciplines including analysis and inference, modeling, transforming, aggregating, validating, testing as well as service composition. This chapter describes the three most essential research categories: EP, PA, and sensing in smart cities: architecture and user requirements.

EP and PA are the concepts needed for the remainder of this dissertation. First, an overview of event processing is presented; then multisensory predictive analysis methods are described to clarify how these two concepts are related, then, the concept of Smart City is introduced, and examples of Smart City use cases are elaborated. The features of these use cases are analyzed to justify the need for using event services in the application scenarios as an integration of Complex Event Processing (CEP) and PA technologies. In recent years, there are few research efforts which have explored the possibility of combining PA methods with CEP to provide proactive solutions. Initially, it was proposed by Fülöp et al. [54], who presented a conceptual framework that combines PA with CEP to get more value from the data. The necessity of providing event processing capability as event services for Smart City applications is explained by analyzing the requirements of smart city applications, emphasizing the need for semantic event integration model, automatic detection, and processing, as well as automatic and adaptive execution on environmental changes.

The next chapter presents an extensive review of studies related to the contributions developed in this research, including event detection methods and models, multisensor analytical methods and their applicability in smart city scenarios.

Chapter 3

Related work

"The important thing is never to stop questioning."
- Albert Einstein

Summary. This chapter presents the latest research in event processing with a focus on automatic event detection methods, event data models, and their practice in smart city domain. This chapter presents a comparison between existing approaches and explains the advantage of the presented solution. Multisensor analytical learning methods are discussed as well as their corresponding function in event processing with a focus on smart city case studies for improving city services using data from social sensors, identifying safe zip zones for pedestrians and improving the prediction of crime events.

3.1. Event processing

In this subsection, we survey two aspects of the work related to this research. The first part presents existing approaches for event detection methods with the focus on the methods used in non-structured data like tweets and their combination with other methods in use case scenarios to improve city services. The second part is focused on characteristics of existing event models in complex event processing and city-based scenarios, and differences from our event model.

3.1.1. Event detection

When data comes from heterogeneous data sources and devices like textual, multimedia, and network, the data can be in a non-structured or semi-structured format; this does not conform to an explicit and well-defined event definition. To extract useful information in the form of events from time-evolving data that comes from various sources event detection methods are used.

Event detection methods are based on the (i) type of event, if it is specified or unspecified (known or unknown), (ii) type of the detection method, supervised or unsupervised, and (iii) detection task like retrospective event detection or new event detection [17]. A set of rules defines specified events; they can be: hand-written rules, machine learning algorithms like

classification, or sequence models like named entity recognition (NER). Unspecified events are identified by the following approaches (Widder et al. [156]):

- Deterministic: identifying casual events
- Probabilistic: represents causality with probability index
- Cluster operations: creating groups of objects based on specific criteria
- Discriminant analysis: using classification methods
- Fuzzy set theory introduces degrees of membership of an object to a set
- Bayesian belief networks: using network graph that describes dependencies between states
- Dempster-Shafer method: creates conclusion by combining information from different sources
- Hidden Markov model: use probability distribution of known process the likelihood of the hidden process is determined.

The appropriate method is chosen depending on the application domain, and if the event is known or not, for example, it is more suitable to choose some of the unsupervised algorithms for hidden or non-specified events. We are interested in detecting known event patterns, more precisely the focus is on known event type detection for the non-structured text data type. Non-structured text data is coming from the sources like social media, blogs, web portals, and so forth. These data sources are widely used and present in daily life and provide useful information for public opinion, network and system usage. Because of its text format, it is considered more challenging to process and detect events.

For the experiments, we chose to use a non-structured dataset from microblogs like Twitter as a data source, also called social sensor. Twitter is one of the first and most famous microblogging providers with millions of active users. Each user can create public posts to initiate discussions, participate in debates, and follow the communication of others. Many significant achievements are accomplished using social networks as a data source in different areas like newscasts, early warning systems for detection of earthquakes, and predicting the federal elections [4], [21], **Error! Reference source not found.**[120], [123], [136]. In this s tudy, each user is considered as a sensor and tweets are sensor information with the time, location, and topic features. Identifying events from social media presents several challenges:

- Heterogeneity and immense scale of the data
- Messages are short, which means that limited content is available for analysis
- Frequent use of irregular, informal, and abbreviated words, the large number of spelling and grammatical errors, and the use of unsuitable sentence structure and mixed language

Event detection methods on tweets are classified into three categories by Zhao et al. [159]: (i) specific event detection, (ii) person related event detection, and (iii) general event detection. For our analysis, we are interested in specific and general event detection. We made a contribution to one of the challenges presented by Goswami et al. [61], that is to couple textual, spatial and temporal along with network structure, by creating a new tool for efficient event

detection. However, while most osf the event detection in micro-blog platforms happens with textual content, to have additional, more accurate detection to use along with temporal, spatial and word-network structure is considered.

For fruitful and precise event detection and because of the noise characteristic of tweets it is necessary to have a pre-processing step. Studies show that pre-processing is an essential step in text analysis [66], it consists of (i) cleaning, (ii) transformation, and (iii) feature selection. The first phase (i) is cleaning the data of noise parts of the text that are uninformative and do not have any valuable impact on the general orientation. If we keep these parts, we will have high dimensionality, which will make the classification process more difficult and not precise since each word is treated equally as one dimension. The next step in pre-processing is transformation (ii) where each abbreviation, acronym, smiley, icon, contractions, and misspellings are replaced with full words, so we have a standard of only text words and sentences. Features (iii) in the context of text mining are the words, terms or phrases that characterize the document. This means they have a higher impact on the event detection or sentiment level than other words. There are several techniques used for preprocessing. They are N-grams [110], part of the speech [59], user-based features [97], tokenization [13], and based on some entities in the text (URL, emotions, words, character, punctuation and slang/offensive words) [21].

There are several ways to assess the importance of each feature by attaching a specific weight to the text. The most popular ones are Bag of Words (BOWs) and Term Frequency-Inverse Document Frequency (TF-IDF). In our analysis, we used context-based pre-processing, and we made a comparison between BOWs and TF-IDF features as two widely used techniques in natural language processing (NLP) and information retrieval. Each term in the vector is typically weighted using the standard term TF-IDF approach, which evaluates how important a word is to a document in a corpus. Most of the algorithms expect binary feature vectors with a fixed size rather than the raw text with variable length. To address this, we used techniques that provide utilities to extract numerical features from text content. We use the most frequently techniques for vector representations BOWs and TF-IDF to represent text messages regarding a feature vector. In the majority of the NLP applications, BOW's and TF−IDF features are commonly used for text processing applications, sentimental analysis on Twitter data, blogs and classification of sentiments from micro-blogs [4], [30], [122].

Existing supervised learning approaches were chosen for event detection analysis. They were successfully applied to several works and achieved excellent results for classification problems, such as earthquake, influence, e-cigarette usage detection, spam detection, sentiment classification, and traffic event detection [13], [123], [59], [97], [110]. Their results show that Support Vector Machines (SVM), Naive Bayes (NB), and Random Forest (RF) gives better results compared to the other algorithms. However, Aphinyanaphongs et al. [13] in their experiments, only used two input classes smoker and non-smoker. In our experiments, we used sixteen different classes. They represented the most discussed topics on Twitter; more

information is presented in Chapter 5.2. We compared existing and widely used algorithms [76] (NB, SVM, RF) for event classification of tweets. NB classifier is a probabilistic classifier, SVM is a discriminative classifier, and RF classifier is an ensemble method where more than one decision tree is used for classification purposes based on voting rule [32]. Table 1 illustrates advantages and disadvantages of these algorithms using the attributes that characterize social sensor data as criterions.

Algorithm	**Advantages**	**Disadvantages**
Naive Bayes	Robust to missing data Can work well with small dataset Fast	Sensitive to noise datasets Not capable of dealing with unbalanced dataset Does not consider dependency between parameters
Support Vector Machines	Efficient with small sample size Can process high-dimensional data High performance on complex classification tasks	Sensitive to noise and missing datasets Not capable of dealing with unbalanced dataset Limited speed and high memory requirements
Random Forest	Robust to noise data Efficient with small and unbalanced dataset	Sensitive to handle missing data Requires more processing time, and this increase in the number of features increases

Table 1: Advantages and disadvantages of different types of classification algorithms

NB can handle missing values better compared with the other two algorithms and has fast performance, but it is sensitive to noisy and unbalanced data. SVM is also efficient with small data sample and processing performances but is not tolerant to noisy and missing datasets. RF is a collection of trees, each independently grown using labeled data, it is tolerant to noisy values and efficient with the small and unbalanced dataset. However, it is sensitive to noisy data and requires more processing time as the number of features increase. We did experiments using these three algorithms with different features. Results showed that RF with any of BOW or TF-IDF features performs better than the other two. More details are presented in Chapter 4 and 5.

Case study context

In a domain of urban context-aware application, we used a case study of improving local services by identifying a set of event types and sentiment measurement from social sensors per contextual information. We focused on tweets that allowed us to analyze the view of the public

on generally discussed topics and measure their perceptions regarding a variety of subjects. Timely understanding of the tweets reporting various concerns about the city is necessary for city authorities to manage city resources. Our case study was inspired by future smart city challenges presented by Ahmed at al. [2].

In the domain of smart cities, some authors [2], [6], [119], [128] are focused on improving city services by clustering the events by similarity. They measure sentiment level for a particular topic, or they combine the physical and social sensor data from an absolute geo-location such as country, city, or neighborhood. Their output is intended to help city representatives, first responders or citizens. However, the existing methods do not consider the impact and similarity relation between event types and sentiment level. Alternatively, in this work, we applied these characteristics to improve city services, and due to its practicability and flexibility, we extended the model to combine temporal, spatial and network of words information.

The similarity between events is typically measured using traditional metrics such as the Euclidean distance, Pearson's correlation coefficient, and Cosine similarity. More recently, other similarity measures have been proposed such as the Hellinger distance [31] and the clustering index [78]. For our experiments, we chose to use one of the most used metrics, Cosine similarity metric between categories to measure how similar they were and how we can group them. The similarity between events was used in a variety of cases, e.g., to cluster real-world events with its associated tweets, for identifying most relevant posts, and for catering previously selected news [20]. Our work is different from these approaches because we added sentiment dimension to the similarity which enriched the understanding of the city topics. In the analysis process, we also considered temporal, spatial and word-network structure.

Another group of authors enriched event detection methods with sentiment analysis to present more accurate results. Sentiment analysis (SA) is an excellent way to measure customers' loyalty, keep track of sentiments towards brands and, products, or just measure their perceptions regarding a variety of topics. Having information about what topics interest people can help to improve service recommendations, such as traffic routes, air pollution zones, and so forth. This type of services can be enriched and made more accurate by measuring the sentiment level. SA involves classifying the text into categories such as 'positive,' 'negative,' 'neutral,' or even in more detailed levels.

It has been used to measure sentiment during Hurricane Irene [97] using Maximum Entropy classifier and sentiment detection mechanism that determines the public reaction by matching them with previously selected keywords related to terrorism [39]. Salas-Zárate et al. [126] combine trend detection and sentiment analysis for decision-making purposes based on the Spanish language using Maximum Entropy Classifier and TF-IDF metric for similarity between event topic. These solutions are only focused on one topic and measure the sentiment on that topic for some period, they do not feel considerate similarity with other subjects and do not have automated way of topic detection. We also have more productive pre-processing step compared with [39], [97] and [126], and we use real-world datasets instead of synthetic [39].

However, the existing methods did not directly consider the impacts of similarity relation between event types together with sentiment information. And, due to its practicability and flexibility, we extend the model to combine temporal and network information.

In summary, we designed and developed an algorithm that automatically extracts event knowledge from collected data from a social sensor like Twitter with context-aware pre-processing algorithm that handles text data and provides full word meaning to each character in tweets. We enriched the algorithm with sentiment identification per event type as well as a similarity measure between event types, to get a better understanding of the detected event knowledge. The end goal of this method is a tool for an automatic tool for event detection that integrates text event streams and use algorithms that complement each other to provide a better understanding of city events. Also, we provide a graphical representation of the event trends for decision makers, such as city representatives to (or "intending to") increase visibility and awareness of city trends during different times of day, week and period of the year, and locations. More details for the proposed theoretical solution and experimentation are presented respectively in Chapters 4 and 5.

3.1.2. Event models

Event recognition techniques employ rich representation that can represent events with complex relational structure, e.g., events that are related to other events with spatiotemporal constraints. Every event instance is represented by the relevant information about the event. This information includes the time of occurrence, data relevance to the application domain, and some additional data. This confirms the explicit and well-defined formal data model with relations, attributes and so forth.

In this work, the focus is on spatial-temporal events, so the core metadata fields have a temporal dimension such as start and end time of the event, spatial dimension, event description, and value, as well as other information related to the application domain. Event models specify metadata fields for the data streams that carry or pertain to events. This data modeling approach was first presented by [1] in 2000, where they organize the events using the following abstractions: classification, aggregation, generalization, and association. Some authors [127] assume that using more primitive is better. Their idea of modeling primitives is that the core data message is independent of any application. While other groups of researchers [112] use the approach of determining data fields during the process of setting up a system.

Authors in [102] define event model as a tuple with the event name, set of preconditions and effects, where the event occurs when all their preconditions are met. The author of [22] used UML representation to design a data event model: this approach is close to our event model. Moreover, a logic-based event model was presented by [132] and [15]: they used logic programming convention with variables, constants, and predicates, and rules for handling noisy data streams. Other researchers have an approach for event models as the algorithm that detects

complicated situation, such as [152]. They define a traffic detection event model as the primitive event, complex probabilistic event, event type, and they applied adaptive Bayesian network that produces better accuracy. The authors in [49] define an event model as a process to model, develop, validate, maintain and implement event-driven applications, where the goal is to derive an event published by a customer that wants to react to it. The same group of authors from IBM [4] explain the event model as a way to improve event processing with the model similar to a decision model with the following benefits: independent of technology, no program code required for understanding, a simple diagram designed to drive event logic correctly, and absence of technical terms. The approach to our EM is also to derive event the customer wants to react to, with the respect of the presented benefits.

We evaluated the existing event models based on the requirements for event-based applications mentioned in Chapter 2. The requirements are classified as (i) processing characteristics support, such as mechanisms for handling missing and uncertainty data, support for any type of data and raw data, (ii) required data fields for the event model, and (iii) provision of other additional requirements.

First, we reviewed the existing work using the first criterion: processing characteristics. We found that some event models are specific to data type or domain type, like VERL [53], SsVM [46], EventOntology [116], and Event E [155]. Table 2 shows the comparison between domain-specific event models. Their event model is applicable in the domains characterized by spatial and temporal dimensions as a necessary basic requirement. Other characteristics of the event data streams are support of a variety of data types (text, multimedia) and raw data, and a mechanism to handle missing and uncertainty data. The authors from LODE [134] and, Event E [155] created their model to provide support for missing data, while the authors from [127] provide support for uncertainty. Models LODE [134], REseT [148], EventOntology [116], and Inventory [151] support any type of data, as illustrated in Table 3. In our approach, we created the EM to support any detected events, with the parameter that points to the original raw data. Also, we created rules for handling uncertainty and missing data that are easily modifiable.

[4] http://www.modernanalyst.com/Resources/Articles/tabid/115/ID/3036/Introducing-the-Next-Horizon-The-Event-Model-TEM.aspx

	Processing				Required fields								Other requirements	
Event Model	Raw data into model	Handle missing data	Handle uncertainty	Data types	Static/dynamic	Action/verb/event	Participant/actor/subject/Agent	Object	Time	Location	Device associated	Device affected	Ontology	Processing language
SsVM (Ekin et al, 2004) [46]	ns	ns	ns	Video	ns	Y	Y	Y	Y	Y	ns	ns	no	SQL
VERL (Francois et al, 2005) [53]	ns	ns	ns	Multi-media	ns	Y	Y	Y	Y	Y	ns	ns	VEML 2.0/OWL	VEML/OWL-DL
EventOntology (Raimond et al, 2007) [116]	Y	ns	ns	Multi-media	ns	Y	Y	Y	Y	Y	ns	ns	Created by authors	RDF/XML
Event E (Westermann et al, 2007) [155]	ns	ns	Y	Multi-media	ns	Y	Y	ns	Y	Y	ns	ns	no	no
EM (Kotevska et al., 2016)	Y	Set by coder	Y	Any	Y	Y	no	no	Y	Y	Y	no	Any	Any

*ns = not specified, Y = yes

Table 2: Comparison chart for event models that support specific data types

Uncertainty is unavoidable in daily life as so in events produced by the environment, to deal with it intelligently we need to represent and reason about it [68]. One way to quantify the uncertainties is by adding a probability distribution to the possible worlds. Challenges related to uncertainty in event processing are classified into three categories [55]: (i) namely model, (ii) usability, and (iii) implementation issues. Challenges for (i) namely model is to construct a flexible generalized model that can match the appropriate model for a specific implementation. Usability (ii) is defining the rules and probabilities (for the cases when the history is not good predictor). Implementation issues (iii) are related to scalability and performance requirements (developing general algorithmic improvements or developing domain and application specific efficient algorithms). According to [15], there are four approaches to deal with event uncertainty; they are the following:

1) by ignoring when the damage is not substantial
2) using sufficient definition to deterministic detection
3) event detection with probability, like PGM, Markov Logic Networks, probabilistic logic programming
4) fuzzy set and possibility theory using knowledge and data from multiple indicators

Authors from [15] dealt with the noisy data stream problem by cross-validating the data from multiple sources and eliminating the ones that report noisy. What they consider types of noisy were known in advance, via machine learning or expert knowledge. Hassan et al.[71] developed a method based on event matching for handling uncertainty. Their method deals with inexact event type matching, by considering reusability based on similarity of event attributes

not event patterns. Our approach was to face these challenges by creating rules that handle known uncertainty at attribute level and using probabilistic methods for the usability challenge, with the focus on specific application.

Regarding the second criterion, required fields, that should be filled; we classified them into few categories based on the definition of the fields. For example, participant and actor is the same entry, event, action, and verb are the same entry, object, time, location, and device associated or affected. Tables 2,3, and 4 illustrate these properties across domain specific and multi-domain event models. Most of the authors except [44] [134] have event entry. Some of them [52] [116] have a device associated with them. Event ontology [116], OpenCyc[5], ABC supports sub-events, and Event F [127] offers a possibility for it, while LODE does not support sub-events. Another group of authors [19] compared event models by using the criteria of stretch in time, location, and the participation of objects, but however other factors should be considered as well.

We created our EM to be scalable, as only some of the fields are required while the rest can be added when needed. Event sources can also be easily added and removed, which allows flexibility. Our EM supports fixed and relative location, for instance, geographical coordinates and city name, both supports modeling sub-events as a separate entity or parameter property. The relation between events is modeled as a distinct functionality that can model complex events, event type, event patterns, and relationships. Also, we separate the entities on static that contains the information that does not change over time and dynamic that contains the information that always changes over time.

	Processing				Required fields								Other requirements	
Event Model	Raw data into model	Handle missing data	Handle uncertainty	Data types	Static/dynamic	Action/verb/event	Participant/actor/subject/Agent	Object	Time	Location	Device associated	Device affected	Ontology	Processing language
OntoEvent (Ma et al, 2015) [94]	ns	ns	ns	Common data types	ns	Y	Y	ns	Y		ns	ns	Created by authors	OntoEvent lang.
REseT (Uma et al, 2014) [148]	Y	ns	ns	Common data types	ns	Y	Y	ns	Y	Y	ns	ns	Created by authors	DL
Common Event Model (Fowler et al, 2009) [52]	ns	ns	ns	Common data types	ns	Y	Y	Y	Y	Y	Y	ns	Created by authors	RDF/XML
Event ontology (Zhong et al, 2012) [159]	ns	ns	ns	Common data types	ns	Y	Y	ns	Y	Y	ns	ns	Created by authors	RDF/XML
SOUPA (Chen et al, 2005) [38]	ns	ns	ns	Common data types	ns	Y	Y	ns	Y	Y	Y	ns	COBRA-ONT/OWL	RDF/XML
EM (Kotevska et al., 2016)	Y	Manu ally	Y	Any	Y	Y	no	no	Y	Y	Y	no	Any	Any

*ns = not specified, Y = yes

Table 3: Comparison chart for event models that support common data typ

[5] http://opencyc.org/

The third criterion for event model is support for additional requirements such as semantics and language. Some of the authors [107] described the importance of semantic enrichment in event data modeling, called semantic because the meaning of the data field titles is essential to the model. Requiring an ontology on a topic for an event model can restrict the data types that can be used, as well as the domain. Ontologies are needed in event models such as Event F [127], OntoEvent [94], REseT [148], and LODE [134]. In some cases, the ontology was created specifically for the event model, such as for OntoEvent and REseT. The metadata fields in our EM, align with those in the DOLCE ontology [127]. That makes it easier when we have data from different streams to fit into the same event model, and it will help ensure interoperability.

Authors in [71] created a thematic event model defined as a pair of two sets, theme tags, and tuples. Where the theme is defined as a set of terms that describe the same thing, for instance, the set {‘energy,' ‘appliances,''building’} refers to an event which conveys power consumption of appliance of the building. Their proposed approach suggests associating events and subscriptions with tags to describe their semantic themes. The topics represent a lightweight way to communicate event semantics across systems. They used a method of semantic relatedness (using Cosine of Euclidian distance) between each pair of attributes or values from the subscription, event and distributional semantics. Our EM allows adding additional parameters which allows one of them to be considerate as a theme.

	Processing				Required fields								Other requirements	
Event Model	Raw data into model	Handle missing data	Handle uncertainty	Data types	Static/dynamic	Action/verb/event	Participant/actor/subject/Agent	Object	Time	Location	Device associated	Device affected	Ontology	Processing language
Event F (Scherp et al, 2009) [127]	ns	Y	ns	Any	ns	Y	Y	ns	Y	Y	ns	ns	DOLCE+DnS	RDF/XML
Eventory (Wang et al, 2007) [151]	Y	ns	ns	ns	ns	Y	Y	ns	Y	Y	ns	ns	no	no
Event ML (IPTC, release 2014)	ns	ns	ns	ns	ns	Y	Y	ns	Y	Y	ns	ns	no	XML
CIDOC CRM (Doerr et al, 2007) [44]	ns	ns	ns	ns	ns	ns	Y	ns	Y	Y	ns	ns	ISO 21127	XML
LODE (Shaw et al, 2009) [134]	Y	ns	Y	ns	ns	ns	Y	ns	Y	Y	ns	ns	Created by authors	RDF/XML
Event Calculus (Shanahan, 2001) [132]	ns	ns	ns	ns	ns	Y	ns	ns	Y	Y	ns	ns	no	Some logic language
EM (Kotevska et al., 2016)	Y	Set by coder	Y	Any	Y	Y	no	no	Y	Y	Y	no	Any	Any

*ns = not specified, Y = yes

Table 4: Comparison chart for event models that support any data type or it is not specified

Another group of authors used specific processing languages (e.g. XML [44], Event ML [6], RDF [127] [52] [159] [44] [38] [134] [116], SQL [46] or logical language [132]) when extracting data from the event model. This approach is an additional constraint and makes it dependable, while our approach is the model to have an absence of technical terms**.**

Case study context

In a domain of urban context-aware applications, we used a case study of improving local city services with the focus on improving pedestrian safety by identifying safe zip code zones for pedestrians. We addressed a real problem by using real-world data from multiple sources and properties like static, dynamic, semi-structured and structured format.

In the domain of a smart city, multiple projects were created, such as EventShop [112], CityPulse framework[7], INSIGHT[8], SmartSantender [64], and OpenIoT [73]. EventShop [112] accepts data streams and includes modules functionalities that query the event instances relevant to the application. Their events are put into a location-based grid structure called Emage, that can integrate events from complex data streams. Their solution works as a standalone application and for now, can not be incorporated into existing solutions. Some solutions for integrating heterogeneous event information resources in smart city scenarios were proposed by [64], [82], [129], and [153]. Gutierrez et al. [64] focused on creating a platform closer to citizens by adding participatory sensing capabilities; their solution also works as a standalone application. However, authors from [82] concentrated on the problem to find an efficient way to handle real-time semantic annotation of sensor data in dynamic environments. Their work is part of the CityPulse framework, and for their experiment, they used publicly available data streams. They underlined the incoming data with Stream Annotation Ontology that links the segment description with time extent. Another group of authors [73], also addressed the same problem. They introduced a new approach to IoT data stream analytics for real-time data acquisition, annotation, and processing of sensor data, where the server deploys complex clustering algorithms to analyze data in real-time. They validated their results by understanding complex phenomena such as the impact of air pollution on human health. Authors from [129] developed a framework for heterogeneous data and validated in the scenario of disaster detection and alarming, their work is part of INSIGHT. On the other side [153] presented a framework that integrates the representation of XML and SQL event streams into unified event fusion format with the general specification for easier event processing. These solutions show the usefulness and effectiveness of their method, but they do not present how it can be used in existing platforms.

[6] https://iptc.org/standards/eventsml-g2
[7] http://www.ict-citypulse.eu/page/
[8] http://www.insight-ict.eu/

The advantage of our approach compared with these solutions is that it can be easily integrated to any of the current event processing platforms ; it is not dependent on any query processing language, and the proposed event model is already sidelined with upper-level ontology which avoids additional semantic annotation for time-sensitive services. Also, the output data can be used by CEP engine or predictive analysis in any domain or application of interest, which corresponds to the event models that are defined as a specific problem solver (more detailed explanation is presented in Chapters 4 and 5).

3.2. Analytical models for multisensory learning

This subsection is organized into three parts; the first part presents the prediction models we used to improve city safety services and explains their characteristics, differences, the reason we choose them, and the domains they showed promising results. The second part gives an overview of how they were used in smart city case studies and the similarities between those solutions and ours, and what we additionally consider in our solution. The third part focuses on the problem of dealing with event stream changes, such as data loss, caused by various reasons, existing solutions in the smart city context, and how our solution is different from the others.

The research area of event processing deals with processing data that are viewed as events, making sense of them, and sending the results to the end consumers (administrators, users, another system) about consistent behavior patterns based on the rules that are determined in advance. The rules or patterns describe the usual circumstances of the events which have been experienced and they improve over time. However, to improve the effectiveness of event processing, it is possible to apply the results from related research area like predictive analytics. Predictive analytics deals with the prediction of future events based on previously observed historical data by applying sophisticated methods like machine learning. The historical data is often collected and transformed by using techniques like the ones of event processing, e.g., filtering, correlating the data, and so on.

To create functionality for proactive event processing, we used the following three types if prediction algorithms: probabilistic (Bayesian) graphical model (PGM), Poisson regression (PR), and time-series (autoregressive [AR] and vector autoregression model [VAR]). Each of them is used for specific purpose, PGM gives probabilistic value to the predicted event and probabilistic influence of each data stream to the result. PR provides a more precise number of anticipated events, not interval of values like PGM, while AR and VAR are used when we want to use time-dependent past values for determining the prediction results. Because of these unique characteristics and distinct advantages, we choose to use these three types of prediction models. Comparison between these methods based on spatiotemporal data characteristics and challenges facing complex data is illustrated in Table 5.

Algorithm	Advantages and Disadvantages				Context of use in our framework
	Support of data fusion	Incompleteness	Uncertainty	Relation between events	
PGM	Yes	Yes	Yes	Yes, probabilistic dependency metric	Likelihood of event occurrence
PR	Yes	No	No	Yes, relationship index (p-value)	Concrete number of events
AR & VAR	Yes	No	No	Yes, causality index	Number of events dependent on past period

Table 5: Comparison table between three different types of prediction algorithms (PGM, PR, AR & VAR)

PGM model is designed for modeling complex probabilistic systems, and it can deal well with complex information characterized by incompleteness and uncertainty. It provides a probabilistic description of the relation between events and context. Usually, is used in scenarios when is necessary to know the likelihood of event occurrence. PGM and it variation have shown satisfactory results in traffic domain [160], energy systems [45], and genetic analytics [138].

Count models were used in cases when we needed to have a precise analysis of expected totals of events. We chose the most straightforward PR because it often provides an adequate representation of the variability observed in count data. This model showed promising results in medicine [96] for diabetes prediction.

Time-series data models are used instead of the other methods because time-series analysis accounts for the fact that data points taken over time may have some internal structure such as trends, seasonal variation, or autocorrelation. VAR was chosen because it captures interdependence between multiple time-series data streams. This model has been used in financial markets [105] and energy consumption [131].

Because of their unique characteristics switching between algorithms depends on the requirements and specific context. Also, PGM and PR are static models; they do not consider the time component for the event streams. But in real-world scenarios event streams change over time, so it is necessary to consider the changes over time when modeling.

The next part of this subchapter reviews the existing work for static and dynamic prediction models in the domain of smart city applications.

Case study context

In the domain of urban context-aware applications, event processing, and prediction models have been used for solving various types of problems. One of the critical aspects of smart cities is public safety [25], the focus of the case study we used was identifying safety areas especially pedestrian safety zones and improving safety by providing more accurate prediction and creating a model that can adapt in the event of data loss or another changing environment. The goal of the experiments was to build models that are beneficial for the city and citizens and use real-world open data.

Successful work related to city safety was done by [29] they demonstrated a scenario where a network of sensors was installed in a neighborhood in Rockville, Montgomery County, Maryland (MD), U.S.A. This sensor network monitored environmental factors, such as explosive gas, smoke, and automatically alerted residents upon discovery of a possible emergency. Their work was more focused on designing the sensor network, implementing detection rules for unexpected and unwanted situations, and notification system. Conversely, our work is oriented to handle the issues with complex data streams, and identify patterns that can help prevent unwanted situations in future.

Hierarchical Bayesian Network has been used for modeling traffic condition prediction system [160]. For their analysis, the authors used three types of datasets: weather, local events (e.g., significant sports events, musical concert, big parties and ceremonies that affect the traffic), and traffic information from E-ZPass system from Western Massachusetts Street at UMass, Amherst, Massachusetts, U.S.A. While weather and local events datasets are available online, traffic-related data does not have open access. They demonstrated successful prediction congestion with 93% accuracy overall and showed the impact of each data source on the congestion during different times of day and week.

We propose a model with the similar goal, but in addition, we can handle data in open access from the perspective that we also used real-world data streams with open access, and we address a real problem of pedestrian safety that was identified by the local government. Also, in our solution, we combine CEP methods with predictive models. For instance, we use event detection mechanisms before data streams are accepted to the system after the data is formatted to the event model for interoperability and this model support the functionality of using event patterns and link identification for analysis.

PGM that explains the conditional dependencies between traffic variables was demonstrated by [11]. They designed a graphical model structure enriched with declarative knowledge from ConceptNet[9]. The evaluation was done using real-world traffic open access dataset[10] for a week from October 14 to October 20 from San Francisco Bay Area, California U.S.A. The results showed that combining the graphical model with declarative knowledge

[9] http://conceptnet.io/
[10] 511.org

provides richer domain model for reasoning; it especially improves correlation and causal based knowledge. However, they did not explain how scalable their model is when the new data stream is added or removed.

Zhu et al. [161] in their research combined CEP with predictive analysis using Bayesian Networks, and their results showed promising results for large-scale IoT applications. This group of authors created simulation system based on SUMO (Simulation of Urban Mobility) simulating virtual FRID and GPS readers and a series of rules that simulate real traffic. Also, routing rules are related to contexts such as weather, congestion state, and car accidents. Their method has better accuracy in comparison to traditional methods but does not use real-world datasets.

Another group of authors focused more on creating models for predicting unwanted behaviors. For instance, predicting truck [100] and car accidents [50], crime rates [106], and highway fatalities [101] to understand and possibly prevent their future occurrence of this event to increase the safety level in cities. However, these solutions used only static datasets for their analysis, and are not scalable regarding adding new datasets. Chan et al. [36] used a dynamic dataset from video cameras and applied PR for crowd counting in open spaces such as streets. This solution is very helpful for safety aspects of cities, but also is not scalable for adding new data streams. These solutions do not provide an automated way of event detection and event model solution for data fusion in the case of using more than one datasets. These solutions are oriented to benefit city authorities. Our initial focus is also city representatives, and authorities but our models can be adjusted for individual users as well.

In our experiment, we devised the case study to answer an on-going need by local government to improve pedestrian safety in Montgomery County, MD, U.S.A. In 2007, a Pedestrian Safety Initiative was introduced in Montgomery County that used rule enforcement (tickets for the drivers and pedestrians who violate traffic laws), education (campaign to raise awareness), and engineering (traffic light adjustments, improving lightening, sidewalks) approaches to reduce the number of accidents [11]. Our solution complements the Pedestrian Safety Initiative by showing where the most accidents are predicted to occur; this information is intended for county representatives. Decision makers for the county can then consider deploying their resources better to change street signs or police traffic, for example, to heighten safety at the times and locations predicted to be unsafe. The proposed solution and demonstrated results predict which county region by zip-code is going to be safer based on the past number of pedestrian incidents per zone.

We experimented with both PR and PGM, PR to predict event occurrence based on several predictor variables, while PGM can represent dependencies among events on a graph. The end goal of our approach was to present and evaluate the event model and not prediction. Instead, we incorporated prediction into the model, to show that the event model is compatible with applying prediction models; we also wanted to provide users a tool for safety awareness

[11] http://www.montgomerycountymd.gov/DOTPedSafety/overview.html; https://volunteer.truist.com/mcvc/org/opp/10610402110.html

during different times of day, week and period of the year, as well as visibility of types of incidents. More details are presented in Chapters 4.3 and 5.3.

However, the existing methods did not directly consider the impacts and changes of the event streams. In this work, we created a model to improve the resilience of the prediction results. We used VAR and created a dynamic method of data loss and environmental changes taking advantage of data source networks. We chose a case study on improving safety by providing more accurate prediction and creating a model that can adapt in the event of data loss or another changing environment. For case study experiments we chose crime dataset, explained in more detail in Chapters 4.4 and 5.4.

Crime analytics is a rapidly growing field, taking advantage of the increase in data collection to identify spatiotemporal patterns in crimes incidents and develop crime prediction models [74] [103]. For example, clustering techniques are used to discover spatial patterns in crime incidents, and regression techniques are used to find relevant and meaningful temporal patterns. Spatiotemporal crime trend analysis, which studies the dynamic interplay of location-dependent and time-dependent aspects of crime, utilizes a wide variety of techniques including pattern mining, association rule mining, and combinations of the previously mentioned methods [84]. For example, the authors of [63]**Error! Reference source not found.** demonstrated a m ulti-agent model to predict areas in which future criminal incidents are likely to happen and use both physical and cyber-criminal activity. The authors of [85] utilized Bayesian inference to create a geographical map to show potential crime factors per area. This weighted geographical profile provided probability estimation for the next crime hot spots and likely locations for future crime incidents. Such results can be used to improve police resource deployment. Gerber et al. [58] applied Latent Dirichlet Allocation semantic analysis to identify crime-predictive Twitter discussion topics. Ranson et al. [117] applied linear regression to map weather factors to crime dynamic, identifying a strong relationship between climate change and crime incident number and type. However, the existing methods did not directly consider the impacts and relationships between cities. Alternatively, in this work, we apply the link analysis information to improve prediction results. Alternatively, due to its practicability and flexibility, we extended our model to combine geographical information.

Some efforts in this direction have been made by Ballesteros et al. [26] they proposed a method for forecasting future safe values in cities based on user context and location safety values. The authors used three types of data sets: crime incidents, Yelp and census datasets from Dade County Miami, Florida, U.S.A. for identifying safe places and predict safety level in each place. They experimented with three time-series types of algorithms: such as ARIMA (Auto Regressive Integrated Moving Average), ANN (Artifical Neural Network) and LES (Linear Exponential Smoothing) models to predict the number of crimes to occur at a location during near future. Their input data was per month for the period of four years, using RMSE and MAPE accuracy metrics. Their results showed that ANN is slightly better than the other two algorithms. Their analysis was focused more on creating various prediction models, such

as daily, based on crime type and safety index. They do not show how the model will adapt when a new data source is added, or when the data stream is missing for various reasons. Their work is oriented to the user as an individual, not for city representatives.

Other works that used open data reports in their analysis and time-series dataset are discussed below. Authors in [111] aimed to forecast traffic safety performance measures using crash data. The output can be used to determine targets for future safety improvement programs. They experimented with two datasets, one corresponding to the number of fatalities and other to injuries from 1994 to 2012, reported by Nevada Department of Transportation, U.S.A., to improve traffic safety across emphasis areas. They experimented with deterministic (Simple Deterministic, Holt, Brown, Damped-trend, Seasonal, Winter-additive, Winter-multiplicative) and stochastic (ARIMA, SARIMA) time-series models for reducing fatalities and serious injuries, and evaluate with RMSE and MAPE (mean absolute percentage error) accuracy metrics. They used only one type of dataset crash data, but no participatory sensing or demographics. Both methods do not consider the problem with missing data and do not adapt to the environmental changes.

Anan et al. [10] deployed a temporal trend sensitive system using a combination of ARIMA time-series models to respond to the dynamics of energy demand. Similarly, [5] combines dynamics via a decision tree with time-series prediction to improve prediction of complex events. The algorithm identified current model prediction error and dynamically determined to increase or decrease the time-series training window accordingly. The VAR method showed good results for solving the problem of traffic flow forecasting [3] in the transport network of the major cities. Although their solutions can be applied to other cases, they do not consider relationship analysis and dynamic nature of changing the environment.

Methods that additionally capitalize on shared temporal trend information across data streams to optimize resilience have been used in the application domains of weather and transportation. Tokumitsu et al. [143] combine support vector machine regression and a data network between neighboring weather sensors to interpolate missing spatiotemporal weather data from one sensor using neighboring sensors. However, this method assumes a static network based on sensor proximity in which all data streams are associated with same sensors. A network of weather data streams is also addressed by the authors of [43]. Network dynamics was allowed with potential changes in data sharing connections. A greedy algorithm selected data streams with the most recent data and rejected data streams identified to provide poor prediction performance. Pearson's product moment correlation function was also used to detect and evaluate data streams for shared temporal trends dynamically, and these learned relationships were then utilized across a set of regression techniques to ensure system resilience to data faults. Pravilovic et al. [113] also utilized correlation in their geo-sensor data resilience networks. Temporal and spatial correlation between data streams was identified and used to establish a spatial-based cluster of data streams. A stationary correlation was assumed, and a static data sharing network was formed. However, while correlation analysis can provide useful information on shared temporal trends, ist does not give an indication of the statistical

significance of these relationships. Additionally, highly useful shared trends between data sources separated by significant distances may be missed in such spatial correlation based techniques.

However, the existing methods did not directly consider the impacts and link relation between data streams. Additionally, we apply the missing data mechanism to improve the service. Due to its practicability and flexibility, we extended the model to combine geographical information and weather temperature.

In summary, the dynamic model can deal with missing data by taking advantage of network nodes between data sources - in our case, cities, and weather. It is scalable in the sense of adding new information streams and selecting the right entities based on relationship analysis using qualitative and quantitative approaches. The output always provides optimal results by having a list of the best models and depending on the environmental changes the model adapts dynamically. More details are presented in Chapters 4 and 5.

3.3. Conclusion

Several methods have been discussed in Section 3.1.1. to solve the problem of automatic event detection on non-structured data for improving local services by using social sensors as input data source. As discussed in Section 3.1.2, an essential prerequisite for efficient data sharing is to identify the event, event context, and structure in the standard format applicable for complex data. Finally, Section 3.2 discussed the prediction models as learning solutions for improving city safety services and dynamic model that adapts to the environmental changes and failures to maintain city services.

The next chapter presents details of the theoretical explanation of the proposed solutions. It also describes the overall idea and the conceptual framework (FEDAP – Framework for Event Detection Analysis and Prediction) that processes complex data in smart environments.

Chapter 4

Contributions: Proposed Theoretical Solutions

"*The question is not what you look at, but what you see.* "
- Henry David Thoreau

Summary. This chapter presents the theoretical explanation of the proposed solutions in this research work. Section 4.1 discusses show the modules in this dissertation are combined into a framework called FNEDAP (Framework for Network Event Detection Analysis and Prediction). The chapter details the FNEDAP design and implementation. Section 4.2 describes the proposed solution for improving local services which include automated text network analysis, the similarity between network topics and sentiment analysis. Section 4.3 describes the semantic event model that format the heterogeneous data in common format to facilitate data sharing among services. Section 4.4 describes dynamic network model for event forecasting that adapts to environmental changes and disruptions.

4.1. FNEDAP: Framework for Network Event Detection Analysis and Prediction

This subchapter discusses how the individual modules developed in this dissertation are working together in the FNEDAP.

This framework treats all data streams that come from various network sources, such as temperature sensors, police reports, and social sensors together in the same context and turn all the data into knowledge so that we can take advantage of it, and improve real-world situations. Therefore, data streams are heterogeneous and can be in both textual and non-textual formats. It combines textual information with non-textual data to bring additional knowledge that can be received if it was used only one type. Non-text data helps text-mining by mining text in the way it is defined by non-text data, while text data helps non-text mining by using text data to interpret patterns found in non-text data.

It collects contextual information from the various interaction devices, methods, and sensors, and use these contexts to provide relevant information. The social sensor is considered a voice of the humans; they express their emotions (e.g., opinion about food), perceptions (e.g., political elections), or locations (e.g., restaurants). A social network represents the vast network of peoples' voices around the world. While physical sensors measure environmental variables such as temperature, humidity, or air quality, and represent outside factors. Other sources of data streams that describe communities where we live are reports from police, health

organizations, and city and government representatives. They provide information about traffic, medical trends, demographics, policy, and laws.

The main contributions to this dissertation are the three modules : (i) event network analysis in Chapters 4.2 and 5.3, (ii) scalable semantic event model in Chapters 4.3 and 5.4, and (iii) event prediction that follows the principle of proactive event processing and adaptation in Chapters 4.4 and 5.5. These modules are grouped together into a framework called FNEDAP. The high-level architecture is illustrated in Figure 15. Its components also correspond to the general event network processing architecture, present by [133].

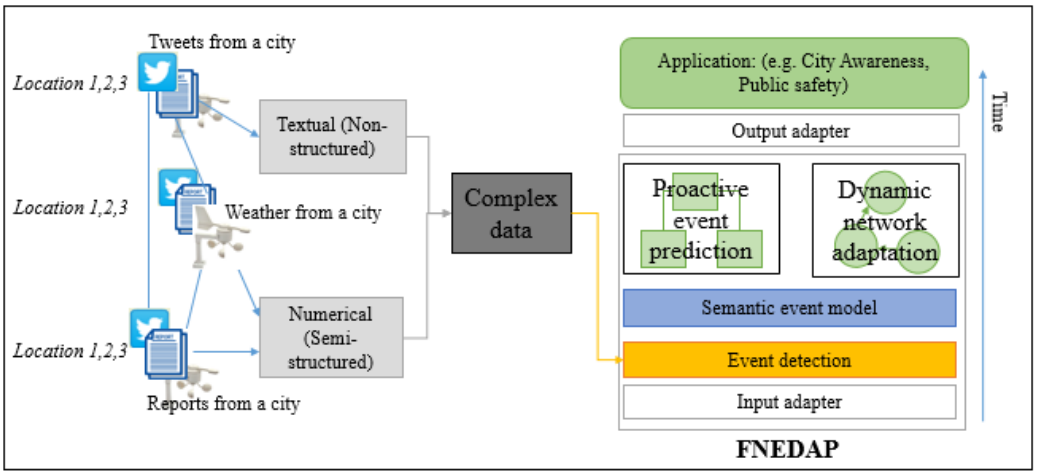


Figure 15: High-level architecture for the FNEDAP

The main parts are event detection, semantic event models, and dynamic prediction. The other layers and components are necessary from a perspective to enable events input and output, evaluation, and demonstration of the effect of the proposed approach. The framework is designed to be generic and facilitate other similar scenarios. It is also considered to work as a standalone application and to be integrated into existing event processing platforms.

The components of the general architecture shown in figure 15 are detailed in the following.

Various data streams (text and non-text) represents complex data; input adapters are responsible for receiving these data streams by connecting the FNEDAP with the outside world. It accommodates the necessary technology and syntax level for handling data stream messages. So far, the developer is responsible for handling input and output events and adapts them to the system.

The event detection module is responsible for identifying the event streams of interest, including performing the necessary pre-processing steps such as cleaning, transforming, feature selection, execute rules for solving data incompleteness and other uncertainties.

The next module is the semantic event model; it extracts the relevant properties of the data streams and converts it into their respective event model objects.

Following module is analytics. It accommodates proactive event prediction and dynamic network adaptation approaches proposed in this dissertation. The analytics are performed using one or multiple event streams depending on the case study. The combination of event processing concepts with other concepts (e.g., sentiment analysis, event stream selection, a relationship between networked data streams, a prediction for the events of interest, similarity identification, adaptable methods) helps to understand better what is happening regarding distributed network events.

The final module is output adapter. It is responsible for connecting the output results from FNEDAP to outside applications and services. The detailed view of the individual functional components inside each module is illustrated in Figure 16.

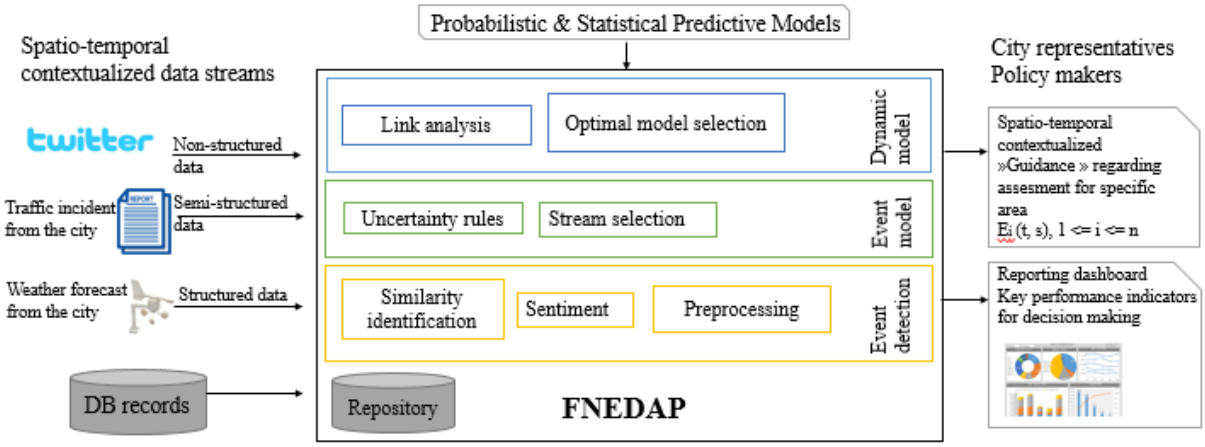


Figure 16. Internal architecture of FNEDAP

The final output of this context is intended to help decision makers for city representatives and policymakers to understand how the changes in the towns and communities, have an impact on well-being. Preferably, this knowledge will lead to the creation of safer cities, communities, and neighborhoods.

More detailed explanation for each module and submodules is presented in the following subchapters, while the details of the deployment of FNEDAP and how it works with real-world data set is provided in Chapter 5.

4.2. Automatic event network analysis

The rapid growth of information has influenced the way people communicate, share and get information. They use various forms to express their thoughts or opinions, such as pictures, videos, and text. It has become more popular, for people to use Twitter, Facebook, Blogs, Forums and so forth for sharing events that are happening in their everyday lives. Specifically, microblogging messages have become a widely used tool for communication on the Internet. Twitter is one of the first and most famous microblogging providers with millions of active users. Each user can create public posts to initiate discussions, to participate in debates, and to follow the communication of others. People tend to comment on real-world events when a topic suddenly catches their attention, for example, a soccer game, adverse weather update, elections, breaking news and so forth. Based on this, in a city context, social sensing can be used to retrieve information about the environment, weather, well-being, traffic congestion, trends in the local economy, dangers or early warnings, and likewise any other sensory information that collectively becomes useful knowledge for the city's improvement and smartness.

In this study, each user is considered to be a sensor and tweets are sensor information with the time, location, and topic featured. Identifying events from social media presents several challenges:

- Heterogeneity and immense scale of the data
- Social media post are short, which means that only a limited content is available for analysis.
- Frequent use of irregular, informal, and abbreviated words, the large number of spelling and grammatical errors, and the use of awkward sentence structure and mixed language.

We are focused on tweets that will result in analyzing the view of the public on generally discussed topics and measure their perceptions regarding a variety of subjects. Timely understanding of the tweets reporting various concerns about the city is necessary for municipal authorities to manage city resources. This information complements sentiment and similarity level measurements.

To do so, we suggest the fully automated event processing algorithm that can accept any text data from the network sources during the time interval, detect the event type, and find the level of similarity and sentiment characteristics for event type groups. Based on this, the following research questions are generated:

- How do we extract knowledge from data collected from sensors (physical and social)?
  - How do we efficiently preprocess the text data streams from social sensors?
  - How do we extract the knowledge from text data streams?
  - How do we identify trends based on extracted knowledge?
  - How do we graphically present the event trends for city decision makers?

In this context event and event types are defined as follows:

Definition 1: Events in social sensors are real-world happenings that discuss the associated topic at a specific place and time. Text stream T = (t_1, …, t_n) where t_i is a tweet (Twitter message). Each tweet consists of a set of features F = (f_1, ..., f_k) at location L_a. The problem of automatic event detection is that it is difficult to identify the facts from a text stream T with the similar set of features F at location L_a, using rules.

Definition 2: " Event type is a specification for a set of event objects that have the same semantic intent and the same structure; every event object is considered to be an instance of an event type " [14]. An event type can represent either rare events deriving from a producer or derived events produced by an event processing agent. An event can be either simple or composite. A composite event type is a particular kind of event, which is made up of a collection of other event types. For example, the following tweet "*I am at Neil Simon Theatre for Gigi NY in New York*," belongs to the category 'art' in our use case scenario.

The data stream processing unit receives the incoming data streams and applies a set of processing modules, like:

(i) Preprocessing module includes: filtering the data by location and language, cleaning the tweets, transforming the tweets to the uniform format (where all the characters are translated into letters).

(ii) Event type detection module uses a set of rules processes events; logically they are defined as condition and action. Rules can be handwritten rules, machine learning algorithms like classification, or sequence models like named entity recognition. Our approach is built on using supervised machine learning methods. Based on the premise each tweet t_i belongs to a topic class C = (c_1, …. c_l), defined as a pair of components t_i.-> c_j.

(iii) Sentiment analysis (SA), involves classifying the text into categories like 'positive,' 'negative,' 'neutral,' or on even in more detailed levels. SA tackles the problem of analyzing the tweets regarding the opinion they express. The sentiment orientation of the topic states whether the topic is positive, negative, or neutral. A set of sentiment S = (s_1, ..., s_m) are assigned to each pair t_i.-> c_j, (t_i.-> c_j) -> s_l. For example, "*I am at Neil Simon Theatre for Gigi NY in New York*" has a positive sentiment score.

(iv) Similarity analysis between event types provides a way to test the difference between event groups. We quantitatively identify which event types are related to each other. For example, "*I am at Neil Simon Theatre for Gigi NY in New York*" has high similarity with "New York City is a great place for artists and athletics."

The output shows the relation between event types and sentiment level categories. By identifying the sentiment and similarity relationship between event types, the meaningful relations are highlighted so the decision makers (automatic or humans) and the services related to them can be assigned. Figure 17 shows the functional flow.

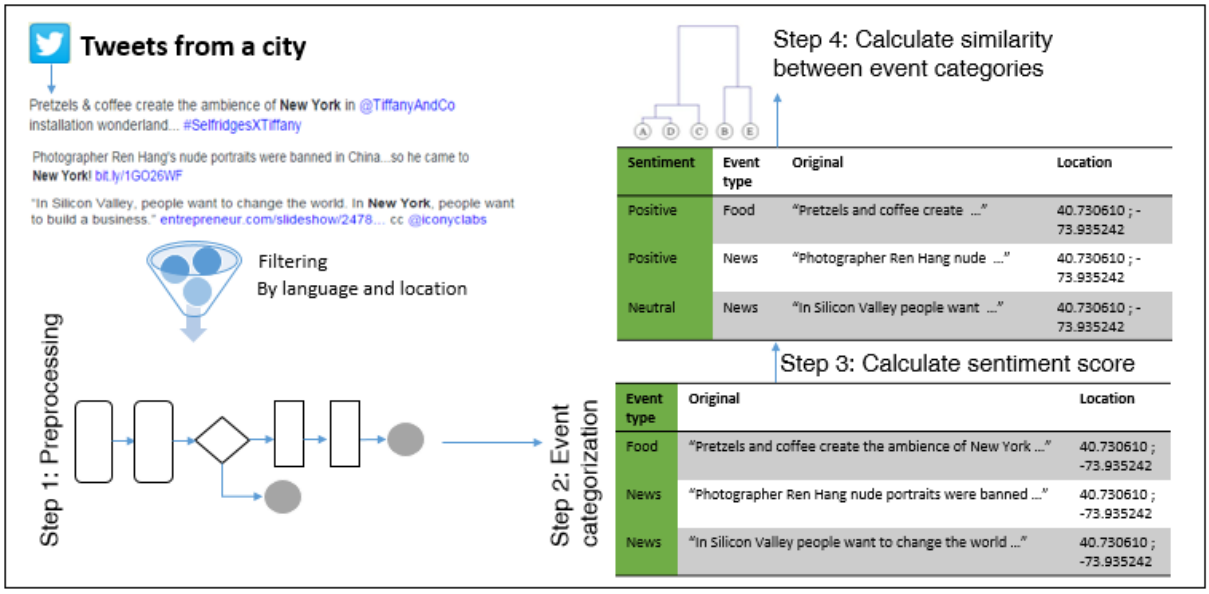


Figure 17: Functional flow diagram for event type detection using Twitter as data source

Therefore, in a domain of urban context-aware application, we use a case study of improving local services by identifying event types and adding a sentiment and similarity measurement from social sensors according to contextual information.

The following, subchapters are presenting in details each of the modules in this fully automated event processing algorithm.

Context-aware pre-processing and feature extraction

Pre-processing is the process of preparing the text for classification, by considering the contextual capabilities of input text streams. For instance, online texts like tweets usually contain a lot of noise and uninformative parts such as HTML tags, scripts, and advertisements. Also, many words in the text do not have an impact on the general orientation of it. Reducing those words makes the dimensionality of the problem lower and hence, the classification less difficult since each word in the text is treated as one dimension. Proper data pre-processing can be summed up in the following hypothesis: to reduce the noise in the text should help improve the performance of the classifier and speed up the classification process, thus aiding in real time sentiment analysis.

Another component that influence accuracy measurement is the feature selection process. Features in the context of event network detection are words, terms or phrases that have a significant impact on the orientation of the text than the other words in the same text. There are several ways to assess the importance of each feature by attaching a specific weight in the text. The most popular ones are Bag of Words (BOWs) and Term Frequency-Inverse Document Frequency (TF-IDF).

Pre-processing procedure cleaning and transformation

As mentioned earlier, Twitter text data is unstructured and noisy in the sense that it contains slang, misspelled words, numbers, special characters, special symbols, shortcuts, URLs, and so forth. The text messages with these particular symbols and, images may be more natural for humans to read and analyze. When the text data is mixed with other types of symbols and pictures, processing is a challenging task compared to processing of standard text data. As a result, pre-processing of tweets plays a significant role in the sentimental analysis. The typical characteristics of tweets that make it a challenging are:

- messages are very short and contain less text
- the message may contain different language text
- it contains special symbols with specific meaning
- data contains many shortcuts
- spelling mistakes

This part of the work discusses the methodology together with Natural Language Processing (NLP) techniques for the efficient processing of Twitter messages for analysis purposes. The proposed algorithm combine cleaning and tokenization techniques together. Cleaning, in this case, means removing tweet messages that have less than five characters in length and removing URLs and other characters presented in the Algorithm 1 below. There are three tokenization types: Treebank-style, Whitespace, Sentiment-aware, as well as a combination of n-grams. This algorithm uses sentiment-aware tokenization [12] in the pre-processing phase since it showed the best performance compared with the other two. The proposed algorithm is described as follows:

[12] http://sentiment.christopherpotts.net/tokenizing.html

Algorithm 1 Context-aware pre-processing algorithm

```
1: procedure Pre-processing of tweets
2:    for each tweet t_i ∈ T do
3:       Remove URLs, re-tweets, hashtags, repeated punctuation's and letters
4:       if length (t_i) < 10 do
5:       for each word w_j ∈ t_i do
6:          emotion icons, smilies, contractions
7:          abbreviations, acronyms
8:          misspelling words
9:       the end for transforming it into full, meaningful words
10:      Remove stop words, punctuation's, non-English words
11:      Convert to lower case characters
12:      end if
13:   end for
14: end procedure
```

Following is an example of what the data looks like after some pre-processing steps.

Original:

"I'm at Neil Simon Theatre - @nederlanderbway for Gigi (NY) in New York, NY https://t.co/WIGeWlYggy 676 taaaatoooo :)))))))))) aka ILY after #nelisimontheatre"
After removing retweets, URL, hashtags, repeated punctuation: "I am at Neil Simon Theatre for Gigi NY in New York NY 676 tato :) aka ILY after."

After conversion of smiley symbols, acronyms, abbreviation, contractors, emotion icons:

"I am at Neil Simon Theatre for Gigi NY in New York NY 676 tato Smile also known as I love you after."

The final output, after removing stop words, numbers, punctuation characters:

"Neil Simon Theatre Gigi NY New York NY tato Smile known love."

Emoticons are regularly used in many forms of social media; it is the same case for acronyms, abbreviations, and slang words. Because of these reasons, we used implementation

unctionality to convert smileys[13], emoticons[14], acronyms and abbreviations[15][16][17], contractions[18] and misspelled words[19] to full, meaningful words.

Tweets are processed by removing characters like repetitions, particular traits, stop words, and English stop words[20]. Even though collected tweets are in the English language, there were words in other languages, in such cases, tweets are ignored for analysis. Despite the advantages of reducing vocabulary, shrinking feature space and removing irrelevant distinctions and icons, pre-processing can collapse relevant distinctions, that are necessary for analytical purposes. Pre-processing text data improves the quality of text for analysis; whereas, coming to twitter data, because of short messages, pre-processing may end up with messages with no text data left for the Twitter message. In many cases, after pre-processing, some of the Twitter messages contain one or two words or less than ten characters. Some messages contain many punctuation marks, stop words, numbers, and non-English words that would not convey any information about the context, and are not used for analysis.

Feature extraction

Text data is a sequence of words, and these words cannot be fed directly to the machine learning algorithms for analysis purposes. Most of the algorithms expect binary feature vectors with a fixed size rather than the raw text with variable length. To address this, we need to use techniques that provide utilities to extract numerical features from text content. We use the most frequently used features called Bag of Words (BOWs) and Term Frequency-Inverse Document Frequency (TF-IDF) vector representations to represent text messages regarding a feature vector. In most of the NLP applications, BOW's and TF−IDF features are frequently used for text processing applications, blogs, classification of micro-blogs as well as news and scientific articles.

Even though these functions are extensively used for most of the text processing applications, a brief explanation is included:

1) *Bag-of-Words (BOWs)*

[13] http://www.netlingo.com/smileys.php
[14] http://en.wikipedia.org/wiki/List of emoticons
[15] http://marketing.wtwhmedia.com/30-must-know-twitterabbreviations-and-acronyms/
[16] https://digiphile.wordpress.com/2009/06/11/top-50-twitteracronyms-abbreviations-and-initialisms
[17] http://www.muller-godschalk.com/acronyms.html
[18] http://www.sjsu.edu/writingcenter/docs/Contractions.pdf
[19] https://en.wikipedia.org/wiki/Wikipedia:Lists of common misspellings
[20] http://xpo6.com/list-of-english-stop-words/

This model represents text as an unordered collection of words, disregarding the word order. In the case of text classification, a word in a text message is assigned a weight per its frequency in the text messages. The BOW representation of Twitter text message 'tn' is a vector of weights

$$'W_{1n}, ..., W_{wn}'$$

Where 'Win' represents the frequency of the i^{th} term in the nth text message. The transformation of a text message 'T' into the BOWs representation enables the transformed set to be observed as a matrix, where rows represent Twitter text message vectors, and columns are terms in each Twitter text message [115].

For example,

Original sentence S1 = "Neil Simon theater gigi ny New York ny tato smile known love."

Dictionary {neli, simon, theatre, gigi, ny, new, york, tato, smile, known, love, trying, bike, miles, week, biking, work, friday, park, fun}

S1 – [1 1 1 1 2 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0]

2) *Term Frequency and Inverse Document Frequency (TF-IDF)*

It is a feature vector representation method where shared and rare terms in the text messages are normalized so that rare terms are more emphasized along with successive terms in the text messages. Term frequency TF (t_i, T) is the number of times the term 't_i' appears in a Twitter text message 'tm', while document frequency DF (t_i, T) is the number of Twitter text messages that contain the term 't_i. If we use term frequency to measure the importance, it is possible to exaggerate terms that appear frequently but carry little information about the Twitter text message. If a term often appears across all the Twitter text messages, it means it does not carry special information about a text message. Inverse document frequency is a numerical measure of how much information a term provides and it is defined as follows:

$$TF{-}IDF\ (t_i, t_m, T) = TF\ (t_i, t_m) \times IDF\ (t_m, T)$$
$$IDF\ (t^i, T) = log\left(\frac{T}{1+ |tm \in T: tj \in tm|}\right)$$

Equation 9: TF-IDF numerical measure

Where $|T|$ is the total number of text messages in the corpus. Since logarithm is used, if a term appears in all text messages, its *IDF* value will become 0. Note that a smoothing term is applied to avoid dividing by zero for terms outside the corpus.

For example,

Original sentence S1 = "Neil Simon theater gigi ny New York ny tato smile known love."

Sentence (tweet) 1 contains 12 words where words "ny" appears 2 times, so tf = 2/12 = 0.167.
If there are 1000 sentences (tweets) and word "ny" appears 200 of them, then idf = log(1000/200) = 0.7, and tf-idf = 0.167 * 0.7 = 0.12.

Categorization, sentiment and similarity score

After tweet messages are converted to vector space model format, they are ready to be processed. The first step is a categorization of event types.

Categorization of event types

For a given tweet $t_i \in T$, the classification algorithm is used to label the tweet as event-related or non-event related by approximating the function f: T → C mapping tweets to their respective classes C = {Event Type1, Event Type2, …, Other}

Classification of online stream tweets helps to find valuable information up to date for each type of category. In this paper, tweets are analyzed and classified into predefined categories using supervised learning techniques: Naive Bayes, Support Vector Machine and Random Forest classifiers. These approaches are used for automatically classifying the tweets into predefined categories. Later, these classified tweets can be further analyzed to extract knowledge and sentiments for information providing purposes. The most challenging part of classification task is building the models that can be used to classify the online tweets automatically. The three different approaches used for classification of tweets are described as follows:

- Naive Bayes' Classifier

It is a probabilistic classifier which interprets the function ci(tj) regarding P(ci/tj). It represents the probability that a vector tj accounts for a tweet tj =< w1, j,...,w|T| j > of terms, which belongs to category ci, and determine this probability by using the Bayes' theorem, defined as:

$$P\left(\frac{c_i}{t_j}\right) = P(c_i) \times P(\frac{t_j}{c_i})$$

Equation 10. Definition of Bayes' theorem

Where P(tj) is the probability that a tweet was chosen at random, has the vector tj for its representation; and P(ci) is the likelihood that a tweet chosen at random, belongs to ci. The probability estimation P(ci/tj) is problematic since the vector number tj possible is too high.

For this reason, it is common to make the hypothesis that all vector coordinates are statistically independent. Therefore :

$$P\left(\frac{t_j}{c_i}\right) = \prod_{k=1}^{T} P(\frac{w_{kj}}{c_i})$$

Equation 11. Definition of Bayes' theorem, vector coordinates are statistically independent

- Support Vector Machine (SVM)

It is a discriminate model where it tries to find optimal separating hyperplane between two classes of examples. This method can be considered as an attempt to know between surfaces σ1,σ2,... of a dimension space |T|, what is separating examples of positive training from negative training examples. The set of training is defined by a set of vectors associated with the belonging category, where yj represents the belonging category.

$$(x_{1,}y_1), \dots, (x_{n,\dots,}y_n)\ X_j \in\ R_n, y_j \in\ +1{,}1$$

In a problem with two types; the first one corresponds to a positive example (yj = +1) and the second one corresponding to a negative example, (yj = −1) Xj represents the vector of the text number 'j'of the training set. The SVM method distinguishes vectors of positive category from those of adverse category by a hyperplane defined by the following equation:

$$W \otimes X + b = 0, W\ \in\ R_n, b\ \in R$$

Equation 12. Definition of Support Vector Machine

Such a hyperplane is not unique. The SVM method determines the optimal hyperplane by maximizing the margin. The margin is the distance between vectors labeled positively and those labeled negatively [130] [79].

- Random Forest

The Random Forest has many individual trees, where each tree votes on an overall classification for the given set of tweets and the algorithm chooses the individual classification with the most votes. The model is based on a different random subset of training tweets, and a random subset of available variables is used for deciding how the best to partition the data set. Each decision tree is built to its maximum size, and the outcome decision tree models constitute the ensemble model where each decision tree votes for the result and the majority wins [32].

Sentiment analysis

Sentiment analysis (SA) on already determined classes of relevant information from online stream tweets helps determines public opinions. We chose a method that uses a representation of the whole sentence based on the sentence structure, the order of words is considerate, as provided by the library from Stanford CoreNLP [21]. Also, this library supports five level of sentiment: Strong Negative, Negative, Neutral, Positive, and Strong Positive. We applied SA after event detection step, and now we have a more detailed view of the context of the trending topics.

An SAs, in this case, is very helpful, it adds a new value in measuring public opinion as well as know how to best harness the potential benefits of public services. For instance, the knowledge based on the observations for the last period of weeks, months or years shows us how the trends per topics and sentiments have changed per location. So the decision makers can use this knowledge for allocating possible resources in the future or taking some actions for prevention. Another example is, if an event in central park is detected and the sentiment is negative or neutral, then this knowledge can be used by the services related to navigation for runners or walkers and will reroute their paths. The recommender systems, in this case, will adjust their algorithms to include sentiment analysis and weigh differently services that receive a lot of negative feedback or fewer instances. However, the importance and sensitivity of the topic (emergency, earthquake) are highly relevant, in this case, the frequency of the tweets for the negative context can be lower. In the case of real-time processing, as topics and sentiments are changing, service recommendation needs to change adequately, too.

Similarity analysis

We also gauge the similarity between generated categories to find which of them are more similar, to provide better service. We use it to determine the sentiment with similarity index. The Euclidian similarity metric was applied to categories to measure how similar they are and Ward [154] clustering method to group them in clusters. The results were presented using dendrogram representation. Ward's method calculates the distance between two event clusters, A and B, as a sum of squares that increase when we merge them. In case of hierarchical clustering, the sum of squares starts out at zero (because every point is in its cluster) and then increases when the cluster merge. Ward's method is keeping this increase as low as possible. This solution is right for cases when the sum of squares should be low. Ward's method prefers to merge the smaller ones. Solving this trade-off enables us to show the behavioral heterogeneity of the entities that compose the analyzed system.

[21] http://nlp.stanford.edu/sentiment/code.html

Evaluation measurement

We evaluate the proposed solution using the evaluation metrics for event type classification, sentiment, and similarity score.

For evaluating the event type methods, text network data is separated into a training data set which is 80% of the whole data set and test data set which represents 20% or the whole data set. The output results are presented in confusion matrix and the performance metrics used to evaluate the classification results are accuracy, precision, recall, and F-measure (F1). Those metrics are computed based on the values of true positive (*TP*), false positive (*FP*), true negative (*TN*) and false negative (*FN*) assigned classes.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Equation 13: Accuracy based on confusion matrix

Precision represents the number of true positives out of all positively assigned documents, while recall represents a number of true positives out of the actual positive documents, and it is given by the equation 14. Finally, F-measure is a balanced method of precision and recall, where its value ranges from 0 to 1 and indicates better results the closer it is to 1. It is used to represent the results better in the case of unbalanced datasets.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Equation 14: Precision, Recall, F1 - metrics based on confusion matrix

The results based on the metrics are presented on the table for visualization purposes for easier understanding.

Sentiment measurement

We measure the number of sentiment scores per event type. There are five sentiment scores; they are strong negative, negative, neutral, positive, and strongly positive, represented as $Sc_{i=1}^{5}$. For each event type class $C_{j=1}^{16}$, we have the following formula :

$$Sp_j^i = \frac{\sum Sc_i}{\sum T_j}, where\ j = 1,..16; i = 1,..,5$$

Équation 1. Sentiment measure for class and type

The results are represented an online figure that presents the rising trend per topic and sentiment.

Similarity measurement

A similarity measure for Ward algorithm says that the distance between two clusters, A and B, is the sum of squares between them added up over all the variables. The formula represents distance method:

$$d(A,B) = \frac{n_A n_B}{n_A + n_B} \|\vec{m}_A - \vec{m}_B\|$$

Équation 2 Ward distance method [154]

Where m_j is the center of cluster *j*, and n_j is the number of points in it, *d* is called the merging cost of combining the custers A and B.

The results are represented using dendrogram figure that shows the group of clusters of similar topics.

Discussion

In this sub-chapter, we present an integrated fully automated event processing system for detecting high-level topics reported as network text data. We developed an efficient pre-processing algorithm where every word is essential for analysis and supervised event identification was performed in several stages like feature selection and classification. Our experiments suggest that a Random Forest classifier combined with TF-IDF yields better performance than many leading classifiers. We also showed that combining classification methods with other algorithms like sentiment and similarity provides a deeper understanding of the detected event types in each location. We validate by using data from social networks like Twitter.

The next sub-chapter explains the generic approach that integrates several complex data streams with an ontology for event semantic analysis.

4.3. Scalable semantic event model

Processing practical problems require a combination of information from different sources. Heterogeneous events arrive from various sources, physical sensors (e.g., temperature, traffic accidents, video cameras) and social sensors (e.g., Twitter). They come in different formats such as for instance text, numerical, image, and video. This variety of data needs to be converted into a standard representation that is generic and does not need to be redefined for every new data source selected. Furthermore, the description needs to capture enough semantic and computational detail so that it can support a variety of situation recognition tasks.

We need an event model that fits various data streams into a standard structure that allows various data streams to be used by multiple services, thus making data integration and processing easier. The fundamental question is related to sharing and the integration of data which are separated because of their type or different collection methods. Contextual access and use of these data types are fundamental, as well as location-based services for which the objective is to increase data and service utility, achieve contextual and location-based usability, and share a standard metadata specification. Based on this, the following research questions are generated:

- How to frame complex data streams from different data sources and types?
  - When is schema structure (event model) more appropriate for event streams?
  - What is the most suitable event model to represent events?
  - What event model proposed to tackle this problem?
  - How to identify automatically relevant data streams?
  - How to integrate incomplete event streams?
  - How to consider event semantics analysis?
  - How to handle scalability in the event model regarding Event attributes and event data streams?

As we mentioned in chapter 2.1, events can be categorized as low-level events coming from GPS, accelerometers, microphones, or cameras, and high-level events concerning punctuality, traffic congestion, and driver safety. In our use case context, where we consider that the event is something that is happening in the real world at a specific place, at a particular time, and which has a thematic dimension to be captured by a topic name. Every event belongs to the particular type; event types are pre-defined in the application domain of interest.

Overview and design of event model

Event models match metadata fields for the data streams that carry or pertain to events. The most basic metadata fields for event models are time, location, and data type. The model primitives idea is that the core data message is independent of any application. What metadata fields go beyond the primitive? Pongpaichet et al. [112] use approach that data fields should be determined during the "thought process" of setting up a system. We follow this logic when we are specifying "parameters" for our model. Most devices would have the following fields: timestamp, location, several readings associated with the apparatus (e.g., temperature, voltage, acceleration, power). Also, the model needs to provide a mechanism to deal with data duplication, normalization, time-frame detection, geocoding, event encoding, classification, multilingual support, handling contextual features, temporal and geographic information.

We survey the existing event models based on (i) content and (ii) deployment. By (i) content, they are categorized by domain type and data fields they support, while (ii) deployment provides information on how to use them into the existing event-based architectures and systems.

Our model is represented in Unified Modeling Language (UML) and has three main classes (Figure 20): EventSource, EventProfile, and Event.

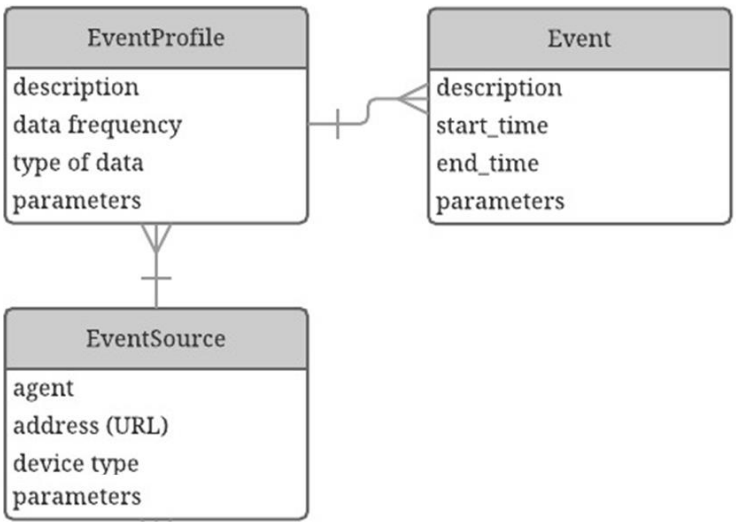


Figure 18: UML representation of the data event model

Each table $\tau = (T, H, p)$ is characterised with $T = table\ name,\ H = attributes\ name,\ p = relations$. Where T = {EventSource, EventProfile, Event}, or $\tau_i = (T_i, H_i, p_i), for\ i = 1,2,3.$

The *EventSource* class represents the data related to sensors and devices. It has the following characteristics: ”agent” that has information for the event source, such as the weather channel or county police. “address” is the address from where data was collected, “device type” is for the device type, such as temperature sensors, mobile devices, and so forth, “parameters” are for the additional information related to the sensor or device, such as serial number, model, and battery expiration date.

The *EventProfile* class represents metadata about the data event stream. The “description” field provides information about the data that is being stored for services. The “data frequency” presents the expected data stream rate to be received, whereas the “type of data” provides information about the collected type of data. The “parameters” can characterize any additional information related to the data event stream that is important, like measurement type, or expected states.

The *Event* class represents the data related to event occurrence, such as rain, high heart rate, pedestrian violation, or car incident. It has the following characteristics: “description” describe the events. “startTime” represents the date and time when the event started, “end time” represents the date and time when the event ended. Parameters for the Event represent additional information that is helpful for a particular service or contextual rule. For instance, the expected frequency of “startTime” and “endTime” can be set by adding a parameter to the *EventProfile*. The events could be in original form, or they could be previously aggregated and then sent at some predetermined frequency.

Location metadata is essential, but it is not presented in the diagram because for some cases it can change depending upon the service. For example, in case of stationary devices, “location” is stored with the device information, whereas for mobile devices, it is required to correspond to events continually. Metadata fields within the three classes of our event model are defined below. Some services require only its specified metadata fields, whereas other services require a consideration of which parameters are relevant. *EventSource* and *EventProfile* send their data at model initialization, whereas the *Event* class sends data continually. Data from the Event class is filtered using rules or machine learning into instances relevant to the service.

For example, Figure 21 presents event model for weather data stream, which is considered as semi-structured data input. We also tried other mentioned data sets like non-structural data sets (e.g., police reports and community events).

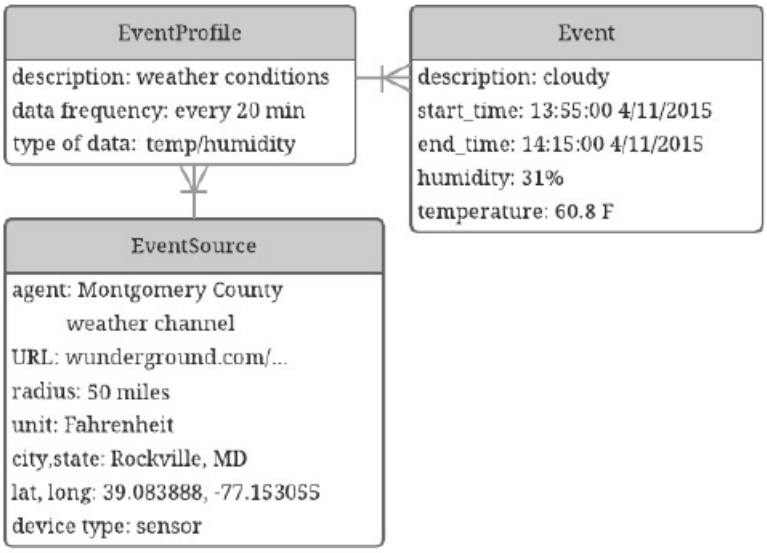


Figure 19: Event model metadata for weather data stream

Alternatively, represented by formal language:

$$\tau_1 = (T_1 = EventSource, H_1 = agent; URL; radius; unit; city_{state}; lat.long; device_{type}, p_1 = relation_{single})$$
$$\tau_2 = (T_2 = EventProfile, H_2 = description; data\ frequency; type\ of\ data, p_2 = relation_{single})$$
$$\tau_3 = (T_3 = Event, H_3 = description; start_{time}, send_{time}, humidity, temperature, p_3 = relation_{single})$$

If the information is created by combining multiple data, to keep track of the process of information and to be able to select the sources of information accurately, it is necessary for the information to be annotated [27]. Semantic annotation helps to describe better and use the related quality parameters of city data in this case. Semantic models provide interoperable descriptions of evidence, and of their quality and provenance attributes. To make semantics scenario independent and to be able to annotate fast and the process, we use upper-level and lightweight semantic models. Moreover, as the data parameters of the data sources update, the changes can be linked to their semantic descriptions. So the processing applications can assess the semantic descriptions to determine the quality parameters of the data descriptors. For complex data that are integrated from multiple sources, provenance parameters can help to trace the variety of information on each origin and quality aspects of the processing algorithms and methods that are applied to the data.

The metadata fields in our model, align with those in the DOLCE ontology. Some alignments by entities are presented in the following table:

DOLCE	**EM**
Social Agent	Agent
Physical Quality	Device type
Physical Region	EventSource-Address
Feature	Event-Profile Description
Temporal Region	Data frequency
Process / Achievement / State	Event-Description
Temporal Quality	Time
Physical Region	Event-Location

Table 6. Aligning Event model with DOLCE ontology

Event description fields can be in alignment with Process, and Achievement and State it depends on the type of the events of interest. The relation between entities can be defined based on the functionality between two entities, for instance, the relation between EventSource and EventDevice is it ‘hasComponent,' and between EventDevice and Event is it ‘includesEvent.'

In this part, we defined and described the unified event data model. The next section explains how the data streams automatically fit into the model and how we can extract the outgoing event streams.

Modeling rules for inbound and outbound data event streams

The process of modeling complex data streams has five steps (see figure 22). Step 1 is getting the data into the data event model, Step 2 is structure events in generic data format, Step 3 is extracted events of interest, and Step 4 is separated into two sub-steps, visualization, and prediction and visualization. Step 4 is divided into two phases because it depends on the application requirements, some of them require making future event predictions.

The next part explains Step 1 and Step 3 in more details.

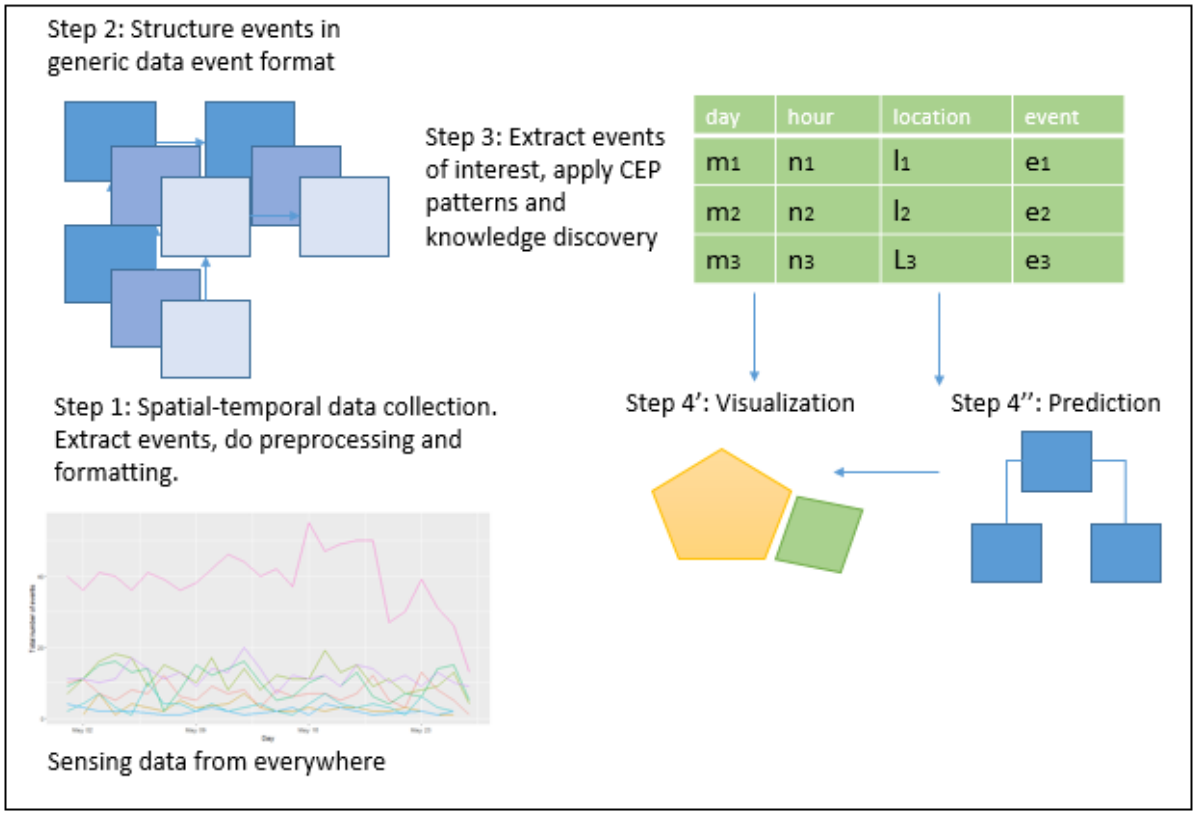


Figure 20: Event model data-flow diagram

Modeling incoming data streams

Getting data into the model includes event detection and matching metadata field names. Therefore, to identify the right events some of the data enrichment methods can be used, such as classification algorithms mentioned before in Chapter 4.2 or using some of the recent Named Entity Recognition (NER) tools that can extract people, organization, location, and other objects that are proper nouns. However, when the data flow does not include metadata, or when it is a natural language stream than Natural Language Processing (NLP) tools will employ data extraction and efficient preprocessing as the one explained in chapter 4.2.

However, heterogeneous of events that arrive from various sources like sensors and social media have inherent uncertainties associated with them. This is one of the challenges of dynamic event modeling that we are trying to solve. For instance, very often we can face incomplete streams regarding time and location of the events. Our approach is to meet these challenges by creating rules that handle known uncertainty at the attribute level. We create rules for handling the common types of uncertainty, that may be found in event processing. For example, instead of a city name field, there is a location field with longitude and latitude value.

We choose rule-based method because it is fast, which makes it applicable to many types of systems. We focus on handling the uncertainty in cases of insufficient event dictionaries, erroneous event recognition or certainty in the event input and uncertainty in the composite event pattern [14], or according to [55], it is uncertainty regarding event attributes. Like the previous example with the location, if some of the attributes are missing we created a rule finding an alternative attribute that will give as the value. Alternatively, represented by conditional statements, such as

$$if\ (\ attribute(city, state)\ from\ data\ source\ (Weather\ Channel)\ is\ missing)$$
$$than\ use\ attribute\ (lat, long)$$

Also, we use probabilistic methods for the usability challenge, with the focus on specific application.

Modeling outgoing data streams

Outgoing data streams are results of applied event design patterns, like aggregation, join, correlation, filtering, pattern matching and so forth. Complex event processing applications correlate data streams as events occur, by using pre-defined rules to identify events of interest. Event stream processing deals with the task of processing streams of event data with the goal of identifying the exact pattern within those streams, employing techniques such as detection of relationships between multiple events, selection, projection, join, event correlation, event hierarchies, and other aspects such as causality, membership, and timing.

For example, if we have two events of interest and we want to use join and selection design patterns, we can formally represent them as follows,

Let $\tau = (T, H, p)$ be a table and C conditions on H, the output obtained by C-selection is the table $\tau_C = \big((T\ where\ C), H, (p\ where\ C)\big)$ where the relation p consist of all tuples that satisfy the condition C.

Let $\tau_1 = (T_1,\ H_1,\ p_1)$ and $\tau_2 = (T_2,\ H_2,\ p_2)$ that have attributes in common. The natural join of the tables $\tau_1\ and\ \tau_2$ is the table $\tau = \tau_1 \boxtimes\ \tau_2 =\ ((T_1 \boxtimes\ T_2), H_1, H_2, p_1 \boxtimes p_2)$.

Concrete case is:

$T_1 = Event, T_2 = EventProfile,$

$H_1 = \{\text{description}, \text{humidity}, \text{temperature}\}, H_2 = \{description\},$

$\text{C} = \{\text{start\_time} = 3/11/2015, \text{end\_time} = 4/11/2015\}$

$$\tau = ((T_1 \boxtimes\ T_2), H_1, H_2, (p_1 \boxtimes p_2)\ where\ C)$$

$$\tau = \begin{pmatrix} (Event \boxtimes EventProfile), H_1, H_2, (\text{description}, \text{humidity}, \text{temperature}), \\ where\ description =' wather\ conditions' \end{pmatrix}$$

Another very used design pattern is correlation coefficient. It is a number that quantifies the statistical relationship between two or more random variables or observed data values [22]. The correlation function is a function that gives the statistical relationship between the variables considering the spatial and temporal distance between them. For instance, if X(s) and Y(t) are random vectors of events with n elements the correlation function is

$$C_{i,j}(s,t) = corr(X_i(s), Y_j(t))$$

Equation 15: Correlation function between two random vectors

Correlation functions are a useful indicator of dependencies between events and can be used as a basis for creating interpolation rules. There are several types of correlation coefficients: Pearson, Rank (Spearman's, Kendall tau, Goodman and Kriskal's gamma), Interclass correlation, and Chi-Square. For categorical data, Chi-Squared is used, while the rest of them are used for continuous data. The most common are Pearson correlations, and it is good when the variables have a linear relationship, while for non-linear there is another method like Rank. For the analysis, we applied spatial correlation which is defined per city and zip code.

In the context of sensor data, the essential characteristics are that nearby sensor nodes probably register similar values. Distance correlation is a measure of the statistical dependencies between two events of vectors of events. It is calculated by dividing their distance covariance by the product of their distance standard deviations. The choice of correlation function and patterns depends on the requirements defined by the application scenario.

The output results of this phase can be visualized and presented to decision makers or multisensory prediction algorithms used for making future assumptions.

Event model in event processing architecture

In complex event processing applications or services, data streams are containing events produced by resources such as people, devices or sensors. They are filtered for event changes of interest, called instances. The diagram in Figure 23 shows event trace in typical Complex Event Processing (CEP) architecture. Data is received from multiple event sources, events are detected, and an appropriate response is triggered. The sequence diagram presented in Figure 23 resembles the official event-driven architecture proposed by Fujitsu [55], Microsoft, IBM, and Oracle.

[22] http://www.ncme.org/ncme/NCME/Resource_Center/Glossary/NCME/Resource_Center/Glossary1.aspx?hkey=4bb87415-44dc-4088-9ed9-e8515326a061#anchorC

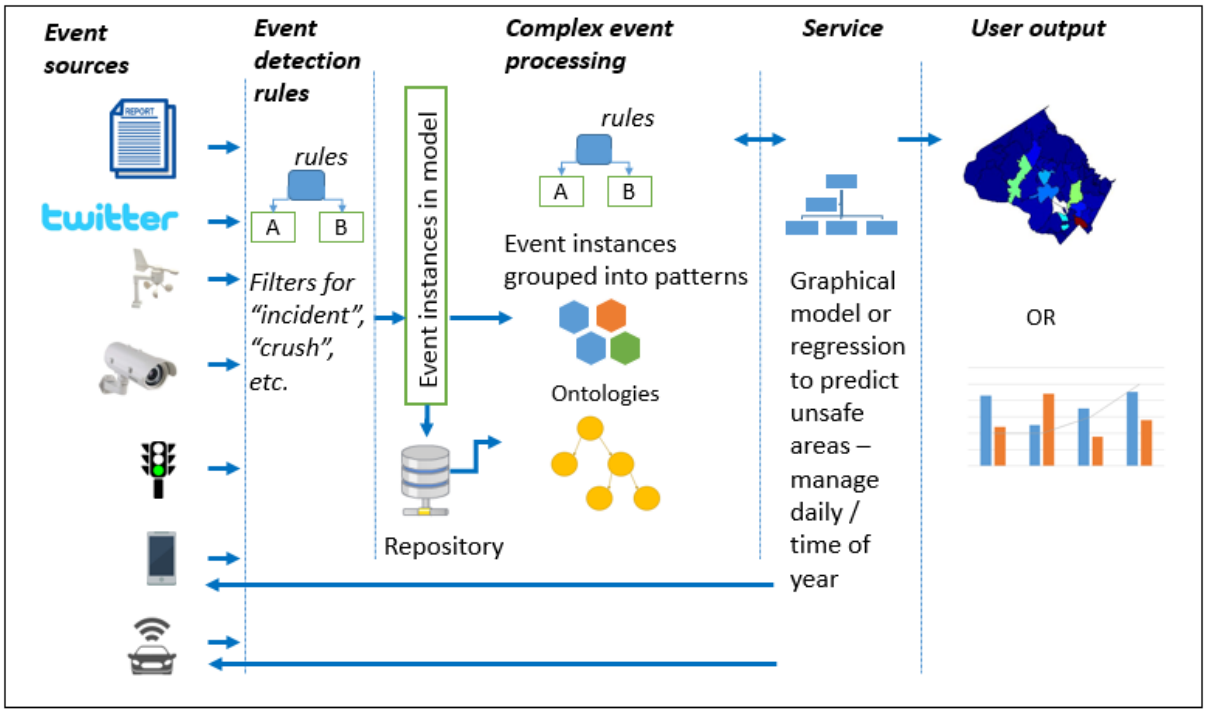


Figure 21: Event trace sequence diagram

Complex event processing architecture, with arrows showing event-data sequence direction. An event model is a component. Arrows indicate event stream flow.

On the left side are various sensors such as reports, temperature, cameras, traffic light, mobile phone, smart cars, social networks and so forth. All of them send information to the system for processing. The first step is using event detection methods like rules to filter the true events, as well as to detect any uncertainties. Previously in modeling rules for incoming data streams, we explain this part in more details.

On the right side are data processing and visualization. Note that the vertical box in Figure 23, "Event instances in the model," where the data is organized to be stored in a database server or event cloud, is central to the process. Data processing and visualization contain the methods we describe previously in modeling outgoing data event streams.

The event model can be part of the CEP architecture. It can be used in a variety of technology platforms, like relational and non-relational management systems, and stream software. Therefore, it is not dependent on technology and language, since it only presents the design concepts and logic.

Evaluation metrics

We evaluate the proposed solution by splitting the input data on training 80% and testing 20% for each weather season, like Winter (from January to March), Spring (from April to June), Summer (from July to September, and Fall (from October to December). The output results are presented in a confusion matrix and the performance metrics used to evaluate the classification results are accuracy and standard deviation. Accuracy is calculated using the Equation 13: Accuracy based on confusion matrix and Equation 14: Precision, Recall, F1 - metrics based on confusion matrix. While standard deviation is calculated using the following formula

$$\boldsymbol{\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}}$$

Equation 16. Standard deviation measure

The choice of evaluation metrics depends on the algorithms used for making predictions or other calculations, plus from the precision of the event detection method.

We better result interpretation we visualize the results using a geographical map.

Discussion

In this sub-chapter, we have presented event model that describes sensor information including location attribute, observation object, time and event, it fits different data types into the same schema, thus making data integration easier. The presented model requires a minimum of metadata fields, and it is efficient, minimizing the amount of network traffic by linking a continuous time-dependent data stream to non-time dependent meta-information stored on an event server or cloud. We also explained how the overlap among different types of evidence sources could be handled by an upper-level ontology like DOLCE. The data field types in our model are aligned with DOLCE categories to facilitate getting data from different streams into the same pattern. We also have discussed how the users can deploy the event model into the existing event processing tools, fill it with data and extract the data of interest from the model.

The presented solution has functionality for event detection and uncertainty before data goes into the model. We demonstrated data modeling and show flexibles rules to guide the deployment of our event model in a city service.

In the next subsection, the multi-model predictive algorithm adaptable to data stream changes such as network leakage, power down, sensor battery low and so forth is presented. This resilience characteristic enables continuity of service for proactive time prediction.

4.4. Dynamic network model

Smart city design seeks to optimize city services, improving the resident experience and reducing waste, through intelligent use of citywide data. Smart city services are expected to respond appropriately to changing conditions, requiring regular data updates on the status of citywide properties, such as weather, road conditions, and infectious disease case numbers. Consequently, optimal deployment of critical resources will depend on data streams. For example, in the event of a disease epidemic, current medical statistics will be used to ensure that ambulances, drugs, and vaccine resources are intelligently distributed to the worst hit neighborhoods and those predicted to be at high risk. In the case of a powerful storm that disrupts traffic, traffic data will be used to distribute police resources for traffic guidance based on current and predicted traffic patterns. In the case of changing crime rates throughout the city, crime statistics and predictions will be used to ensure that police, medical, and emergency resources are intelligently distributed to reduce response time. As city services become more dependent on smart city data streams, the services also become more susceptible to disruptions in the data streams. Such disruptions can affect critical services, for instance, by increasing ambulance response time. Interruptions in data streams and the resulting loss of data can occur for any number of reasons, including anomalous signal to noise reduction, power loss during data collection, or data loss on a network level, either benign or malicious. Additionally, due to the spatial and temporal dependence of smart city data streams, with data collected periodically by distributed sensors or local human-based reporting, spatiotemporal events can impact data service. For example, a storm can knock out neighborhood-wide communications, interrupting data collection as the storm travels from one location to the next. For these reasons, smart city services require a level of resilience to data stream disruption.

To improve smart city applications, resilience to data loss, such a scheme must be implemented. As data disruptions occur and data loss is identified, the lost data is estimated with minimal prediction error to reduce the impact of the data loss on dependent services. Based on this we identified these research questions:

- How can a predictive model based on knowledge be created?
    - How do we efficiently identify relationship links between data streams?
    - What is an appropriate prediction model?
    - How can the prediction model be adapted?
    - How to integrate scalability for the model that you can choose?
    - How to graphically present the network dynamics and relationship between data streams for decision makers from city representatives?

To find the answers, we propose an application-layer algorithm that can be used to ensure resilience across regions of different scale, from smart neighborhoods to smart countries, and establish both inter- and intra-smart city networks.

Design phases of dynamic network model

Sets of multivariate, spatiotemporal smart city data streams such as the status of multiple traffic lights, the number of locally available vaccine units, and neighborhood air quality are represented by $\boldsymbol{Y} = [\boldsymbol{y}^{k=1}, \boldsymbol{y}^{k=2}, \dots, \boldsymbol{y}^{k=N}]$, where the superscript $k \in \{1, \dots, N\}$ provides the data stream index for a set of N data streams. For spatial data, each index k corresponds to a location. Individual data streams are represented by a time-series vector $\boldsymbol{y}^l = [y^l_{t=0}, y^l_{t=-1}, \dots, y^l_{t=-v}]$ with the subscript t providing the time series sample index, beginning at the time of interest to be predicted $t = 0$ and extending to *v* periods in the past $t = -v$. An individual data stream $\boldsymbol{y}^l$ thus has dimensions $\mathbb{R}^p$ and the set of N data streams $\boldsymbol{Y}$ has dimensions $\mathbb{R}^{N \times p}$. A snapshot of the state across all streams at time t is given by $\boldsymbol{y}_t = [y^1_t, y^2_t, \dots, y^N_t]$. For this work, we assume that all data streams are simultaneously sampled at regular time intervals. Data loss in a data stream is indicated by the absence of data at a time, $y^l_t = \emptyset$.

We created a dynamic network-based model that provides improved and reinforced resilience to data loss. The VAR-based mode uses past data from the data stream of interest as well as data from 'related' data streams – streams that share temporal trends, to achieve optimal estimation accuracy. The model dynamically identifies the optimal set of data sources to reinforce the resilience of each data stream with Granger causality and MDS analysis. Model dynamics is achieved through recurrent updates, which identify the optimal network for each data stream to maintain optimal estimation accuracy.

An example is shown in Figure 24. Three data streams are presented with their values indicated for times t_{-4} through t_5. At time t_0, the data stream of interest $\boldsymbol{y}^1$ experiences data loss. (Now the data for times t_1 through t_5 have yet to be collected.) Resilience in the data stream can be established by estimating the lost data using auto regression (AR) – extrapolating the value of y^1_0 from past data. Alternatively, if either available data stream $\boldsymbol{y}^2$ or $\boldsymbol{y}^3$ shows similar trends to data stream $\boldsymbol{y}^1$, information from that stream can be used to improve the estimate of y^1_0 using VAR. During the period of $t = \{-4, \dots, 5\}$, there are four potential resilience models which can be evaluated for their utility in reinforcing estimation of y^1_t:

1) AR using only data stream $\boldsymbol{y}^1$: $\hat{y}^1_0 = f(\boldsymbol{y}^1)$
2) VAR using data streams $\boldsymbol{y}^1$ and $\boldsymbol{y}^2$: $\hat{y}^1_0 = f(\boldsymbol{y}^1, \boldsymbol{y}^2)$
3) VAR using data streams $\boldsymbol{y}^1$ and $\boldsymbol{y}^3$: $\hat{y}^1_0 = f(\boldsymbol{y}^1, \boldsymbol{y}^3)$
4) VAR using all three data streams: $\hat{y}^1_0 = f(\boldsymbol{y}^1, \boldsymbol{y}^2, \boldsymbol{y}^3)$

Data streams 2 and 3 are first tested for shared trends with data stream 1 using the Granger causality test and MDS to determine the viability of models 2-4. If shared trends are identified, the models are gauged for their performance at each period, performance ranks the models, and the best performing model is selected for implementation. If instead a supporting data stream is found not to provide utility, that data stream can be removed from the prospective analysis, reducing the amount of data traffic required for the network. In this example, model 2

is determined to provide optimal performance for period $t = \{-4, \dots, 0\}$ and once data has been collected for $t = \{1, \dots, 5\}$, model 3 is found to provide the best performance for this period. Models 2 and 3 are graphically represented by a directed graph, with edges connecting from the node representing $\boldsymbol{y}^2$ or $\boldsymbol{y}^3$ to the node representing $\boldsymbol{y}^1$ (See Figure 24). If an issue is identified with the optimal resilience model, e.g., the supporting data stream stops reporting, then the next best performing model is chosen, and so on.

The set of all top performing, concurrent models for all data streams composes the resilience network. The network is represented by the resilience network graph – the collected graphical representation of all concurrent models. A user determined intervals; the system is iteratively updated, re-evaluating the performance of each model to update model rankings and identify and implement the optimal model. This provides dynamics to the resilience system, allowing it to run independently and self-adapt to issues in the data streams such as a reduction in data stream quality, the loss of an entire data stream, or the addition of a new data stream that may contribute to higher performing resilience models.

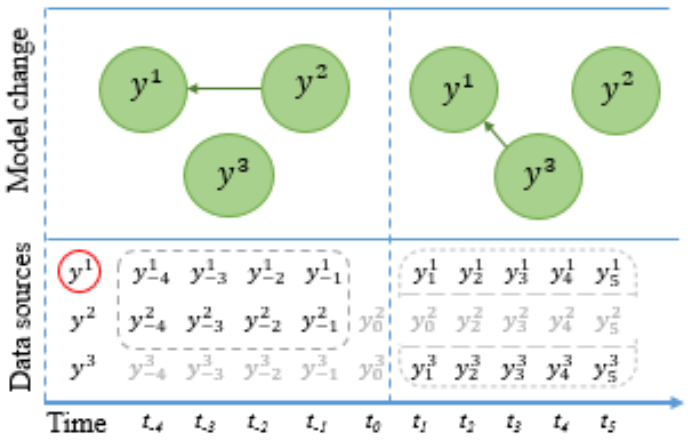


Figure 22: An illustration of three event-based data sources y_1, y_2, y_3 and dynamic model adaptation over time $t_{-4,\dots,}$ t_5 depending on data stream changes.

The resilience network methodology is diagramed in Figure 25, with each step explained below. The system begins with preprocessing the data streams, followed by relationship analysis for sets of streams. Relationship analysis is performed to reduce the search space for optimal resilience models, as described below. Potential resilience models are then evaluated, the optimal models are selected, and the network is identified. These steps are iterated at user-determined intervals to maintain an updated, optimal resilience network. Qualitative analysis is also used to determine the correlations between the network and any pertinent data relating the data streams. Through this qualitative analysis, additional

information sources can be used to reduce the search space of potential resilience models and subsequently reduce computation time and cost.

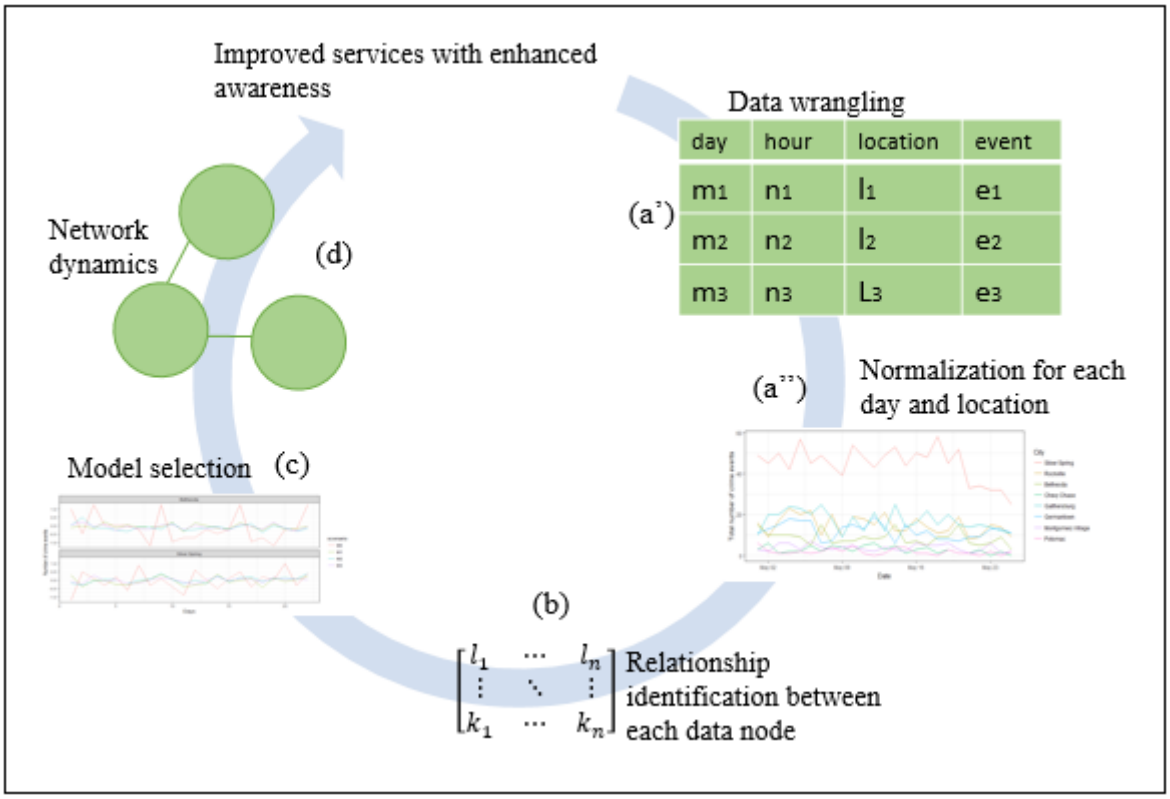


Figure 23: Overview of the proposed solution

Data analytics framework with periodically based iterations. The right side represents the first two steps which constitute the data preprocessing (a) step: (a') data wrangling and (a'') data normalization. Next step is data stream relationship analysis (b), used to identify the streams that share temporal trends and narrow down the hypothesis space of potential data sharing models for the network. Step (c) is to determine the list of best models for analysis based on minimum prediction error, and step (d) is dynamic implementation of the set of best models and available resources

a) Data Preprocessing

Data preprocessing can involve many steps including data cleaning, unifying formats and metrics, feature extraction, and feature vector normalization. The choice of pre-processing methods is application and data dependent. A description of the particular techniques used for the case study can be found in chapter 5.4.

b) Data Stream Relationship Analysis

i. Quantitative Analysis

Once data preprocessing is complete, relationship analysis is performed on each pair of data streams to identify streams with shared temporal trends. For each data stream of interest, the reduced set of 'related' data streams can then be used to narrow in on potential resilience models. This approach can significantly reduce search time, and computation cost as the original resilience model hypothesis space for each data stream includes all models covering the range of possible dependencies on all other data streams. For this work, the Granger causality test [72] is used to identify whether one data stream can be used to improve prediction estimate accuracy of another data stream due to shared temporal trends. More specifically, the null hypothesis of no causal relationship is investigated with an F-test, and the resulting p-value is compared to a threshold to identify whether the null hypothesis can be rejected. Here, 'causality' is a misnomer, as the method does not identify causality between data stream sources, and instead implies predictive causality. The method does not take into consideration the possibility that both data streams are consequences of the same cause, i.e., the existence of latent variables that Granger-cause both data streams of interest.

The Granger test is used rather than a more common correlation metric such as Pearson's product moment as it indicates the statistical significance of using past values of data stream $\boldsymbol{y}^m$ to assist in predicting $\boldsymbol{y}^l$ rather than using past values of $\boldsymbol{y}^l$ alone. Identifying these relations can assist in identifying possible underlying relationships between the data sources, which can also be used to improve resilience models. For this work, bidirectional causal relationships were tested between each pair of data streams. As a pre-processing step, each data stream was first confirmed to be stationary by use of the Augmented Dickey-Fuller (ADF) and Kwiatkowski Phillips Schmidt Shin (KPSS) unit root tests. In evaluating the Granger casual relationships, the lag parameter was programmatically selected using the Akaike information criterion (AIC), ensuring a dynamic response from the system.

The Granger test provides information about the relationship between a dependent and independent data stream. For this work, models are also used which rely on two separate data streams. Identifying the predictive causal relationship between one dependent and two independent variables can be performed using the multivariate Granger causality test which is reliant on the results of VAR analysis. Thus, in the first iteration, the prediction accuracy of all relevant VAR models $\hat{y}_0^l = f(\boldsymbol{y}^l, \boldsymbol{y}^m, \boldsymbol{y}^n)$ can be computed and the field of potential models whittled down for future iterations by subsequent multivariate Granger analysis. Using this method can greatly reduce the hypothesis search space for resilience models, as a set of ten data streams results in a hypothesis space of 10 AR models, 90 VAR models with one independent data stream, and 720 VAR models with two independent data streams.

ii. Qualitative analysis

Qualitative analysis can be used to determine if underlying latent parameters dictate the relationship between data streams. If such parameters are found, they can be used to reduce the hypothesis space of possible resilient models, thus reducing computation cost and required data

sharing network traffic. For this work, the multi-dimensional data scaling (MDS) method was used to visualize the relationship between potential descriptive variables and resilience model performance. MDS operates by mapping points from a high dimensional Euclidian space to a lower dimensional space while attempting to preserve dissimilarity relationships between the points. For the case study, geospatial topological and demographic parameters are investigated for their utility in predicting resilience model accuracy.

c) Model Selection and Evaluation

The next step is identifying and ranking resilience models by prediction accuracy. For this work, the hypothesis space of resilience models is limited to linear AR and VAR models with one to three independent data stream variables. Linear AR and VAR models were chosen due to their ease of computation and interpretation for dynamic multivariate time series, as well as their availability on scalable big data platforms. For N data streams, the set of possible models include:

1) N AR models using past data from the stream of interest
2) $(N^2 - N)$ VAR models using past data from the stream of interest and a supplemental data stream ('two-city')
3) $(N^3 - 3N^2 + 2N)/2$ VAR models using past data from the stream of interest and two supplemental data streams ('three-city')

The three model types can be expressed by the time series p-th order VAR equation which uses p past data stream values:

$$\hat{y}_t^l = c + \sum_{k=\{l,S\}} \sum_{n=1}^{p} \beta_{t-n}^k y_{t-n}^k$$

Equation 17: Three model types based on VAR equation

where $\hat{y}_t^l$ is the approximation for the missing data value y_t^l, c is a constant, and β_{t-n}^k is the auto-regression weight learned for data value y_{t-n}^k for data stream k and time t-n. k is summed over the set of data streams to be used in the approximation analysis including the data stream of interest l and the set of supplemental data streams S. For model type one, simple AR, S is the empty set and regression is performed over only the past values of y^l. For model types two and three, S is composed of the one or two supplemental data streams, respectively. The order of the VAR model used, p, is dynamically determined by AIC.

Model evaluation is performed using time series cross-validation, and performance is measured using mean square error (MSE). Time series cross-validation is selected over

generalized cross-validation as it provides better estimates of model prediction performance. For each run of the cross-validation, testing is performed on the value y_t^l, for each possible t, and training is performed on all possible sets with target y_{t-r}^k and independent inputs $y_{t-r-q}^k, q \in \{1, \dots, p\}$. MSE is computed over the set of $\hat{y}_t^l$ estimated. Ranking model performance is achieved by comparing the MSE for each model to the AR model MSE for the same target data stream. This emphasizes the improvement in prediction performance provided by the model of interest relative to the baseline of AR. The formula used is :

$$RelMSE = 100 * \frac{MSE(f^{AR}) - MSE(f^i)}{MSE(f^{AR})}$$

Equation 18: Improvement of prediction performance using MSE

d) Network Dynamics

As trends change in the data streams, the network should respond dynamically, self-adapting and reform the network connections to maintain optimal performance. For example, a weather sensor network should respond appropriately as a storm travels from one neighborhood to another. If a sensor experiences data loss, the supporting sensor data used to reinforce resilience should be from those currently experiencing similar weather patterns. Network dynamics are introduced by iterating network evaluation on a user-defined interval, ensuring that network connections reflect current data stream trends. Network re-evaluation is diagramed in Figure 2, starting with data pre-processing, followed by a performance of relationship analysis for data stream sets, a ranking of models by performance and the implementation of the optimal round of patterns in the current resilience network. Additionally, if an anomaly in the network is detected, such as the loss of a networked data stream or the addition of a data stream, the network can dynamically select the next best models to replace those affected, or a reevaluation of the network can be triggered. In implementing such a system, a delay may be necessary to improve system stability, reducing the likelihood of rapidly alternating between models due to small variations in data. Network reevaluation can also be triggered based on an external signal ensuring user control or interaction with a relevant event detection system.

Discussion

This approach suggests a solution to the following challenges, like merging dynamic with static information sources, continuous processing, scalability regarding narrow down the data source selection (data sampling), and distribution in a sense minimizing data transmission among the units and modularizing reasoning. Also, it provides dynamic selection of data

streams. We validate by using crime events data from police reports. The overall idea aims to improve the local services to maintain the service even when the data is in an unstructured format, uncertain and dynamic. It was shown that an event could be related to other events by time, causality, and geo-location. Most of the readings are autocorrelated, for instance, the temperature reading is softer profoundly affected by an earlier time step's reading. Some of the machine learning algorithms do not consider autocorrelation, and they will do well while predicting this type of data stream; therefore, we choose a time series analysis model. One of the main types of uncertainty that may be found in incoming event streams is incomplete or missing data streams, corrupt data and pattern uncertainty. Incomplete or missing information is when the sensor fails to report specific events due to some hardware malfunctions or network leakage. The event may have a noise component added, this corruption of the input stream can be caused by the limited accuracy of sensors or distortion along a communication channel. Lack of knowledge or due to the inherent complexity of a domain, it is sometimes impossible to capture precisely all the conditions that the pattern should satisfy.

4.5. Conclusion

By taking advantage of the event processing and predictive technologies, we gather, filter, categorize and store the event related information from the desired location that will finally be presented to the user in the form of a list or on a map. We designed a framework that efficiently handles some of the challenges of complex data streams that comes from the dynamic environment. We describe each functionality in details and present the evaluation metrics.

For instance, network text analysis triggered the rise of SA which brings new possibilities to city government in general and decision-making [2]. SA can contribute to a better understanding of and appropriate reactions to the public’s needs and concerns by city governments. Measuring the sentiment at specific areas and topics help to determine the relevant services for the users and promote relevant recommendations (content, collaborative, or hybrid filtering) based on that.

We presented the event model that integrates complex data streams from various formats in standard format. At the same time, it is underlined with upper-level ontology which adds an additional interoperability component. An event model with its scalability and flexibility properties facilitates data sharing among city services.

We presented dynamic network model that adds resiliency properties to city services so that they can run even in the cases of network leakage. It supports static and dynamic data streams from multiple data sources and various data types (as well as data source availability and frequency changes). Based on prediction error and relationship analysis, it provides a list of the best models that can be used in case of data loss. It also identifies a dynamic data sharing network between independent, smart cities or generalized smart communities to ensure minimal

data estimation error for each smart city. With each smart city, it is assumed to be associated with a single multivariate data stream. Inter-smart city networks are of interest for county-wide occurrences such as the spread of epidemics, while intra-smart city network can provide resilience for the scenarios as traffic management.

The next chapter presents the implementation of the described solutions. We carried several experiments for each of the challenges to evaluate the accuracy of the framework.

Chapter 5

Evaluation, Results, and Discussion

"*Discovery consists of seeing what everybody has seen and thinking what nobody has thought.* "
- Albert Szent-Györgyi

Summary. Several experiments evaluate the effectiveness of the proposed framework solutions with different settings. Section 5.1 is a summary of the data set that characterizes cities today; describing the process of data collection, period, metadata properties and pattern description. Section 5.2 outlines the steps to set up the experiments and presents the results of the proposed model for automatic event network analysis with the aim to show that event detection in combination with other algorithms can help with better understanding of cities. This chapter also lists the results of comparing existing event classification algorithms with the different feature selection. Section 5.3 describes the steps to set up the experiments and shows the evaluation results for the semantic event model. Thus showing that various complex data sets can be integrated into the event model. This chapter also demonstrates how the previously described methods for event detection are incorporated into the event model and overall framework. Section 5.4 also describes the steps to set up the experiments and explains the results when the dynamic network model was applied with the aim to deal with data loss and data stream changes over time.

5.1 Data collection, description, and patterns

For the experiments, different types of data sets are used that present various perspective for the cities, such as traffic incidents, weather, demographics and so forth. The data was available online as a part of an open data initiative; all experiments use the data from Montgomery County, Maryland, U.S.A. We collected structured, semi-structured and non-structured data sets characterized by static and dynamic behavior, to deal with the challenges related to data complexity like their format, dynamics, and uncertainty. Following their characteristics are described, and it is intended to give a degree of familiarity with the data and how we collected it.

Stationary datasets are demographics and commute time

Census dataset[23]: Census data was collected for cities in Montgomery County, Maryland, U.S.A. including demographics properties like population count, education degree bachelor or higher, and median household income, see Table 3.

City name	Population	Bachelor's degree or higher	Median household income
Germantown	86395	47.8	86472
Silver Spring	71452	53.4	72289
Rockville	61209	62.3	98530
Bethesda	60858	82.2	145288
Gaithersburg	59933	51.6	78441
Potomac	44965	80.4	181385
Montgomery Village	32032	44.4	77537
Chevy Chase	9545	81.4	159963

Table 7: Demographic properties for Population, Bachelor's degree or higher, and Median household income, measured in 2010

Geospatial topological distance: The distances between cities were collected using the Google Maps[24] service, which also includes the schedule from the Ride On public transportation service for Montgomery County, Maryland, U.S.A. see Table 4.

City name	Silver Spring	Rockville	Bethesda	Chevy Chase	Gaithersburg	Germantown	Montgomery Village	Potomac
Silver Spring	0	10.2	4.6	4	18.1	22.1	19.7	13.5
Rockville	10.2	0	9.2	9.1	5.1	11	8.6	6
Bethesda	4.6	9.2	0	1.5	12.9	18.6	16.3	7.2
Chevy Chase	4	9.1	1.5	0	15.5	18.6	16.4	9.6
Gaithersburg	18.1	5.1	12.9	15.5	0	6	2.5	11
Germantown	22.1	11	18.6	18.6	6	0	6.2	14.5
Montgomery Village	19.7	8.6	16.3	16.4	2.5	6.2	0	12.7
Potomac	13.5	6	7.2	9.6	11	14.5	12.7	0

Table 8: Distance of miles between the cities in Montgomery County, Maryland, U.S.A.

[23] http://www.census.gov/

[24] The identification of any commercial product or trade name does not imply endorsement or recommendation by the NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

Dynamic non-structured dataset is social network like Twitter

Twitter data set: Twitter data is collected using its Application Programming Interface (API). Received tweets using streaming API are anywhere from 1% of tweets to over 40% of tweets in near real-time. Twitter provides two types of location data; one uses the name of the city and other uses the exact Global Positioning System (GPS) coordinates. For the experiments, we choose to use the name of the location and nearby radius because we can consistently collect tweets for each category. We downloaded the data from Germantown, Maryland, U.S.A. within 5 miles radius for the 2015 year, with a total number of 10569 records, see example table 5.

Date	Time	Message
1/1/2015	16:22	I'm at iPic Pike & Rose for Into the Woods in North Bethesda, MD https://www.swarmapp.com/c/fL1Lwd3ABca
1/1/2015	15:39	Started out the New Year with my favorite CU girls in MD @ Baltimore, Md http://instagram.com/p/xUzX3tgH1W/
1/1/2015	20:32	Recovery mode (@ Dona Bessy Pupuseria in Montgomery Village, MD) https://www.swarmapp.com/c/2eSJeZ3mSus

Table 9: Example of Twiter messages

Dynamic semi-structured data set are police reports like crime and traffic incidents

Crime data set[25]: We collected 116375 records related to crime events reported throughout Montgomery County, Maryland, U.S.A. for the period from 1/1/2014 to 5/26/2016 period. Each record has twenty-four attributes including date and time (start, end, police dispatch) of the incident, location (longitude, latitude, zip code, city, state, address), police district name and number, agency, uniform crime reporting number, and description. A plot of the number of crime events occurring in each city during the last month of data for the available dates of May 1st through May 26th (showing the last 26 out of 877 days) is shown in Figure 26. From this figure, it is evident that Silver Spring dominates in several crime events while Chevy Chase, Potomac, and Montgomery Village typically fall near the bottom. This trend is shared throughout the period investigated. The eight cities with the highest number of crime events were chosen for analysis, they are (in descending order) Silver Spring, Bethesda, Gaithersburg, Rockville, Germantown, Montgomery Village, Potomac, and Chevy Chase. The rest of the cities were not considered for analysis as they exhibited a little number of events per day, e.g., between zero and two-day events.

[25] https://data.montgomerycountymd.gov/

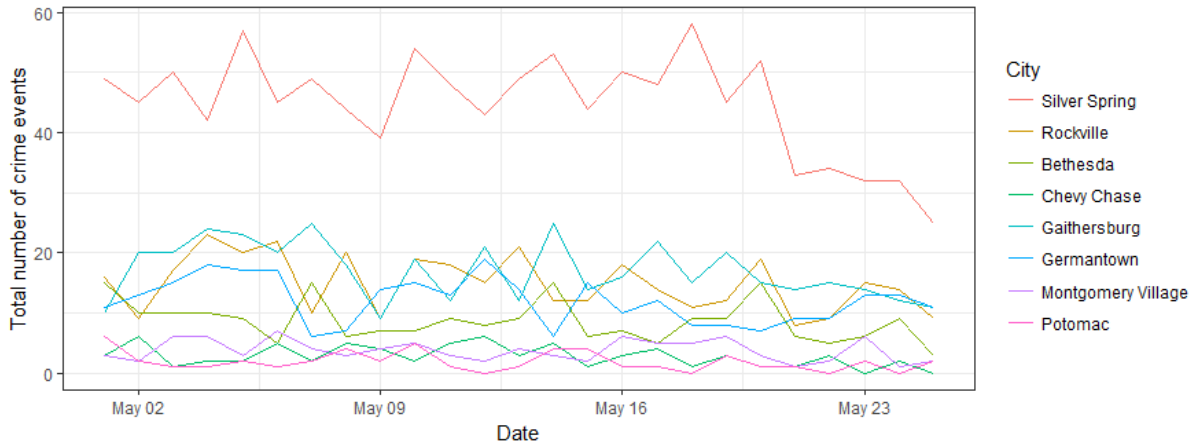


Figure 24: Representation of some daily crime events produced during 01-26 May 2016

Traffic incidents data set: We collected 235 264 records of traffic incidents reported throughout Montgomery County, Maryland, U.S.A. for the period of 1/1/2015 to 31/12/2015. From them, only 2874 were events related to pedestrian safety. Plot representation with the monthly amount of traffic incidents related to pedestrians is presented in Figure 27. Because the dynamic is different for the different period, we will perform periodic analysis (Spring, Summer, Fall, Winter).

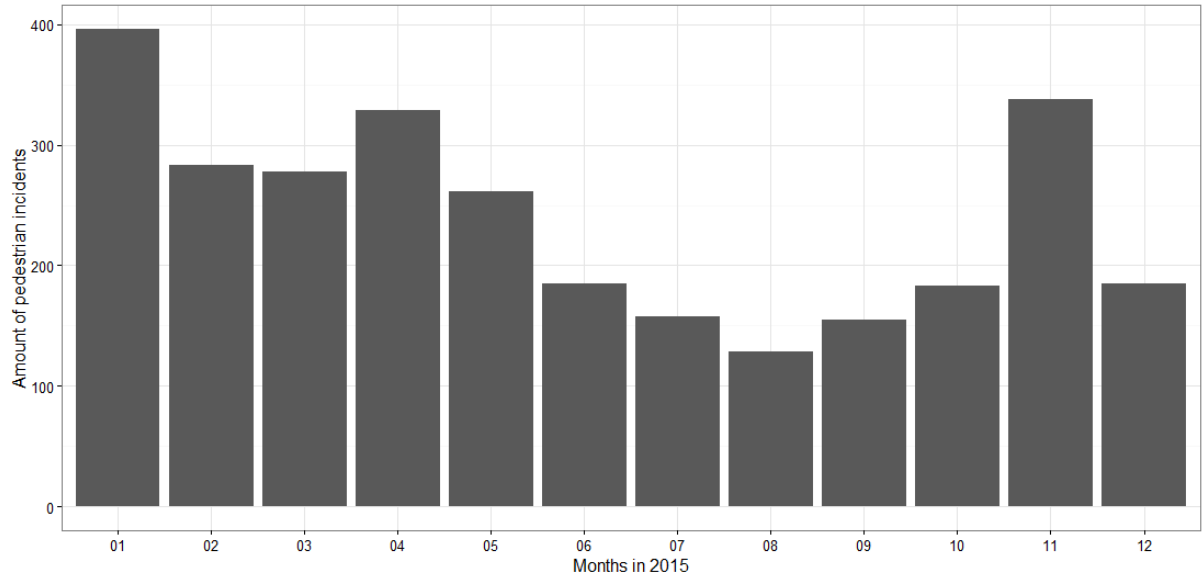


Figure 25: Representation of traffic incident events related to pedestrian safety for the year 2015

Dynamic structured data set is weather and community events

Weather dataset[26]: Daily data were collected over the same period as the crime dataset, for the cities in Montgomery County, Maryland (MD), U.S.A. Each record has the attributes: Temperature, Humidity, Sea Level Pressure, Visibility Miles, Wind Speed Direction, Dew point, Precipitation, and Cloud Coverage. Each attribute is described with Min, Mean, and Max features. For this work, only the daily mean temperature and mean humidity is used. Also, Montgomery County is covered by three weather centers (College Park Airport, MD, Ronald Reagan Washington National Airport, VA and Montgomery County Airpark, MD). For this analysis, data from the College Park Airport, MD was selected because it covers most of the cities under investigation.

Date	**KGAI**		**KDCA**		**KCGS**	
	Mean TemperatureF	Mean Humidity	Mean TemperatureF	Mean Humidity	Mean TemperatureF	Mean Humidity
1/1/2016	36	59	42	58	38	62
1/2/2016	36	55	40	54	39	71
1/3/2016	37	60	43	59	40	70

Table 10 Example representation of the original weather data

Community happenings data set: The data was collected for the period of 1/1/2015-31/12/2015. Each record has the attributes: Event name, Date, Time (From-To), Address, Zip Code, and Category. Table 7 gives example representation of the dataset.

Description	**Date, Time**	**Address (Street name, number, City, Zip code)**
Volunteer Recognition	Tuesday, November 24, 2015, 1:30 – 3 pm EST	White Oak Senior Center1700 April LaneSilver Spring, MD 20904
Wednesday Library Trip	Wednesday, August 5, 2015, 9:30 – 11 am EDT	Schweinhaut Senior Center1000 Forest Glen RoadSilver Spring, MD 20901
Wesak or Buddha Day	Monday, May 4, 2015	Montgomery County, MDBethesda North Marriott Hotel & Conference Center, 5701 Marinelli Road, North Bethesda, MDSchweinhaut Senior Center

Table 11 Example representation of original dataset for community events

[26] www.wunderground.com

The crime, traffic, weather, and social network datasets are characterized with regular daily updates, while census, distance in miles, and community events have regular updates determined individually by their policy.

The Twitter dataset, which is marked as a dynamic non-structured data set is used for evaluation the automatic event network analysis use case for improving the public city service case study. Traffic incidents, weather, and community events were used for evaluation of the event model, while crime, weather, demographics and distance in miles were used for the evaluation of dynamic network model use cases.

Also, these data sets have been approved by the NIST IRB [27](Institutional Review Board) review process following the requirements by the NIST Human Subjects Research Determination Form.

Experiments we performed were in an R statistical programming environment, using various packages for each of the functionalities. Details about the libraries and versions are described in Appendix E.

The following sections 5.2, 5.3, and 5.4 explains experimentation done for each of the previously proposed theoretical solutions.

5.2 Methodology: NIST Big Data Framework

There is ongoing work at the National Institute of Standards and Technology (NIST) on defining and prioritizing Big Data requirements, including analytics, extensibility, data usage, interoperability, portability, reusability, and technology infrastructure. The NIST Big Data Public Working Group (NBD-PWG) created a standards roadmap that describes the adoption of the most effective Big Data techniques and technology. This group has a goal to provide standard consensus for some important fundamental questions, related to the essential characteristics of Big Data environments, integration with the existing architectures and difference between traditional data environments.

To describe our case studies, we used NIST Big Data Interoperability Framework V1.0 with the focus on reference architecture and use case requirements. The formal description and details are presented in Appendix C - NIST Big Data Requirements Use Case. We follow this methodology to validate the proposed framework; we have the following experimentation and interpretation of the results.

[27] https://www.nist.gov/content/institutional-review-board

5.3 Evaluation of automatic event network detection

As we mentioned before, in a city context, social sensing can be used to retrieve information about the environment, weather, well-being, traffic congestion, trends in the local economy, dangers or early warnings, likewise any other sensory information that collectively become useful knowledge for the city improvement and smartness. In this case study, each user is considerate as a sensor and tweets are sensor data with the time, location, and topic features. The case study we used is focused on tweets that will result in analyzing the view of the public on generally discussed event topics and measure their perceptions regarding a variety of subjects. The output of this algorithms is intended for modern understanding of the tweets reporting various concerns about the city, which is necessary for municipal authorities to manage city resources.

Experiment set up

The effectiveness of the proposed automated event network detection model is evaluated by performing several experiments with different settings were carried out. We collect dynamic non-structured data set from a social sensor like Twitter from one city geo-location, Germantown, Montgomery County, Maryland, U.S.A. (more details about the process of collecting the data is explained in subchapter 5.1.).

Results and discussion

We followed the functional flow diagram for event type detection presented in Figure 17 and explained in subchapter 4.2. The input adapter from FNEDAP collects the data, after the next step in processing the Twitter messages (tweets), is pre-processing which is part of event detection module. Since it is a text data with a lot of noise pre-processing is a necessary step. As mentioned in Chapter 4.1 for the experiment was chosen sentiment aware tokenization (explained in details in Chapter 4), which makes useful information from each punctuation and non-standards words to increase classifier effectiveness and model portability. We choose this approach because it was reported by Potts[28], as the most efficient compared to the other existing methods like Whitespace and Treebank. The dictionary list that we used to convert the non-standard words to relevant words is presented in Table 8.

[28] http://sentiment.christopherpotts.net/tokenizing.html

Dictionary list name	Number of lines
Smiles and Emoticons	249
ContractorsS	51
Acronyms, Abbreviations, and Initials	736
Misspelling	5921
Stop words	319

Table 12 Dictionary list used for context-aware pre-processing of tweets

Table 8 presents the statistics for to the characteristics from the dictionary list for the data set we used for the experiment.

Statistics/Database Name	Dataset
Tweets	10569
Tokens	132022
Tweet length < 10	32
URLs	8837
Re-tweets	2611
Contractors	3832
Misspell words	674
Punctuation marks	47100
Abbreviations	430
Acronyms	1940
Smiles	86
Stop words	29613
Numbers	3517

Table 13 Data statistics for dictionary content found before pre-processing

Feature space refers to the n-dimensions where variables live represented as R^n. In ML we view all variables are features, or in the case of text analysis like this where each word can be considerate as a feature, we can have more than 1000. Therefore in the next step is feature selection, where each sample is represented as a point n-dimensional space or high dimensional vector. This dimension is determined by the number of features (numeric representation of raw data) used to describe the patterns. Similar patterns are grouped together, which allows the use of density estimation for funding models. Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions. We experiment with Bag of Words (BOWs) and Term Frequency and Inverse Document Frequency (TF-IDF) techniques. While BOWs gives more value to the favorite words, TF-IDF is normalizing the word counts so that the

favorite words are discounted, and for the experiments, we experiment with feature space of max 100, 150 and 1000 features.

After the input text data was transformed to vector space and feature space was selected, we train the data with predefined event type categories. As we defined earlier in chapter 4.2, " event types is a set of event objects that have the same semantic intent, and every event object instance is considerate to be a case of the event type ." Based on the knowledge learned during the training step, the classification system will be able to infer to the some of the event type classes. Figure 28 presents the functional flow of event type classification. Set of predefined categories was used to create a training set of labeled text objects, the categorization module classifies the text object into one or more of the categories, in this case into one of the categories. For the experiment we label tweets to refer to named entities, in this case, named entities used for extraction of tweets are art, books, celebrities, fashion, film, food, health, holidays, music, news, religion, sport, shopping, travel, tech, weather. These entities are chosen from analysis based on the recent study by Klout about the most frequently used topics in social media[29].

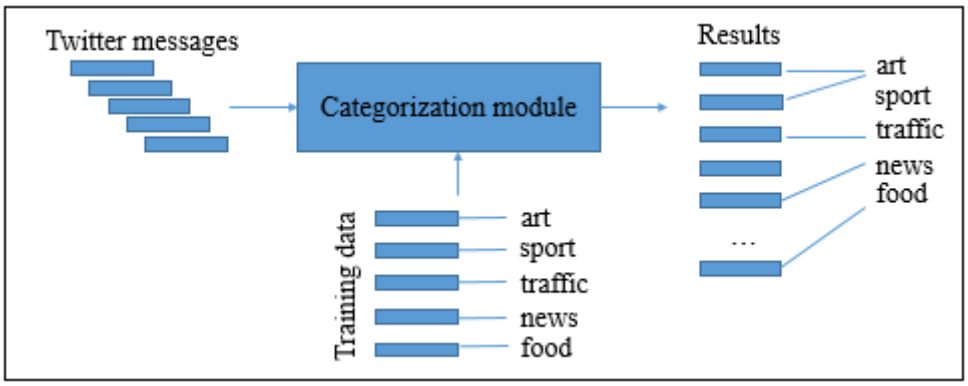


Figure 26. Functionality flow of event type categorization

Some tweets used for analysis are 10569; the data set was randomly split into a training size of 2115 records or 20% and testing size of 8454 or 80% of each category. Separation by each category is presented in Table 10.

Label Name	Tweets for Training	Tweets for Testing	Total Number of Tweets
Art	224	56	280
Music	128	32	160
Film	257	64	321
Books	20	5	25
Fashion	87	22	109

[29] http://www.marketingprofs.com/charts/2014/25346/the-most-popular-topics-on-facebook-and-twitter

Food	2322	580	2902
Health	164	40	204
Holiday	69	18	87
News	522	130	652
Other	3044	760	3802
Shopping	298	74	372
Sport	420	106	526
Tech	38	10	48
Traffic	596	150	746
Travel	220	56	276
Weather	45	12	57
Total	8454	2115	10569

Table 14 Tweets used for experimental analysis

Event type categorization was performed using three types of classifiers Naive Bayes (NB), Support Vector Machines (SVM) and Random Forest (RF). Table 11 shows the results for two feature selections, using max features of 100 and 150 trees for RF classifier. Confusion matrix was used to describe the performance of a classification model. We measure the quality of categorization by using precision, recall, F1-score and accuracy metrics.

Classifier	**BOWs**				**TF-IDF**			
	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy
NB	0.63	0.64	0.62	0.644	0.70	0.68	0.63	0.677
SVM	0.67	0.65	0.63	0.654	0.79	0.78	0.78	0.771
RF	0.69	0.68	0.66	**0.677**	0.78	0.77	0.77	**0.784**

Table 15 Evaluation metrics for classifying tweets into predefined categories

From Table 11 we can observe that RF overfits the other two classifiers in BOWs feature selection, while SVM is slightly better when is used TF-IDF. This is because TF-IDF is normalizing the word counts so that the favorite words are discounted, and probably our data set even after rigorous pre-processing there are still many words that do not provide useful information for the classification task. To the best of our knowledge, this is the first time that RF was tested with more than two classes. It showed the best results for classification of tweets into two categories (smokers, non-smokers) [13].

Class	F1-score	Precision	Recall	Support
Art	0.9	0.93	0.91	56
Music	1	1	1	5
Film	0.56	0.41	0.47	22
Books	0.98	0.72	0.83	64
Fashion	0.73	0.76	0.75	580
Food	0.79	0.78	0.78	40
Health	0.69	0.5	0.58	18
Holiday	0.93	0.78	0.85	32
News	0.89	0.72	0.8	130
Other	0.75	0.82	0.79	760
Shopping	0.77	0.81	0.79	74
Sport	0.86	0.62	0.72	106
Tech	0.71	0.5	0.59	10
Traffic	0.99	0.98	0.98	150
Travel	0.75	0.54	0.63	56
Weather	0.79	0.92	0.85	12
Average / Total	0.79	0.78	0.78	2115

Table 16: Accuracy by category using TF-IDF feature with Rf classifier

NB classifier can be utilized because of its simplicity, speed, and space efficiency. In our model classification parameters are independent. However, it has low accuracy rate. SVM cannot be used in our model because of its limited speed and significant memory requirements. RF can be utilized because of its space efficiency

When we visualize the output results like in Figure 29 we can observe the quality of the best event type categorization method, in this case, it is RF.

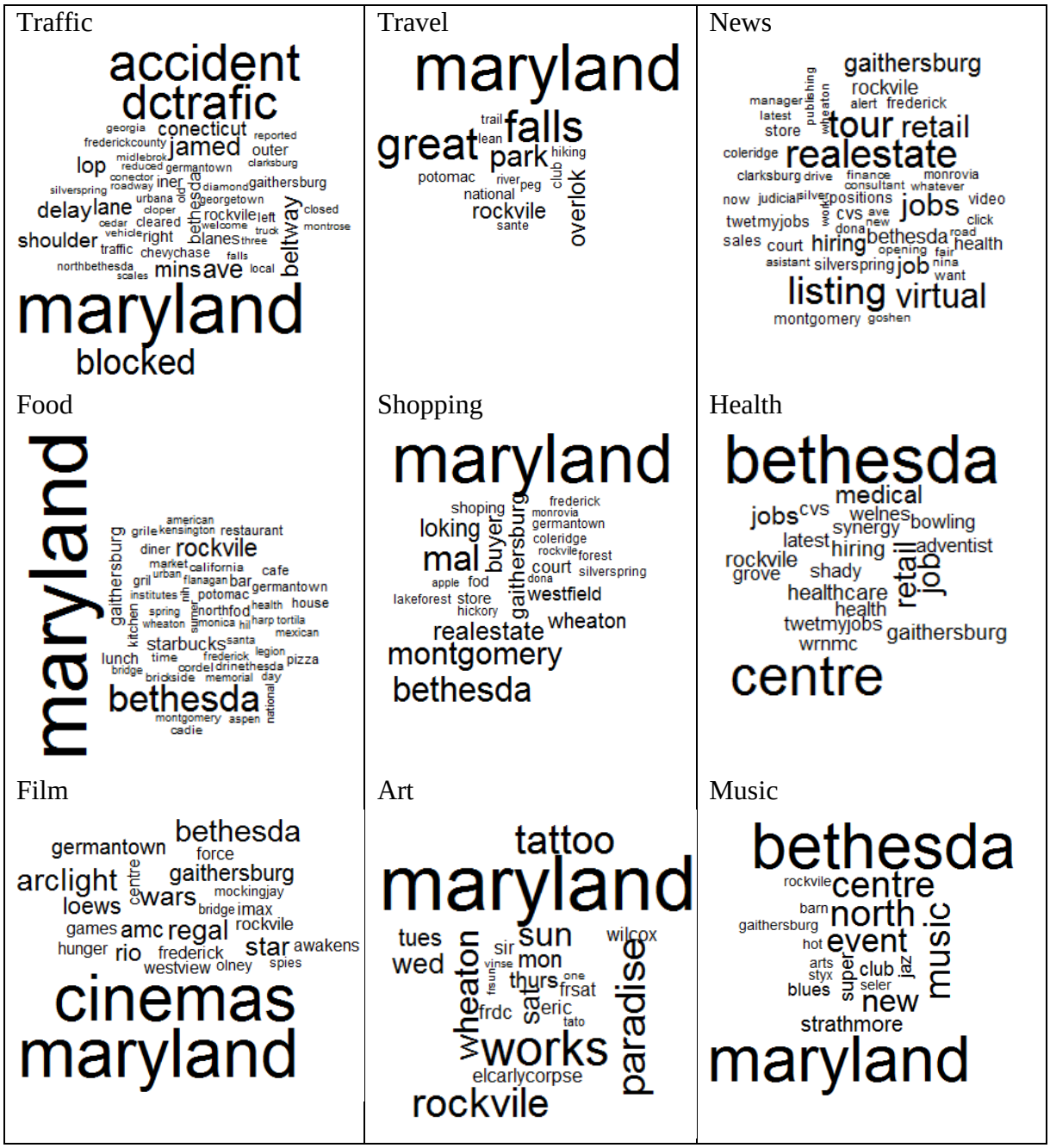


Figure 27: Word cloud illustration of tweets using RF with TF-IDF

As we can observe from Figure 29, in both of these word clouds, the most dominant words are highlighted. It is interesting to see that the most dominant word in each of the categories is ” Accident ” and “ Maryland ” which associates as category labels. It is also worth noting that, in both the clouds, there are dominant words that are not related to the category of the tweets like “ Old ” in Traffic cloud, or “ Peg ” in Travel cloud.

The next step in event network analysis is combining the results that we got with other algorithms to enrich the output results. Sentiment analysis (SA) was applied after the categorization step, and now we have à more detailed view in which sentiment context people are talking, positively or negatively about the trending topics. For instance, topic category *Weather* "Rain perfect weather stay read Zenzoris Returns," sentiment *positive.* For illustration Figure 30 show the line graph of topic categories with their sentiment measured level.

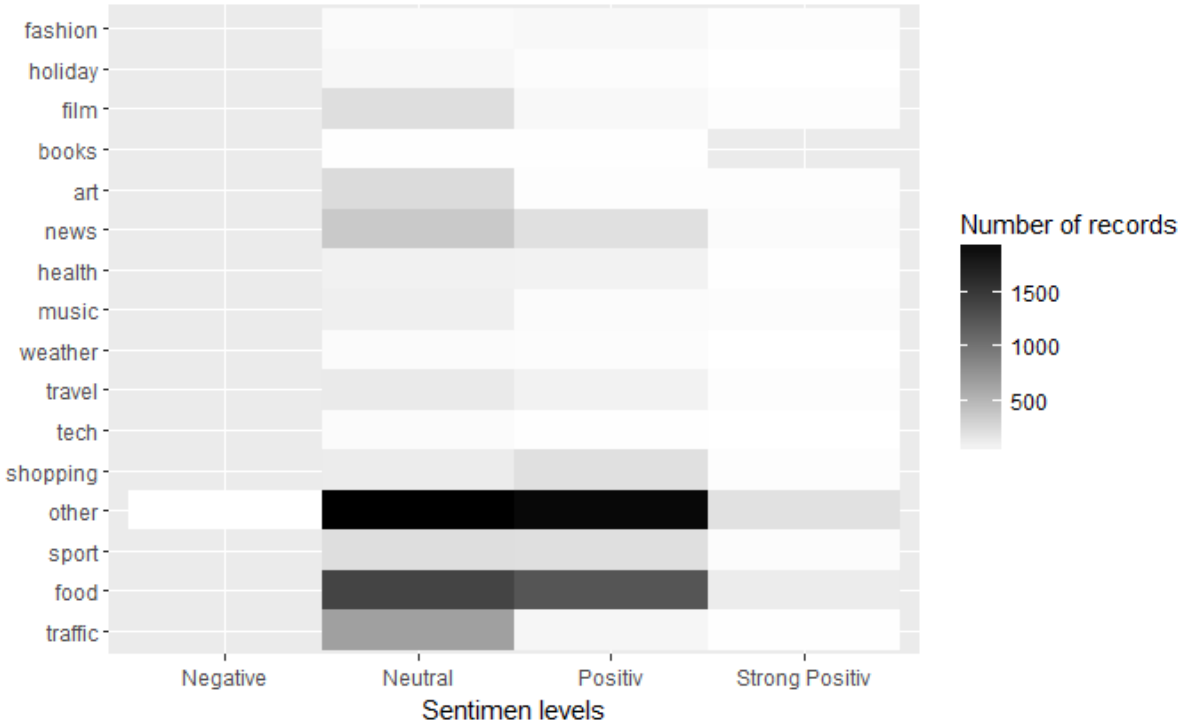


Figure 28: Sentiment level measure for all categories

Figure 30 shows that people were mostly talking about the weather but in a negative context, while when they speak of music was in high positive and few strong positive contexts. Moreover, the opinions of books and health are almost the same. Figure 5 shows that people were talking about fashion and religion in a positive context, while opinions for sport are mostly negative and neutral and are very close.

An SA, in this case, is very helpful, it adds a new value in measuring public opinion as well as know how to best harness the potential benefits of public services. For instance, if an event in central park is detected and the sentiment is negative or neutral, then the services related to navigation for runners or walkers will reroute the paths. Collaborative and personal recommendation services can be activated depending on their settings. The recommender systems, in this case, will adjust their algorithms to include sentiment analysis, and weight

differently services that receive a lot of negative feedback or fewer instances. However, the importance and sensitivity of the topic (emergency, earthquake) are highly relevant, in this case, the frequency of the tweets for the negative context can be lower. In the case of real-time processing, as topics and sentiments are changing, service recommendation is changing adequately, too.

Another technique that enriches the event network detection is a similarity. We used similarity metric to identify links between tweets semantically. We consider the distance between two topic clusters to be equal to the shortest distance from any element of one cluster to any element of the other cluster.

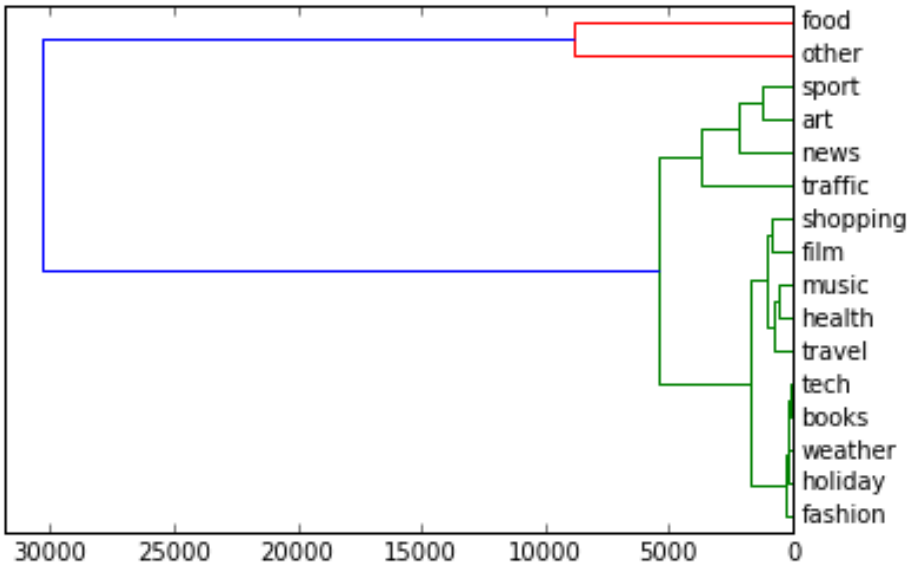


Figure 29: Similarity between categories

These are analysis for the whole period of one the year 2015; the following illustration 32 shows how the social sensor topic patterns changed over time during this year and present mood analysis for the particular subject (mood measurement for travel).

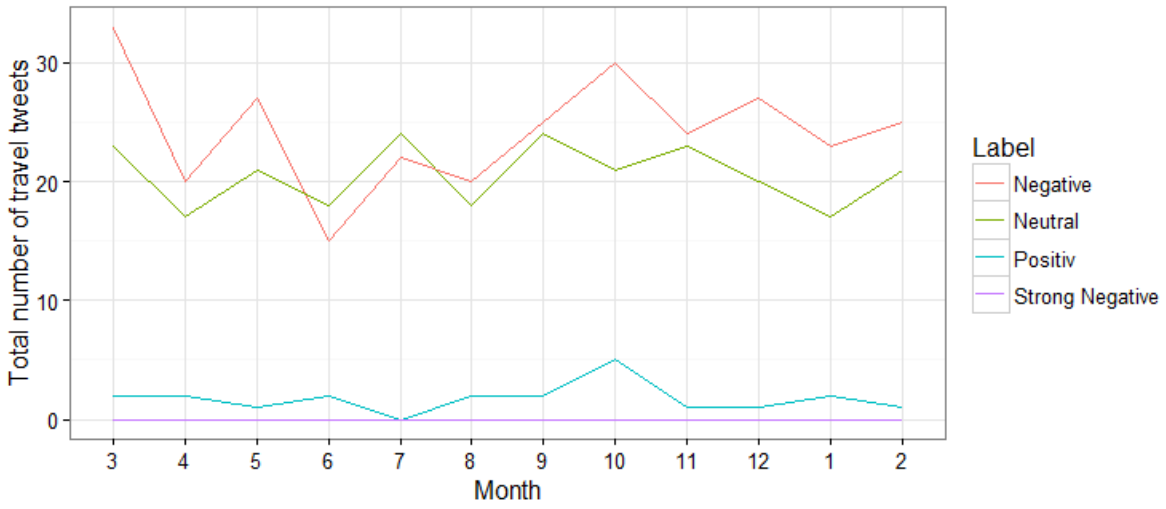


Figure 30. Sentiment dynamics per month for topic Travel

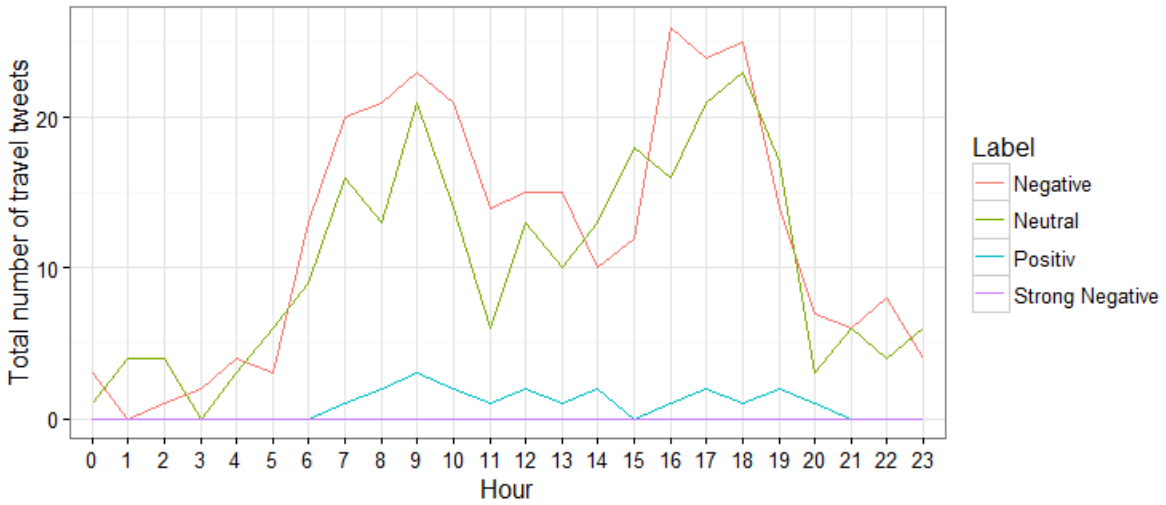


Figure 31. Sentiment dynamics per hour for topic Travel

The monthly observations show that there are more negative opinions about traffic during the fall and winter months (from October to March), while during the summer months June and July the negative view drops. However, the reason for that can be the number of tweets during that period. The hourly observations for topic traffic show that it is profoundly negative during the rush hours in the morning and evening, while the positive sentiment is shallow during the whole day. These observations require more investigations in terms which locations or part of the city the traffic issues are more affected. So it is an indicator that shows to decision makers for city stakeholders that some topics require more attention to considering. Solving this trade-off enables the behavioral hetereneity of the entities that compose the analyzed system.

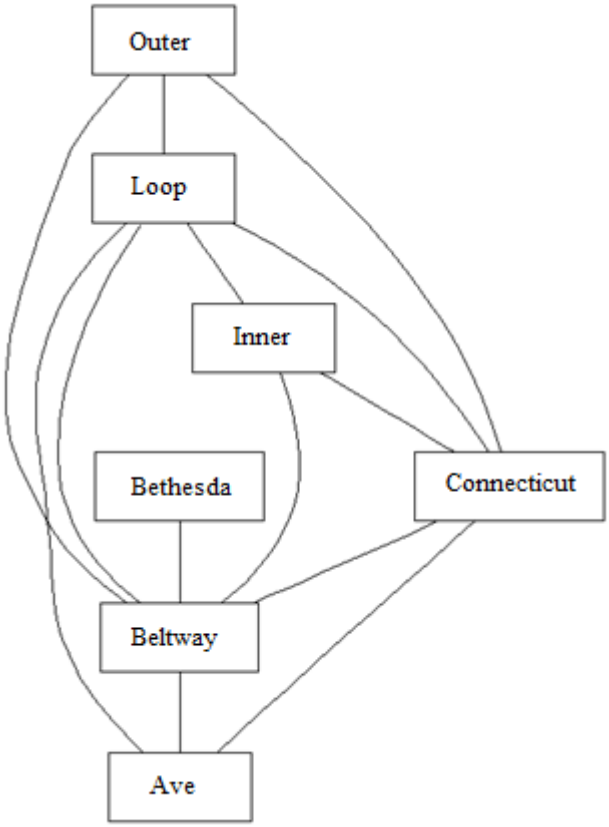


Figure 32. Network of words for topic Traffic

Moreover, when we investigate the correlation relationship for some of the phrase from the ne twork graph on figure 31, we can find that the closest one is: grill, bar, market, food, and matc hbox.

In conclusion

These experiments demonstrate a fully automated algorithm for extracting knowledge from social sensors, based on previously created a global approach to addressing this problem. Therefore was explored the general patterns of social media usage and presented a model for automatically categorizing the analytics for a broad range of predefined event type identifiers over one concrete geo-location, in this case, Germantown, MD, U.S.A. The experiments showed that the context-aware pre-processing algorithm (where every word in a tweet is necessary for analysis) used to process the tweets helps to categorize the tweets efficiently. It is shown that RF classifier combined with TF-IDF feature gives better results compared to SVM and NB classifiers. Moreover, sentiment analysis measures together with topic similarity provide additional information layer for determining public opinion and service recommendation based on social sensor data streams.

5.4 Evaluation of scalable semantic event model

As mentioned earlier events come in various types, format, and sources. This variety of data needs to be converted into a standard representation that is generic and does not need to be redefined for every new data source selected. Furthermore, the description needs to capture enough semantic and computational detail so that it can support a variety of situation recognition tasks.

We designed an event model (EM) that fits together these differences into the same framework, thus making data integration and processing easier. Moreover, the standard data structure that will allow different data streams to be used by multiple services. To build the combination between each kind of data and to raise the utilization of data. This model was described previously in Chapter 4.3.

In following, we will demonstrate the development of the EM, and illustrate how to use event detection, pattern identification, and prediction methods together. The case study for the validation experiments supplements the Pedestrian Safety Initiative in Montgomery County, Maryland, U.S.A. for improving the safety level for pedestrians by offering an algorithm to predict unsafe event areas per zipping code.

Experiment set up

For this experiment we used three different real-world data sets, dynamic data set is semi-structured traffic violation data sets, and dynamic structured data sets are weather and community events from Montgomery County, Maryland, U.S.A., more details for the data sets are presented in subchapter 5.1. Each of them characterizes the city from a pedestrian safety perspective; we will investigate his or her factors of safety influence and make a prediction for each zip code zone.

Results and discussion

As was described in Chapter 4.2 event model has three design phases: (i) event pre-processing and identification, (ii) event model formation, and (iii) event analytics. Also, this method follows the methodology of the new principle of proactive event-driven computing, and that is Detect – Predict – Decide -Act. The first is the event detect phase where events of interest are identified. Next is prediction period where the future number of events are predicted. The last step is decision-making stage where city authorities, representatives are making a decision what to do about the possible future situations. The next is acting, which represents the final actions product of the decision phase, can be ambulance allocation or police relocation depending on application domain of interest. For our case study since we are interested in

predicting a likely number of pedestrian events, the final action can be an allocation of more city resources, like the police. Let preview in details each of the design phases:

(i) Step one, before the data goes to event model, there is a need for pre-processing. We experiment with structured, semi-structured and non-structured data sets, or weather, traffic incidents and community events respectfully. Since traffic incidents are in text format and have few types of events, event detection method is applied to classify the events of interest into the event model. Be used previously developed algorithm (Chapter 5.2) for event type detection including pre-processing and feature selection phases. For this experiment was used a data set of 235,264 records, of which 2874 describe pedestrian incidents (but do not indicate incident severity). Evaluation results 0.98, 0.92, and 9.5 respectively Precision, Recall, F1-score metrics are high due to the homogeneity of the data. For weather and community events we did not apply event detection techniques because the data streams were already classified.

Since community events data streams were not in complete format, especially some of the attributes were missing or were not precise enough, we used uncertainty techniques to solve it. For instance, one approach was to find an alternative attribute that can fill in the missing one. An example of that is a longitude and latitude attribute, and in the cases where city, state, and zip fields are missing, for the goals of the use case, we need to have values for this fields. Another example is that location parameter has a broader diameter like the whole country, in this case, Montgomery County, Maryland, which means that event is affected by all cities inside the county. Uncertainty is complicated, and it is unlikely that one approach cannot handle all its complexities.

(ii) Step two, after event of interest were identified they are formatted into the event model structure. Depending on the importance of the event attributes event model entities are adapting. For instance, since weather sensor is a static device and its location parameters like longitude, latitude, city, and state are fixed this parameter is listed in Event Source entity. While in the case of pedestrian incidents, those events are dynamic, so location parameters are listed in Event entity. Figure 32 shows how different event streams fits into the event model.

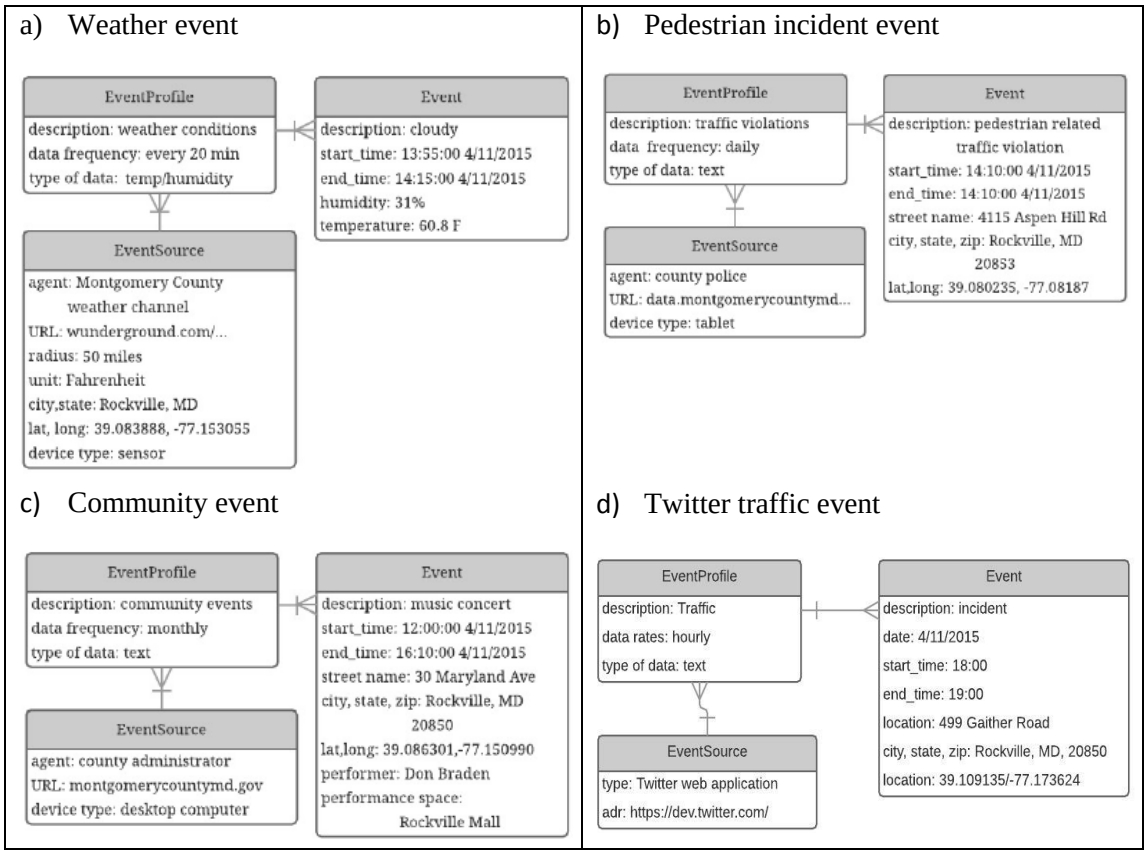


Figure 33. Event model metadata

a) cloudy weather event, b) pedestrian related event, c) community music event, d) Twitter traffic event

If some of the data sources are images or video cameras, the first step is event discovery and identify the semantics of that data stream. Recall that the event model is middleware (see Figure 23), and is transparent to the application service. The data would be stored in formats dictated by the model on a server that is either local or remote.

(iii) Step three, event analytics phase is oriented to event extraction, pattern detection and trends between events as well as to provide predictive assumptions. We extract the events of interest using event patterns like aggregation, selection, grouping based on application requirements. For instance, possible case studies are, extract the number of pedestrian events per zipping code and weekdays during one year, find the location where usually most of the pedestrian incidents happen, and what is the likelihood of pedestrian incidents during the rain. To extract these queries we used SQL language since we save the data to the database.
The output for the first case study, extract all events from pedestrian incidents for zipcodes 20910, 20906, 20876, and 20814 per day for the 2015 year. Since the model supports a variety of spatial attributes like street name and number, city, state, zip code, longitude, and latitude.

select Event.description from Event, EventProfile where Event.zip in (20910, 20906, 20876, 20814) and EventProfile.description = 'Pedestrian incidents' group by day

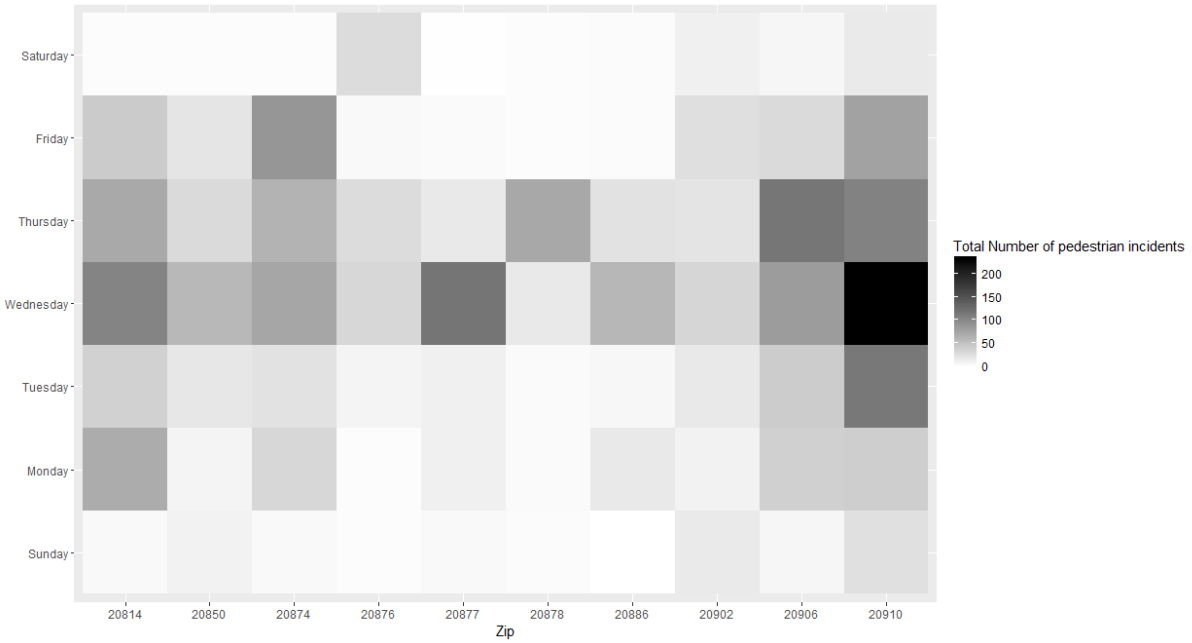


Figure 34. Total number of pedestrian incidents per zipping code and weekdays for the 2015 year

select Event.location
from Event, EventProfile
where EventProfile.description = " Pedestrian incidents "
group by Event.location

Results show that usually, events related to pedestrians happen around the shopping areas, downtown or restaurant area. For instance,*'264 Odendhal Ave, Gaithersburg, MD 20877, USA, 19899 Crystal Rock Dr, Germantown, MD 20874, and 8434 Colesville Rd, Silver Spring, MD 20910, USA*', are the location with the highest number of incidents. If we look deeper in the understanding of the context of these events, we can find that usually, they did not obey the signals for pedestrians like 'PEDESTRIAN FAIL TO OBEY UPRAISED HAND SIGNAL.' From entire 2874 events, only 63% of them are a pedestrian fault, while the rest of 37% is driver fault. This requires further investigation with a domain expert to find a reason why this is happening and what can be done to (or "intending to") decrease the results.

We used predictive modeling to (or "intending to") making assumptions for a future number of event related to pedestrians, and we investigate more the factors that potentially have an influence on pedestrian events. We model a case where we used time, location, weather and community events to detect the number of pedestrian's incidents. The output results predict which zones will be safer regarding some incidents compared to the other zones based on the past number of pedestrian incidents per zone.

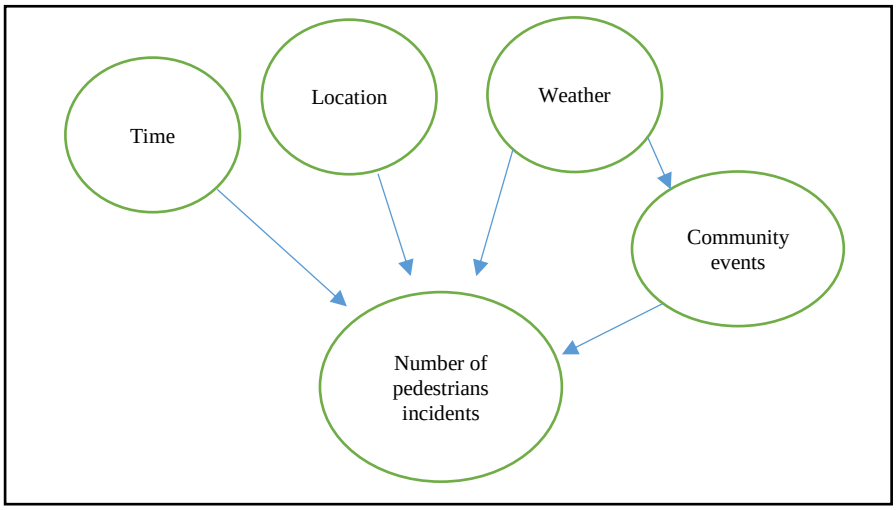


Figure 35. Graphical representation for modeling pedestrian incidents

We evaluate the model using Probabilistic graphical model (PGM) and Poisson regression (PR), we choose these two different types of statistical algorithms because they give different output, one is probabilistic, and the other gives the more precise number of predicted events. Accuracy is presented in Table 13, where PR give slightly better results compared to PGM.

	Probability of a (negative) pedestrian event in a location	
	Probabilistic Graphical Model	Poisson Regression
Average accuracy	0.864	0.881
Average standard deviation	0.213	0.096

Table 17: The accuracy of PGM and PR to predict hazardous locations

Also, the probability of sunny weather to the prediction of pedestrian events is 60%, while city events contribute with 80%.

If weather(rain) **then** there 20% chances of the pedestrian incident to happen

Predicted results and real-world data are visualized on maps presented in Figures 33 and 34.

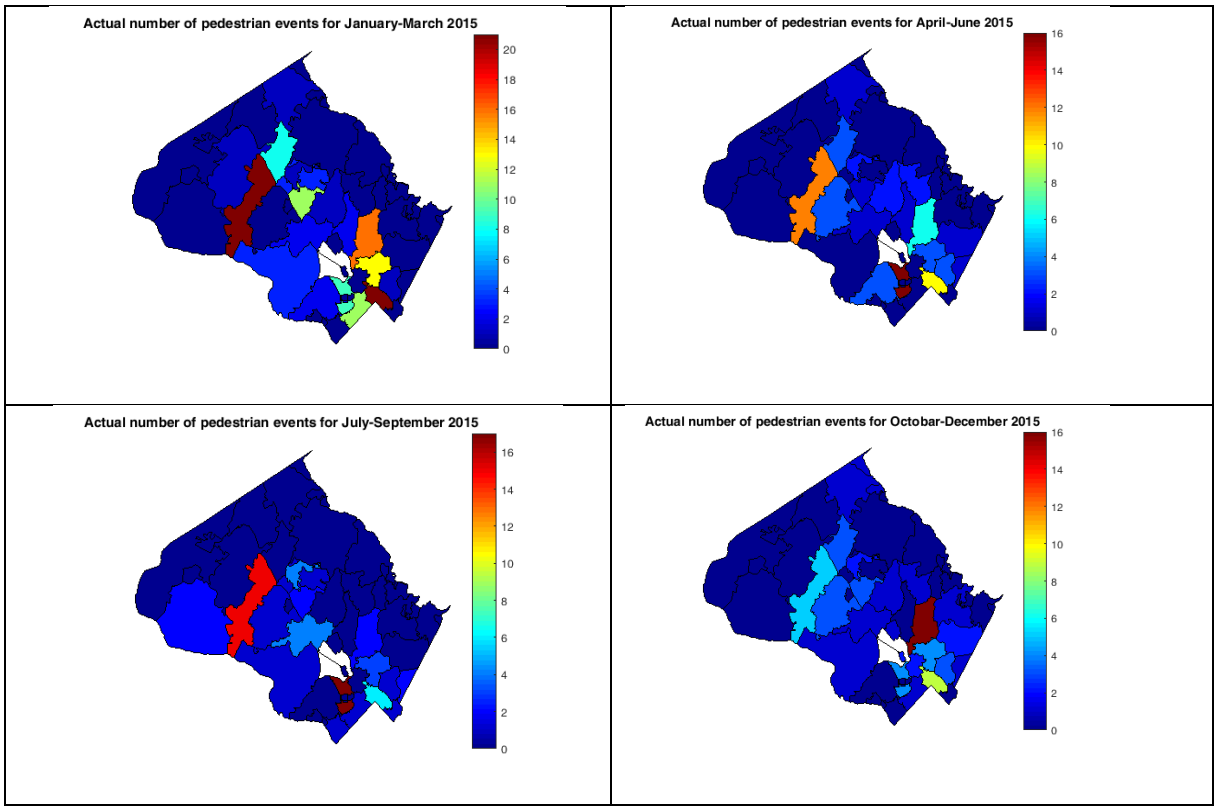
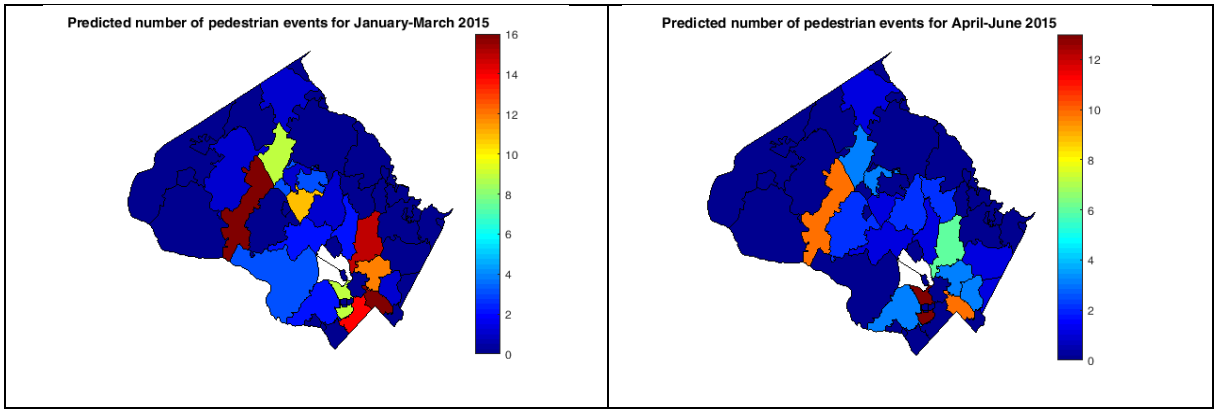


Figure 36: Actual number of pedestrian events by zipping code for 2015 in Montgomery County, Maryland, U.S.A.

Note, The diagram made in R (we have no data for the region in white)



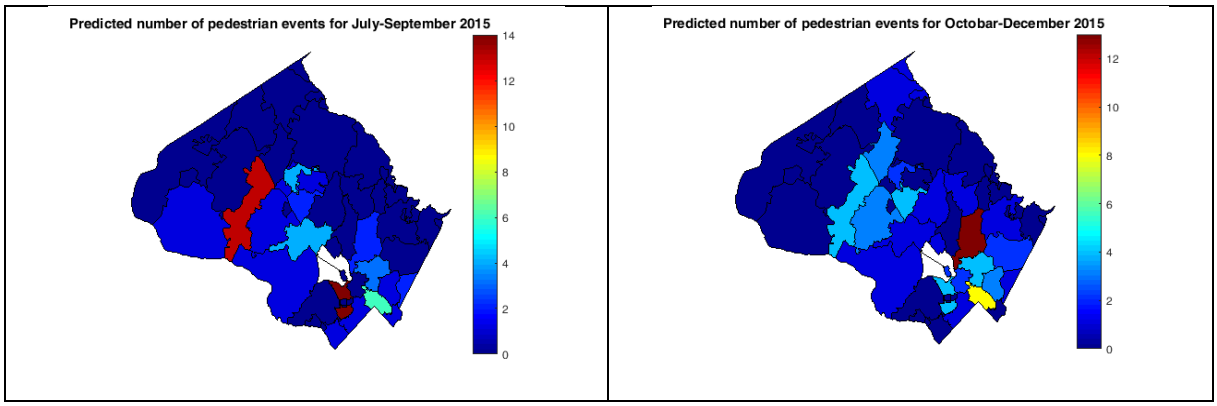


Figure 37: Events predicted using a Probabilistic Graphical Model by zipping code for 2015 in Montgomery County, Maryland, U.S.A.

Note, The diagram made in R (we have no data for the region in white)

We believe that the presented event model and other methods can be applied in hierarchical architecture like the one presented in [114]. Also, the event model can be applied to a horizontal organization based on open and shared platforms and resources like sensor data and access to actuators shared by several actors, that helps the user to control and supervise the city.

It can be implemented as a badge job or real-time service in a standalone application or can be integrated into the existing event platforms.

In conclusion

The presented event metadata model is unique among event models in being extensible as needed. We illustrate the model by showing it with different data types, and we validate it using real-world data in a case study for pedestrian safety. Broad adoption of our event metadata model has the potential to broaden the number and scope of smart city services. An event model with an ontology to control word descriptions will bridge smart cities services by reusing data streams across applications.

5.5 Evaluation of dynamic network model

We assume that each data source collects information independently, but for various factors, the event model sometimes cannot receive the expected data streams. We design a model that adapts dynamically on network data streams changes where each data stream is a separate node, considering that source to source data sharing can improve resource deployment. This model resilient on failures allows an analyst to capitalize on underlying shared trends between data streams to mitigate the effects of data loss and day-to-day data rate variance to detect multi-day trends better and ensure improved service estimation. We evaluate the results by demonstrating the case study in cross-country crime for Montgomery County, Maryland, U.S.A. where each city is considerate as a separate node, considering city-to-city data sharing can improve county-wide resource deployment.

Experiment set up

For this experiment, the data describes the number of police-reported incidents, organized by city, throughout the Montgomery County, Maryland, U.S.A. area between 01/01/2014 and 06/26/2016, constituting a spatiotemporal multivariate time series. The process of getting the data is explained at the beginning of this chapter, or in subchapter 5.1.

Results and discussion

Relationship Analysis: Identifying Potential Network Connections

i. Bivariate and Trivariate Granger causality test

After data normalization, each city crime rate data stream and the weather data stream were programmatically confirmed to be stationary. The Granger test was then applied to each pair of data streams to quantify the bi-directionally predictive causal relationships, as described above, with the lag parameter automatically selected by the AIC method. Similarly, the multivariate Granger test was performed for each triple of data sources. For both types of models, the weather was removed from the set of target variables. The results of the Granger test analysis for two-city models are shown in Table 14.

A Granger test p-value below or equal to the significance level of 0.05 is used to identify that the forecast data stream is a good predictor for the target data stream**Error! Reference s ource not found.**. The Granger test indicates that in 57% of two-city models and 37% of three-city models the forecast data stream provides statistically meaningful information about future values of the target data stream, and can, therefore, be used to improve prediction of the target data stream. Using the Granger test narrows the hypothesis space from 289 potential models (90 two-city models and 199 three-city models) to 120 models or 42% of the original hypothesis

space. Each indicated Granger-causal data stream pair can now be investigated for prediction accuracy using the VAR-based resilience. It was also found that for all eight cities, weather Granger-causes the daily crime rates either individually or with an additional supplemental data stream, confirming the results from.

The resilience performance for all models was computed and compared to the Granger test predictions to identify the efficacy of the Granger test. It was found that the Granger test accurately identifies a predictive causal relationship among 61% of the two-city models and 61% of the three-city models. However, among the top three models for each city, only two models were misclassified, and these were both the third best models for their cities. This shows an excellent ability for the Granger test to restrict significantly the hypothesis search space while still retaining the best performing models for each city. It was also confirmed that the significance level of 0.05 is optimal in detecting Granger-causation over the range of 0.03 to 0.08 with maximum performance at 0.05.

		Forecaster								
	Data Stream	Silver Spring	Rockville	Bethesda	Chevy Chase	Gaithersburg	Germantown	Montgomery Village	Potomac	Weather
Target	Silver Spring	----------	**0.00009**	**0.01685**	**0.03399**	0.17606	0.10106	**0.00925**	0.06527	**0.00796**
	Rockville	**0.02921**	----------	0.09925	0.28008	**0.00313**	**0.03449**	**0.00111**	0.05617	**0.01097**
	Bethesda	0.11235	**0.00658**	----------	**0.01854**	**0.00786**	**0.01437**	**0.00026**	0.21347	**0.02932**
	Chevy Chase	**0.01069**	0.67876	**0.01969**	----------	0.55178	0.85620	**0.04985**	0.05769	**0.03993**
	Gaithersburg	**0.01166**	**0.01822**	**0.00011**	**0.00072**	----------	0.09518	0.06589	0.10123	**0.00426**
	Germantown	**0.01208**	0.35502	0.39609	**0.04996**	0.31938	----------	0.05555	0.77694	**0.01047**
	Montgomery Village	**0.00944**	**0.00173**	**0.04038**	**0.04164**	**0.00491**	0.05677	----------	**0.02273**	**0.00316**
	Potomac	0.07615	0.170987	0.25884	0.24689	**0.01017**	0.06066	0.23479	----------	0.24719

Table 18: Granger causality relation index between top eight cities by the number of crime events

ii. Qualitative relationship identification (perspective)

Once a set of resilience models have been selected and analyzed for their performance (described in the next section) in the first iteration, the space of possible two-city resilience models for future iterations can be whittled down by identifying underlying city parameters that may predict the performance of those resilience models. For instance, if it is found that cities separated by vast distances tend to be weak predictors for each other, a threshold on city-to-city distance can be used to reduce the model hypothesis space. The city parameters investigated include city-to-city distance as well as the city demographics of population, education bachelor or higher, and average household income. MDS is used for qualitative analysis of underlying city parameters. First, a two-dimensional mapping is performed of resilience model performance, with each pair of cities described by the maximum prediction error (Figure 38a).

Here Chevy Chase, Montgomery Village, and Potomac were removed as their crime rates are so low as to be poor comparisons with the rest of the cities. This does not affect the MDS plot as the three cities fall near the origin, and the five other cities retain their relative position. This mapping is compared to the geospatial map of city-to-city distances (Figure 38b). Additionally, each city is described by a vector of the city-based demographics data. The demographics data is normalized by subtracting the mean and dividing by the standard deviation of each demographic parameter. An MDS two-dimensional mapping is performed using the Euclidean distance (Figure 38c).

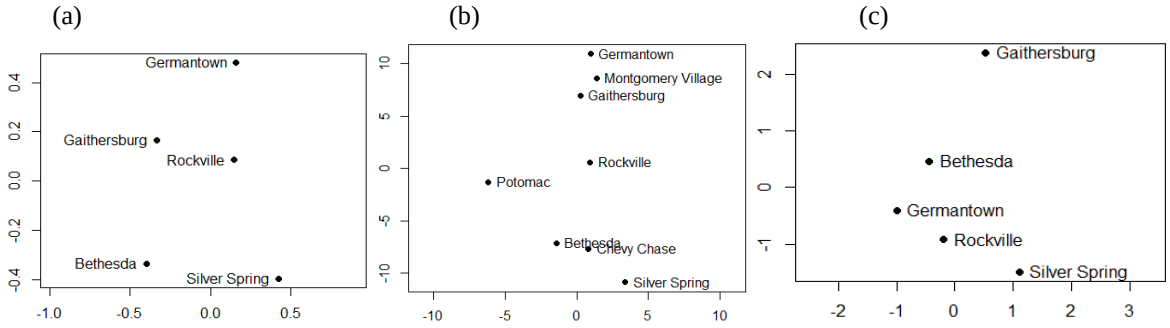


Figure 38 :: Graph representation of cities in Montgomery County, Maryland by three dimensions.

(a) mean square error from scenario two, (b) distance in miles between the cities and (c) demographics (population, education, and income).

The mapping of prediction performance is highly like the geospatial mapping, with the towns occurring in similar relative locations except for Gaithersburg. All city pairs also occur at the same relative cardinalities, e.g., in both mappings, Bethesda appears to the left and above Silver Spring. The similarity between mappings indicates that geospatial positioning may be a good predictor for resilience model performance and may also be a good choice of city parameter to reduce the hypothesis space of possible resilience patterns, with cities that are geospatially far apart less likely to have high performing resilience models. By restricting the hypothesis space for two-city VAR to only the two nearest neighbor cities among the five cities of interest, the best or the second-best models for each city is captured. Thus, a search space of $N^2 - N$ models can potentially be reduced to $2N$ models.

Investigation of the demographics mapping shows a lower agreement with the resilience performance mapping, indicating that two cities may be more likely to share crime rate trends if they are neighbors than if they share demographic trends[30]. However, these demographics results may be due to the demographics was chosen and the demographics normalization used, suggesting further investigation.

[30] "Everything is related to everything else, but near things are more related than distant things", First Law of Geography.

Resilience Model Evaluation

All possible resilience models were investigated for their prediction performance. The hypothesis space includes all possibilities of the three model types: 1) AR models, 2) two-city VAR models, and 3) three-city VAR models.

Table 15 shows the MSE prediction errors computed for the first two model types, over the full-time range, with forecast data streams listed as columns and target data streams listed as rows. AR models fall along the table diagonal with the rest describing two-city VAR models. Table 16 provides the percent improvement in prediction for the two-city VAR models over the AR models for each data stream. The best model is indicated with the color coding.

		Forecaster								
Target	MSE	Silver Spring	Rockville	Bethesda	Chevy Chase	Gaithersburg	Germantown	Montgomery Village	Potomac	Weather
	Silver Spring	1.11729	1.05356	1.13045	1.09662	1.09721	1.10938	1.09081	1.12080	1.08126
	Rockville	0.95685	0.97789	0.98851	0.96154	0.95159	0.94610	0.99172	0.98421	0.96799
	Bethesda	0.97911	0.97003	0.99561	0.99549	0.96035	1.01674	0.93352	1.00682	1.00749
	Chevy Chase	0.75906	0.75919	0.77145	0.76441	0.77892	0.76088	0.75464	0.75101	0.76816
	Gaithersburg	1.00187	0.94575	0.97302	0.98055	0.97684	0.96209	0.98702	0.96385	0.96160
	Germantown	1.01198	0.99755	1.01942	0.98335	1.00427	1.01820	1.02760	1.03902	1.001856
	Montgomery Village	0.92669	0.90070	0.93184	0.93170	0.89326	0.94129	0.92424	0.93403	0.92309
	Potomac	0.68546	0.67539	0.67157	0.65870	0.64944	0.71280	0.67835	0.66590	0.670914

Table 19: Validation metrics, mean squared error (MSE) for scenario one and two

		Forecaster								
Target	MSE	Silver Spring	Rockville	Bethesda	Chevy Chase	Gaithersburg	Germantown	Montgomery Village	Potomac	Weather
	Silver Spring	----------	5.70353	-1.17856	1.84972	1.79683	0.70770	2.36989	-0.31464	2.43592
	Rockville	2.15196	----------	-1.08537	1.67231	2.68973	3.25112	-1.41444	-0.64589	0.73619
	Bethesda	1.65704	2.56950	----------	0.01200	3.54149	-2.12269	6.23599	-1.12574	-1.00834
	Chevy Chase	0.69909	0.68260	-0.92130	----------	-1.89847	0.46142	1.27733	1.75175	-0.76476
	Gaithersburg	-2.56224	3.18280	0.39153	-0.37934	----------	1.51011	-1.04254	1.32985	1.42131
	Germantown	0.61066	2.02838	-0.11969	3.42264	1.36901	----------	-0.92275	-2.04451	1.60526
	Montgomery Village	-0.26557	2.54621	-0.82285	-0.80734	3.35167	-1.84473	----------	-1.05966	-0.22616
	Potomac	-2.93622	-1.42485	-0.85108	1.08158	2.47264	-7.04199	-1.86857	----------	-0.75244

Table 20: Percentage improvement of model two using model one as a base model

For the third model type, three-city VAR, all the approximately one thousand models were evaluated. For simplicity, the top three performing three-city VAR models for each city

is listed in Table 17 along with the models' MSE and their percent improvement over the AR

City	M_1	M_2	M_3
Silver Spring	Silver Spring + Rockville + Chevy Chase (1.038511; 7.050432%)	Silver Spring + Rockville + Montgo mery Village (1.044873; 6.481486%)	Silver Spring + Rockville + We ather (1.046729; 6.315370%)
Rockville	Rockville + Silver Spring + Germantown (0.9253039; 5.377611%)	Rockville + Germantown (0.94610; 3.25112%)	Rockville + Silver Spring + We ather (0.9482259; 3.03348%)
Bethesda	Bethesda + Rockville + Montgomery Village (0.9177663; 7.818525%)	Bethesda + Silver Spring + Montgomery Village (0.9289618; 6.69420757%)	Bethesda + Montgomery Villag e (0.93352; 6.23599%)
Chevy Chase	Chevy Chase + Silver Spring + Germantown (0.7378122; 3.478879%)	Chevy Chase + Bethesda + Silver Sp ring (0.7388459; 3.34429167%)	Chevy Chase + Potomac + Silv er Spring (0.7402985; 3.15426 276%)
Gaithersburg	Gaithersburg + Montgomery Village + Rockville (0.9369844; 4.080038%)	Gaithersburg + Rockville + Bethesda (0.9417804; 3.589083166%)	Gaithersburg + Rockville (0.94575; 3.18280%)
Germantown	Germantown + Chevy Chase (0.98335; 3.42264%)	Germantown + Rockville + Weather (0.9930925; 2.465970%)	Germantown + Rockville (0.99755; 2.02838%)
Montgomery Village	Montgomery Village + Rockville Gaithersburg (0.8883964; 3.877734%)	Montgomery Village + Gaithersburg (0.89326; 3.35167%)	Montgomery Village + Rockville (0.90070; 2.54621%)
Potomac	Potomac + Gaithersburg (0.64944; 2.47264%)	Potomac + Weather + Gaithersburg (0.6579727; 1.191014%)	Potomac + Chevy Chase (0.65870; 1.08158%)

model. It was found that three-city VAR models are among the top performing or second best-performing models for each city.

Y_1	Y_2	Y_3	MSE	Improvements %
Silver Spring	Rockville	Chevy Chase	1.038511	7.050432
Rockville	Silver Spring	Germantown	0.9253039	5.377611
Bethesda	Rockville	Montgomery Village	0.9177663	7.818525
Chevy Chase	Silver Spring	Germantown	0.7378122	3.478879
Gaithersburg	Rockville	Montgomery Village	0.9369844	4.080038
Germantown	Weather	Rockville	0.9930925	2.465970
Montgomery Village	Rockville	Gaithersburg	0.8883964	3.877734
Potomac	Weather	Gaithersburg	0.6579727	1.191014

Table 21: Validation metrics, mean squared error (MSE) for scenario three and percentage of improvements compared with the MSE of scenario one as a baseline

Table 18 provides a ranking of the top three models for each city across all three model types along with each model's percent performance improvement over AR. As can be seen, for all city data streams the use of additional data sources provides improved prediction and thus improved resilience in the case of data loss. For each city data stream, at least one other source can be used to improve prediction accuracy over simple AR with a maximum improvement of 7.8%, an average improvement of 4.7% for all cities, and an average improvement of 5.6% when excluding the cities with few crime events per day.

The top model for each city is chosen for implementation in the resilience network, see Figure 4. In dynamic operation (discussed in the next section), if an event results in the inability to use the top model, that model is then replaced by the next best model, and so forth.

Table 22: The best three results from all scenarios for each data stream, and the percentage of improvement compared with scenario one as a baseline

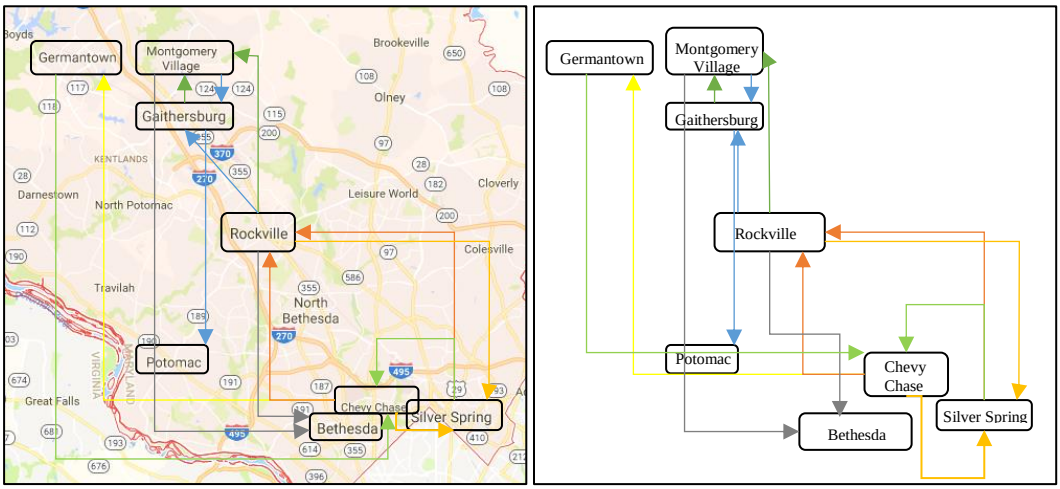


Figure 39. Network graph representing data sharing directionality between the cities

Figure 40. Real data (M_0) and predicted values for Silver Spring using the best models (M_1, M_2, M_3) presented in Table 4

Resilience Network Dynamics

Resilience network dynamics allow the model to self-adapt to changes in the data streams, so that it always provides optimal performance. Dynamics is achieved by iterating network determination at user-determined intervals or from a user provided a trigger signal. Figure 40 shows a dynamic implementation for Silver Spring with only models of type one and two investigated. For this implementation, at each date, the network is provided data from the previous four weeks, ensuring that trends learned by the models are local in time. The model which provides the best performance is chosen dynamically for network implementation. Here it can be seen that for the first four weeks, use of Montgomery Village data provides the best prediction performance. The network graph for Silver Spring is diagramed above these dates, with a directed edge from Montgomery Village to Silver Spring. On day 28, the optimal resilience model changes, with Montgomery Village being replaced with the weather data stream. On day 83, the network updates again to depend on the Rockville data stream. As discussed above, in implementing such a system decrease the delay between model analysis and model selection may be necessary to improve system stability. Selecting the best model over a user-specified period will reduce the likelihood of rapidly alternating between models due to small variations in data.

Figure 41: Dynamic Network model for Silver Spring, using model two

Legend : Silver Spring (SS), Bethesda (B), Chevy Chase (CC), Gaithersburg (Ga), Germantown (Ge), Montgomery Village (MV), Potomac (P), Rockville (R), Weather (W).

In conclusion

We present a dynamic network model for improving smart city resilience to data loss. The system utilizes the Granger causality test to identify statistically significant shared temporal trends across multivariate data streams and utilizes VAR to capitalize on those trends to ensure improved data prediction in the case of data loss. Each data stream is provided a ranking of potential resilience models with the top performing model selected for implementation. If the top model can no longer be executed, the next best model is selected. Iterative evaluation of the the model provides a dynamic, self-adaptability to changes in data quality, loss of data streams, and the addition of new information flows.

The network model is demonstrated on City-based daily crime rates reported in eight cities across Montgomery County, Maryland, U.S.A. as well as a daily weather data stream. The optimal resilience network is identified and successfully demonstrated. It is shown that utilizing shared temporal trends between cities provides improved crime rate prediction and resilience to data loss, compared to the use of city-based AR. Additionally, the weather is shown to be a top choice for a supporting data stream by both the Granger causality test and VAR performance. This reinforces the finding that weather is a good predictor of crime rates. It was also qualitatively found that small city-to-city distances are a good indicator that temporal trends between city pairs will provide useful utility in VAR models.

5.6 Conclusion

This chapter has focused on evaluating the research challenges mentioned in Introduction chapter: (i) extract and validate knowledge from text data stream to improve context understanding in city-based services, (ii) examine, design and develop uniform format for complex data streams, (iii) and examine, design, develop a robust mechanism that makes service decisions more efficient. Research questions (i) and (ii) are validated in Chapters 5.2 and 5.3, while research question (iii) was validated in Chapter 5.4.

At first, with the help of Open Data, we set as a goal to provide clear, reliable and easy open event related information from multiple sources to offer a broader view of the activities in a specific geographical area of interest. All experiments contain details of the experiments like dataset description, processing steps, as well as experiment environment and used software packages explained in 5.1, which makes it reproducible.

The next chapter presents the conclusion of the proposed work, it discusses the open questions and gives criticism for the proposed solutions, in the end, it presents possible future directions, challenges, and applicability.

Chapter 6

Conclusion and future work

"Science never solves a problem without creating ten more."
— George Bernard Shaw

Living in a new era of smart and connected communities various devices are continually collecting information with the goal of helping communities in the environment by collecting data about air pollution, infrastructure by analyzing traffic, and socially by sharing this information. Understanding real-world event interactions and their dynamics from observed data is a challenging problem. Among the challenges are *detection*, *heterogeneity*, *incompleteness*, and *adaptation* of real-world events and their manifestations in observational data. The objective is to design, create, and evaluate a framework that will provide the tools that can be used by cities to justify the allocation of city and government resources that will make the city more productive, livable, equitable, safer. Citizens are rewarded in the sense of improved service delivery, by empowerment through increased information about their lifestyles, vielding greater productivity.

Our work has drawn from diverse areas such as data mining, natural language processing, algorithms, geospatial analysis, visualization, network science, predictive models, and statistical learning. Our approach is inspired by compact, common sense approaches. We summarize the main contributions of this dissertation below, together with limitations of the solutions presented and future perspectives.

6.1. Conclusion

Integrating multi-modal data streams from diverse domains can have different qualities, modalities, importance, and trust, which need to be identified and associated with a set of criteria that represent the application service. Given that cities are dynamic and evolving ecosystems, there is also need to continuously link, interpret and share dynamic knowledge across city stakeholders and citizens to make use of information before it is out of date.

Data analytics is a bottleneck within the proposed framework. We analyze spatiotemporal datasets collected from different data sources, times, formats, semantics, and context. We design a framework and develop solutions that solve the following challenges:

- Extracting knowledge from complex data format like text. We design and develop an algorithm for automatic context-aware preprocessing that takes into consideration the meanings of symbols and signs. We also develop functionality for event type detection and identify similarities between them enriched with sentiment levels. We graphically visualize the results so that the city representatives can understand them easily.
- Structuring incomplete data: Transforming unstructured and semistructured data to a unified format for later analytics. Typically text data has lots of acronyms, abbreviations, dates and location values; we develop an algorithm that will automatically detect the events and event attributes. While also handling uncertainty and incompleteness at the attribute level.
- Generic data event model: Making the spatiotemporal data capable of use in knowledge discovery. However, different data have different formats, context, semantics, and complexity. We develop functionality that provides a unified understanding of data semantics and extract new knowledge and the construction of a multi-dimensional spatiotemporal data model describing sensor information, including location attribution, observation object, time and status with a flexible structure.
- Knowledge discovery to understand the nature (e.g., correlations, context, and meaning) of data. We implement functionality for identifying relationships between events and identify causative correlations among events.
- Prediction analytics as an important technology in supporting proactive complex event processing. We add different types of prediction models that present another perspective for future event predictions.
- Dynamic network adaptation increase resilience capabilities of the proposed framework for event analysis and prediction. Because data loss may happen for various reasons, we created the dynamic network model that is fault-tolerant to environmental changes and adaptive as needed to support service continuity.
- We also used efficient multidimensional visualization of spatio-temporal data.
- We recognize the challenges for Smart City use cases/applications, and we identify appropriate real-world case studies for experimentation.
- We comply with NIST ongoing standard for Big Data Use Cases.

This results of this dissertation are of interest to both researchers and practitioners. As a theoretical contribution, we contribute to filling the gap in the existing literature that deficiencies techniques to deal with unified structuring, and resilient event processing. As for the contribution to decision-making practice, we provide a tool that supports the decision makers for city stakeholders in defining strategic and multisectoral action plans for city resource management. This tool can efficiently process the data and correspondingly visualize the results. Also, it can work standalone or can be implemented over existing event platforms.

We conducted different experiments and validation actions using real-world data in the context of the smart city. We need to make improvements to the proposed solutions by investigating more in the domains of intelligent event processing, handling uncertainty and incomplete events and smart adaptation to the environment. Details are presented in the following section 6.2.

6.2. Open questions and future improvements

The work discussed in this dissertation opens numerous avenues for further investigations. In this chapter, we outline several promising directions.

Rather than an event detection algorithm based on a predefined group of event types of interest, which is a costly and time-consuming process especially when we have a high-velocity rate, we want to enrich our algorithms with topic modeling techniques which provide a more appropriate solution to the detection algorithm adaptable to change. We want to create a dynamic classification engine which classifies the data into clusters and sub-clusters recursively and assigns documents to predefined thematic groups that share some common traits. Also, we want to relate the detected events with real-world events on the news and provide more realistic analysis. These events can be utilized for better situational awareness by city authorities and people, by also taking advantage of investigating more features, such as network features and ranking them.

Trust systems are usually based on qualitative and/or qualitative user's experiences, interacting with the services and resources. Identifying relationships among Twitter user and topics of interest can be used for creating possible trust relations. While social media is an excellent source of real-world events as demonstrated in this dissertation, there are several challenges in utilizing them. Some of the problems include data quality, trustworthiness, and redundant and biased event reports. It would be interesting to explore data quality issues while also making use of people's observations. One possible research direction is to study the trustworthiness issues associated with various real-world events. Trustworthiness of the reported events is crucial for decision making in crisis or the presence of an adversary. Some events may get reported more often compared to other activities resulting in the propagation of "popular" events. Understanding such biases would provide ways to calibrate how we synthesize information from reported events. Data privacy algorithms (such as K-anonymization, Randomized Response, and Differential Privacy) and data/knowledge access authorization [88] [90] are necessary for data owners.

In future work, we want to test the event model with various velocity levels, and scalability with the data streams with many attributes, up to 100 or more. Also, we want to test for city services that follow multidimensional classification standard for services which are becoming more and more diverse and complicated, as per the kind of services, be the public or private, global or local, permanent or instantaneous [65].

Integration of remote sensing and sensor webs within the event model can expedite this urban reality. It is impractical to obtain digital measurements for every point across an entire city community: the available information will always be incomplete; a decision maker scan is better informed through such technology integration even if loosely coupled. For this reason, we want to introduce more efficient methods for handling incompleteness and uncertainties, like fuzzy rules and these rules. Additionally, we want to implement functionality for composite event patterns by using inductive logic programming and weight learning [14].

Available information also carries uncertainties in different levels. Future research will, therefore, be devoted to modeling uncertainties that affect the estimation of optimization model parameters. We will address the integration of the decision-making model of other urban-based subsystems. Also, we want to add the functionality for detecting compromised data streams.

Practice shows that there is a need for standardized event data format, for instance, the same data sets such as traffic incidents do not have the same schema in the open data repositories in New York City and Chicago. Also, developing more testbeds like the one prepared by DARPA (Defense Advanced Research Projects Agency) and TREC (Text Retrieval Conference) that produces the best-automated coding for event data, this would allow scholars to test tools already developed.

In terms of adaptive event processing, we want to investigate time difference coefficients of VAR and mixed frequency data. We also want to consider more different data types categories (e.g., various crime types, traffic types and so forth), and to explore data stream trust weight to increase the impact of more trusted data sources and reduce those of less trusted sources.

While demonstrated with an inter-city network, the system can be implemented on other data stream networks such as distributed local clouds, where each local cloud (or cloudlet) is considerate as a separate network node. To implement such an exchange, we are considering techniques for efficiently distributing and replicating data among a network of data sources, considering scalability and traffic spikes. As communications and data exchange are closely related, this will allow us to consider information related to location and network topologies to improve resilient data exchange between devices. We plan to explore such problems as the placement of broker replicas, aggregating sensor data at local peers to alleviate congestion, and caching information on nearby peers and local servers both to improve the chances of recovering that data in the event disaster and to enable local event-detection on the nodes holding these caches.

The other approach to enable resilient deployment and execution of Smart City applications, is with network architecture, instead of the traditional centralized approach to use a distributed execution environment for logic and analytics. Such an architecture can potentially mitigate the effects of component failures on an overall system by providing redundant channels to accomplish business logic. We plan to investigate a mobile intelligent agent-based approach in which application logic is modularized and distributed across multiple devices and data

centers. Considering those nodes as mobile agents that consume some data and forward the answer to the next node in line : we can realize a framework for developing application workflows that naturally maps to a dynamically adaptive distributed execution environment. These agents can then migrate between devices for scalability, replicate for reliability, or stay within system boundaries for policy compliance. This platform could leverage the resilient data exchange by relocating nodes, removing or integrating data streams together to utilize locality and improve the efficiency of the system.

In addition to the context of the Smart City application domain, we want to extend the work and contribute more to solving the problem of data sharing and knowledge transfer. In future work, we want to experiment with transfer learning, and use the knowledge from the data sources with more data points to the smaller cities with not enough data, since there are many cases like that in Maryland, U.S.A. Therefore this is a potential for small, more rural areas to become ” smarter ” and promising for the future of the connected world. Collaborative sensing [37] improves service performances by providing better awareness and control of the dynamic environment and correlated data streams, with integrating and analyzing spatial-temporal data. Collaborative knowledge discovery algorithms to enable collaboration between static and dynamic sensors, as well as between crowd and city sensors. However, due to the problem of data integration, the study of collaborative knowledge discovery is still limited.

6.3. Future perspective and challenges

The ideas and algorithms presented in this dissertation raise challenges in different domains such as event management, cloud computing, power grid management, healthcare, system health monitoring, and the smart city. However, in this dissertation, we choose to focus on the smart city, due to its contemporary interest and importance in many countries around the world.

We identify the following research areas that can be investigated in the future; they are organized into subsections for clarity.

Summarization of things

Discovering similarities and sentiment links of events can be used in the field of summarization of things to support advanced summarization, especially in summarization through interactions in cyber-physical space and network of things.

The solution we proposed is tested only on Twitter data streams, but it should be flexible enough to be applied to any text data like participatory sensing for Facebook, or Myspace, or Foursquare or another form of text data sources like articles and documents. We want to do additional testing in that direction. It was shown that high-level topics could be useful for a variety of upstream tasks such as summarization. In this direction, we believe that the output of this work can be used in event management application such as creating large-scale festivals,

conferences, concerts and so forth, by creating organization calendars and grouping similar events together.

For decision-making visualization can be utilized, or some of the pairwise comparison functions can be applied.

Self- capabilities*

Self-aware technologies/service frameworks were studied by Nakamura, M., and Du Bousquet, L. [93], they proposed two kinds of the model that provide a standard view of smart city services, execution, and life-cycle models. They identified the self-* capabilities[31] that will make city services highly manageable by limited human error.

The resilient processing algorithm we develop can be used in situations that require the ability to reduce the magnitude and duration of disruptive events. Relationship identification between countries [149] improve when there are significant upticks in event counts. The effectiveness of resilient characteristics depends on its capacity to anticipate, absorb, adapt to, and recover rapidly from a potentially disruptive event. Also in non-deterministic scenarios where methods for discovering useful and correlative information from data and utilizing them for a better lifestyle, in real-time mode, are the absolute requirements.

In future, we want to design and develop a set of adaptive learning methods that uncover complex and hidden patterns in extensive time series data.

Cloud and network performances

Identifying the volume of traffic on Twitter and identifying the topics users are interested can help in better allocating cloud resources and network demand. Using social networks as a source of data we can detect the users' events of interest and provide better service recommendations. For instance, if we detect that during the tennis championship the request for services if higher we allocate more resource on the cloud and make an adjustment on the network demand.

Event model and dynamic relationship identification can be applied to distributed cloud architectures. This functionality can be structured in event processing networks with intelligent agents responsible for each function adding modularity to the system.

The models presented can be used in predicting performance over time, spatiotemporal network traffic dynamics and interactions of networks, as well as detecting persistent and transient performance "anomalies." Identifying major factors that influence the overall network performance across the network and over time is important for on-time instance of system performance degradation. Characterizing and analyzing network performance captures crucial static and dynamic inter-dependencies and models their common effects on network

[31] e.g. self-adaptation, self-organization, self-optimization, self-configuration, self-protection, self-healing, self-description, self-discovery, and self-energy-supplying

performance and robustness. Detect persistent and transient performance “anomalies” helps in problem diagnosis by guiding detailed analysis with additional (e.g., low-level) data sources.

For instance, detailed analysis of factors such as dynamics over time, spatial and temporal correlation, describing the behavior of the event can be used in monitoring overall network performance and give a better forensic view.

Security

The event detection techniques can be used in understanding network log files. Understanding these log files provides the potential impact of events on the network like malicious attacks; detection processes are maintained to ensure timely and adequate awareness of unusual events.

The use of predictive analytics can assist in providing homeland security stakeholders with information to better prevent, prepare for, and recover from an all-hazards event. Integrating or partnering with the homeland security community can help develop processes and procedures to use predictive analytics to safeguard the safety-related threats better. Once this information is collected and time-stamped, homeland security agencies can actively monitor dangerous areas through the analysis of semantic patterning, to potentially mitigate an unwanted situation. Townsend A.M. [144] lists the cloud, low-cost broadband, open-data, and open-source technologies as the prominent enablers for smart cities.

Appendix

A. Selected Event definitions

a. The Oxford English Dictionary defines an *event* as "Something that happens or is thought of as happening."

b. An *event* is an occurrence within a system or domain; it is something that occurred or is contemplated as having happened in that domain. The word *event* meaning refers to an actual occurrence (the *something that has happened*) in the real world or some other system [47].

c. An *event* is an object that is a record of activity in a system. The event signifies the activity. An event may be related to other events. An event has three aspects: form, significance, and relativity [91].

d. Primitive event: A primitive *event* is a data tuple with a unique id, a list of the attribute, and a timestamp, denoted as E = E (id, {a1, a2, …, an}, t}. id uniquely identifies the event and the source of the event stream. The primitive event is atomic which occurs at a specific time point t.

   Complex event: A complex *event*, denoted as E = E (id, p, tb, ts), is composed of the primitive event, where p is the pattern function or expression to describe how the complex event is composed. Tb and te are the starting and ending times of the event, satisfying tb < te.

   Event Stream: An *event* stream is an ordered sequence of event occurrences in a timeline, denoted as S = S (e1, e2, …, en), in which ei is the instance of the primitive or complex event. [95]

e. Representation of *events* allows connecting facts into a coherent representation of history. Linking of items, places and time thought events. Split up the evolution in discrete events in time and space. [44]

f. An *event* is a record of activity in a system and may be related to other events. It has the following aspects: (form) – These are the formal attributes of an event, such as timestamp, place or originator; (significance) – It is the activity, which signifies the event; (relativity) – This describes the relationship to with other events. An event can be related to other events by time, causality, and aggregation. It has the same relations as the signified activity of the event [156].

B. NIST Big Data Requirements Use Case

Use Case Title	Urban context-aware event management for Smart Cities – Public safety	
Vertical (area)	Complex networks; Smart City	
Author/Company/Email	Olivera Kotevska, Ahmed Lbath, Abdella Battou	
Actors/Stakeholders and their roles and responsibilities	Spatial-Temporal Analysts Complex Social Systems Analysis Decision Makers Policy Makers	
Goals	To use advanced methods to analyze complex data streams from socio-technical networks to improve the quality of urban applications. - Detect events from various network streams - Ability of intelligent data integration and structuring in the common format for diverse data streams - Relationship analysis between entities in the network - Reasoning from varied and complex data streams - Trends in sentiment for text data streams - Understanding how communication spreads over networks - Support for visualization	
Use Case Description	The real-world events are now being observed by multiple networked streams, where each is complementing the other with his or her characteristics, features, and perspectives. Many of these networked data streams are becoming digitalized, and some are available in public (open data initiative) and available for sense-making. The networked data streams provide an opportunity for their link identification, similarity, and time dynamics to recognize the evolving patterns in the inter-intra-city/community. The delivered information can help to understand better how cities/communities work (some situations, behavior or influence) and detect events and patterns that can be remedied a broad range of issues affecting the everyday lives of citizens and efficiency of cities. Providing the tools that can make this process easy and accessible to the city/community representatives will potentially impact traffic, event management, disaster management systems, health monitoring systems, air quality, and city/community planning.	
Current Solutions	**Computer(System)**	Fixed and deployed computing clusters ranging from 10s of nodes to 100s of nodes.
	Storage	Traditional servers
	Networking	Gigabit wired connections, Wireless including WiFi (802.11), Cellular (3g/4g), or Radio Relay.
	Software	Currently, baseline leverages: 1. NLP (several variants) 2. R/Python/Java 3. Spark/Kafka 4. Custom applications and visualization tools
Big Data Characteristics	**Data Source (distributed/centralized)**	Police reports for various city situations. Web scrapped data, wireless data, e-transaction data, individual contributors via web pages. Social

		media data and positioning data from different sources. Distributed IoT sensors (Physical devices that contain electronics, sensors, actuators and software, and that can collect and exchange data about and in some cases, interact with the physical environment.)
	Volume (size)	Depending on the sensor type and data type, some sensors can produce over a gigabyte of data in the space of hours. Other data is as small as infrequent sensor activations or text messages.
	Velocity (e.g., real time)	Depends on the use case, can be from hundreds to thousands of new information records per day. Some data streams are in real time (social media) other are less real time more daily. Data should be analyzed periodically.
	Variety (multiple datasets, mashup)	Everything from text files, raw media, imagery, electronic data, human-generated data all in various formats. Heterogeneous Datasets are fused together for analytical use.
	Variability (rate of change)	Continuous data streams are coming from each source. Sensor interface formats tend to be stable, while the human-based data may be in any format. Much of the data is unstructured. There is no critical variation of data producing speed or runtime characteristics variations.
Big Data Science (collection, curation, analysis, action)	**Veracity (Robustness Issues)**	Identification and pre-selection of appropriate data, uncertain and noisy data are possible. The semantic integrity of conceptual metadata concerning what exactly is measured and the resulting limits of inference remain a challenge. Data must have high veracity and systems must be very robust.
	Visualization	Displaying in a meaningful way complex data sets using tables, clustering, geospatial maps, time-based network graph model, and visualization techniques.
	Data Quality	Data Quality for sensor-generated data is known. Unstructured data quality varies and cannot be controlled.
	Data Types	Semi-structured datasets like numeric data (various sensors) Unstructured datasets like text (e.g., social networks, police reports, digital documents), multimedia (pictures, digital signal data);
	Data Analytics	- Pattern recognition of all kind (e.g., event behavior automatic analysis, cultural patterns). - Classification: event type, classification, using multivariate time series to generate network, content, geographical features and so forth.

		- Clustering: per topic, similarity, spatial-temporal, and additional features. - Text Analytics (sentiment, entity similarity) - Link Analysis: using similarity and statistical techniques - Online learning: real-time information analysis. - Multiview learning: data fusion feature learning - Anomaly detection: unexpected event behavior - Visualizations based on patterns, spatial-temporal changes.
Big Data Specific Challenges (Gaps)	Data that currently exists must be accessible through a semantically integrated data space. Some data are unstructured which requires significant processing to extract entities and information. Improving analytic and modeling systems that provide reliable and robust statistical estimated using data from multiple sources.	
Big Data Specific Challenges in Mobility	The outputs of this analysis and intelligence can be transmitted onto or accessed by the city representatives.	
Security & Privacy Requirements	Open data web portals and social networks like Twitter publicly release their data. Although, data-sources incorporate IoT meta-data, therefore, some policy for security and privacy protection must be implemented as required by various legal statutes.	
Highlight issues for generalizing this use case (e.g., for ref. architecture)	Definition of high-level data schema to incorporate multiple data sources and types providing structured data format. Therefore, it requires integrated complex event processing and event-based methods that will span domains.	
More Information (URLs)		
Note:		

Table 23: NIST Big Data Use Case

C. Application framework – City assessment tool

List of requirements for the following kind of Smart City applications	
Category:	Public safety, policy & Em.Res.
Sub-Category:	City surveillance and crime prevention
ICT Levels:	
Geo-Domanis:	

Aspect	Concern	Abstract requirements	Specific implementation requirements
Functional	**Actuation**	- to get data from surveillance systems - to elaborate data received surveillance systems	- sensors - security devices - actuation capabilities
	Communication		
	Controllability	- to remotely control/access to the systems	- Internet connection - remote control software - security/privacy protocols
	Physical context	- to precisely identify the location of people	- placement sensors - motion sensors

	Sensing	- to precisely identify the location of people - persistent communications - capacity to analyze and elaborate received data and make decisions	- security devices - sensors - persistent communications technology - decision maker systems
	Monitorability		
Business	**Quality**	- to provide feedback in time to act	- fast and reliable network - real-time systems
	Utility	- to provide useful information to reduce costs - to improve the quality of life of residents	- fast and reliable network - real-time systems
Human	**Usability**	- to provide human readable, unambiguous and harmonized data	
Trustworthiness	**Safety**	- persistent monitoring - to provide data in time to act	- fast and reliable network - real-time systems
	Privacy		

	Security	- to preserve authorized restrictions on access and disclosure - to prevent modification or destruction of the system - to ensure non-repudiation and authenticity - to ensure timely and reliable access to and use of a system	- firewall - antispyware - antivirus
Timing	**Logical time**	- to take into account the sequence of the events	
	Time awareness		
	Managing timing and latency	- to send data in a timely manner	
	Synchronization	- to send data with a common time scale	
Data	**Data semantics**	- to correctly understand the meaning of the data	
	Operations on data	- to harmonize data from different sources	
	Relationship between data	- to connect data from different sources - public, shared and standard data models	

Boundaries	**Behavioral**		

Table 24. City assessment tool

D. Experiments environment

All work was performed on Dell Latitude E7440 with Intel Core i7-4600U CPU 2.10 GHz, and 16GB RAM under Windows 7 Service Pack 1 package.

Experiments were conducted in the R statistical computing language using R studio environment (version 1.0.136). For the evaluation of the event model, including preprocessing, analytics which includes implementation of PGM and PR, and visualization are used the following packages and functions: Rgraphviz, gRim, gRbase, gRain, caret/sandwich, and plyr and gml. For the evaluation of dynamic network, the model has used the vars package, it was used for implementing AIC, Granger test, AR models, and VAR models, while graphics were generated using the ggplot package.

List of Publications

" A scientific publication is considered scholarly if it is authored by academic or professional researchers and targeting at an academic or related audience. "
- Muktikesh Dash

Journals :

(In progress) Survey : Analytical methods for Smart Cities/Using statistical prediction models with social media.

Kotevska, O., Kusne, G.A., Samarov, V. D., Lbath, A., Battou, A. (2017). Dynamic Network Model for Smart City Data-loss Resilience. *IEEE Access - Advanced Data Analytics for Large-scale Complex Data Environments, 2, 1-12.*

Kotevska, O., Lbath, A., (2017). Sentiment analysis of Social Sensors for Local Service Improvements. *International Journal of Computing and Digital Systems*, 6(4).

Kotevska, O., Lbath, A., & Bouzefrane, S. (2016). Toward a Real-Time Framework in Cloudlet-Based Architecture. *Tsinghua Science and Technology Journal*, *21*, 80-88.

Conferences and workshops:

(In progress) Multi-view regression for detecting safe areas in cities.

Kotevska, O., Lbath, A., & & Gelernter, J. (2016). Event Model to Facilitate Data Sharing Among Services. In *IEEE 3rd World Forum Internet of Things (WF-IoT)* (pp. 577-584).

Kotevska, O., Padi, S., & Lbath, A. (2016). Automatic Categorization of Social Sensor Data. *Procedia Computer Science*, *98*, 596-603.

Kotevska, O., Lbath, A., & Bouzefrane, S. (2015). Toward a Framework for Cloudlet-Based Architecture for a Real-Time Predictional Model. In *IEEE International Conference on Cloud and Big Data Computing*.

Posters :

“Machine Leraning for IoT and Smart City Data Sharing”, ITL Science Day, NIST, November 2, 2017.

"Framework for Network Event Detection and Analysis," ACM Women in Computing, Barcelona, Spain, September 6-8, 2017.

"Mobile Cloud Computing project toward a Cloudlet based Architecture framework for the assessment of services, exchanges, security, and metrics," ITL Science Day, NIST, October 27, 2015.

Bibliography

[1] Adi, A., Botzer, D., & Etzion, O. (2000). Semantic event model and its implication on situation detection. *ECIS 2000 Proceedings*, 2.

[2] Ahmed, K. B., Radenski, A., Bouhorma, M., & Ahmed, M. B. (2016, January). Sentiment Analysis for Smart Cities: State of the Art and Opportunities. In *Proceedings of the International Conference on Internet Computing (ICOMP)* (p. 55). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[3] Agafonov, A., & Myasnikov, V. (2015, April). Traffic flow forecasting algorithm based on a combination of adaptive elementary predictors. In *International Conference on Analysis of Images, Social Networks, and Texts* (pp. 163-174). Springer International Publishing.

[4] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R. Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media. Association for Computational Linguistics; 2011, p. 30–38.

[5] Akbar, A., Carrez, F., Moessner, K., & Zoha, A. (2015, December). Predicting complex events for pro-active IoT applications. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on* (pp. 327-332).

[6] Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart Cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, *22*(1), 3-21.

[7] Albakour, M. D., Macdonald, C., Ounis, I., Pnevmatikakis, A., & Soldatos, J. (2012, August). SMART: An open source framework for searching the physical world. In *SIGIR 2012 Workshop on Open Source Information Retrieval* (pp. 48-51).

[8] Albek, E., Bax, E., Billock, G., Chandy, K.M., Ian Swett: An Event Processing Language (EPL) for Building Sense and Respond Applications. IPDPS 2005.

[9] Alves, A., Mishra, S. (). Design Patterns for Complex Event Processing (CEP). Oracle Corporation. http://www.oracle.com/technetwork/server-storage/ts-4783-159494.pdf

[10] Aman, S.; Frincu, M.; Chelmis, C.; Noor, M.; Simmhan, Y.; and Prasanna, V. (2015). Prediction models for dynamic demand response: Requirements, challenges, and insights. In IEEE International Conference on Smart Grid Communications.

[11] Anantharam, P., Thirunarayan, K., & Sheth, A. P. (2013). Traffic analytics using probabilistic graphical models enhanced with knowledge bases.

[12] Anthopoulos, L. G. (2015). Understanding the smart city domain: A literature review. In *Transforming city governments for successful smart cities* (pp. 9-21). Springer International Publishing.

[13] Aphinyanaphongs, Y., Lulejian, A., Brown, D.P., Bonneau, R., Krebs, P. Text classification for automatic detection of e-cigarette use and use for smoking cessation from Twitter: A feasibility pilot. In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*; vol. 21. NIH Public Access; 2016, p. 480.

[14] Artikis, A., Etzion, O., Feldman, Z., & Fournier, F. (2012, July). Event processing under uncertainty. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems* (pp. 32-43). ACM.

[15] Artikis, A., Weidlich, M., Gal, A., Kalogeraki, V., & Gunopulos, D. (2013, October). Self-adaptive event recognition for intelligent transport management. In *Big Data, 2013 IEEE International Conference on* (pp. 319-325). IEEE.

[16] Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Keith Ito, Itaru Nishizawa, Justin Rosenstein, Jennifer Widom: STREAM: The Stanford Stream Data Manager. SIGMOD Conference 2003: 665.

[17] Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection on Twitter. *Computational Intelligence*, *31*(1), 132-164.

[18] Asaf Adi, Opher Etzion: Amit - the situation manager. VLDB J. 13(2): 177-203 (2004).

[19] Astrova, I., Koschel, A., Lukanowski, J., Martinez, J. L. M., Procenko, V., & Schaaf, M. (2014).

Ontologies for complex event processing. *International Journal of Computer, Information Science, and Engineering*, *8*(5), 556-566.

[20] Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence*, *31*(1), 132-164.

[21] Avvenuti, M., Cresci, S., La Polla, M.N., Marchetti, A., Tesconi, M. Earthquake emergency management by social sensing. In: Pervasive Computing and Communications Workshops (PERCOM Workshops). IEEE; 2014, p. 587–592.

[22] Baekgaard, L, (2002) Event modeling in UML. Issues & Trends of Information Technology Management in Contemporary Organizations. Ed. Mehdi Khosrow-Pour (Information Resources Management Association, USA) 3 p.]

[23] Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002, June). Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems* (pp. 1-16). ACM.

[24] Baiardi, F., De Francesco, N., & Vaglini, G. (1986). Development of a debugger for a concurrent language. IEEE Transactions on Software Engineering, (4), 547-553.

[25] Ballesteros, J., Carbunar, B., Rahman, M., Rishe, N., & Iyengar, S. S. (2014). Towards safe cities: A mobile and social networking approach. *IEEE Transactions on Parallel and Distributed Systems*, *25*(9), 2451-2462.

[26] Ballesteros, J., Rahman, M., Carbunar, B., & Rishe, N. (2012, October). Safe cities. A participatory sensing approach. In *Local Computer Networks (LCN), 2012 IEEE 37th Conference on* (pp. 626-634). IEEE.

[27] Barnaghi, P., Bermudez-Edo, M., & Tönjes, R. (2015). Challenges for the quality of data in smart cities. *Journal of Data and Information Quality (JDIQ)*, *6*(2-3), 6.

[28] Balakrishnan, H., Balazinska, M., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Galvez, E. F., Salz, J., Stonebraker, M, Tatbul, N, Tibbetts, R, Zdonik, S.B.: Retrospective on Aurora. VLDB J. 13(4): 370-383 (2004).

[29] Benson, K., Fracchia, C., Wang, G., Zhu, Q., Almomen, S., Cohn, J., .. & Venkatasubramanian, N. (2015). Scale: Safe community awareness and alerting leveraging the internet of things. *IEEE Communications Magazine*, *53*(12), 27-34.

[30] Bermingham, A., Smeaton, A.F. Classifying sentiment in microblogs: Is brevity an advantage? In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. New York, USA: ACM; 2010, p. 1833–1836.

[31] Brants, T., Chen, F., & Farahat, A. (2003, July). A system for new event detection. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 330-337). ACM.

[32] Breiman, L. Random forests. *Machine learning* 2001;45(1):5–32.

[33] Bates, J., Bacon, J., Moody, K., Spiteri, M.D.: Using events for the scalable federation of heterogeneous components. ACM SIGOPS European Workshop 1998: 58- 65

[34] Blaschke, T., Hay, G. J., Weng, Q., & Resch, B. (2011). Collective Sensing: Integrating Geospatial Technologies to Understand Urban Systems—An Overview. *Remote Sensing*, *3*(12), 1743–1776. http://doi.org/10.3390/rs3081743

[35] Chakravarthy, S., & Jiang, Q. (2009). Stream data processing: A quality of service perspective modeling, scheduling, load shedding, and complex event processing (1st ed.). Berlin: Springer Publishing Company, Incorporated.

[36] Chan, A. B., & Vasconcelos, N. (2009, September). Bayesian poisson regression for crowd counting. In Computer Vision, 2009 IEEE 12th International Conference on (pp. 545-551). IEEE.

[37] Chen, Y., Lee, G. M., Shu, L., & Crespi, N. (2016). Industrial Internet of Things-based collaborative sensing intelligence: framework and research challenges. *Sensors*, *16*(2), 215.

[38] Chen, H., Finin, T., & Joshi, A. (2005). The SOUPA ontology for pervasive computing. In Ontologies

for agents: Theory and experiences (pp. 233-258). Birkhäuser Basel.
[39] Cheong, M., & Lee, V. C. (2011). A microblogging-based approach to terrorism informatics: Exploration and chronicling civilian sentiment and response to terrorism events via Twitter. *Information Systems Frontiers*, *13*(1), 45-59.
[40] Cocchia, A. (2014). Smart and digital city: A systematic literature review. In the *Smart City* (pp. 13-43). Springer International Publishing.
[41] Cugola, G., & Margara, A. (2012). Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, *44*(3), 15.
[42] Della Valle, E., Ceri, S., Van Harmelen, F., & Fensel, D. (2009). It is a streaming world! Reasoning upon the rapidly changing information. *IEEE Intelligent Systems*, *24*(6).
[43] Derguech, W., Bruke, E., & Curry, E. (2014, December). An Autonomic Approach to Real-Time Predictive Analytics using Open Data and Internet of Things. In Ubiquitous Intelligence and Computing, 2014 IEEE 11th Intl Conf on and IEEE 11th Intl Conf on and Autonomic and Trusted Computing, and IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UTC-ATC-ScalCom) (pp. 204-211).
[44] Doerr, M, Ore, C.E. & Stead, S. The CIDOC conceptual reference model: a new standard for knowledge sharing. In Conceptual modeling. Australian Computer Society, Inc., 2007
[45] Dvijotham, K., Chertkov, M., Van Hentenryck, P., Vuffray, M., & Misra, S. (2017). Graphical models for optimal power flow. Constraints, 22(1), 24-49.
[46] Ekin, A, Tekalp, A. M, & Mehrotra, R. Integrated semanticsyntactic video modeling for search and browsing. IEEE Transactions on Multimedia, 6(6), 2004.
[47] Etzion, O., & Niblett, P. (2010). *Event processing in action*. Manning Publications Co.
[48] Etzion, O. (2010). Event processing: past, present, and future. *Proceedings of the VLDB Endowment*, *3*(1-2), 1651-1652.
[49] Etzion, O., & Forunier, F. (2014, November). On the Personalization of Event-Based Systems. In *Proceedings of the 1st ACM International Workshop on Human Centered Event Understanding from Multimedia* (pp. 45-48). ACM.
[50] Famoye, F., Wulu, J. T., & Singh, K. P. (2004). On the generalized Poisson regression model with an application to accident data. Journal of Data Science, 2(2004), 287Á295.
[51] Farasat, A., Nikolaev, A., Srihari, S. N., & Blair, R. H. (2015). Probabilistic graphical models in modern social network analysis. *Social Network Analysis and Mining*, *5*(1), 62.
[52] Fowler, C., & Qasemizadeh, B. (2009). Towards a common event model for an integrated sensor information system. In 1st International Workshop on the Semantic Sensor Web (SemSensWeb), 13 p.]
[53] Francois, A. R. J., Nevatia, R, Hobbs, J, & Bolles, R. C. VERL: An ontology framework for representing and annotating video events. IEEE MultiMedia, 12(4), 76-86.
[54] Fülöp, L. J., Beszédes, Á., Tóth, G., Demeter, H., Vidács, L., & Farkas, L. (2012, September). Predictive complex event processing: a conceptual framework for combining complex event processing and predictive analytics. In *Proceedings of the Fifth Balkan Conference in Informatics* (pp. 26-31). ACM.
[55] Fujitsu Develops Distributed and Parallel Complex Event Processing Technology that Rapidly Adjusts Big Data Load Fluctuations Online. Available: http://www.fujitsu.com/global/news/pr/archives/month/2011/20111216-02.html. Last time accessed: 26/02/2016.
[56] Gal, A., Wasserkrug, S., & Etzion, O. (2011). Event processing over uncertain data. In *Reasoning in Event-Based Distributed Systems* (pp. 279-304). Springer Berlin Heidelberg.
[57] Gatziu, S., & Dittrich, K. R. (1994, February). Detecting composite events in active database systems using Petri nets. In *Research Issues in Data Engineering, 1994. Active Database Systems. Proceedings Fourth International Workshop on* (pp. 2-9). IEEE.
[58] Gerber, M. S. (2014). Predicting crime using Twitter and kernel density estimation. Decision Support Systems, 61, 115-125.

[59] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, *1*(12).

[60] Gomez, C., & Paradells, J. (2015). Urban Automation Networks: Current and Emerging Solutions for Sensed Data Collection and Actuation in Smart Cities. *Sensors*, *15*(9), 22874–22898. http://doi.org/10.3390/s150922874

[61] Goswami, A., & Kumar, A. (2016). A survey of event detection techniques in online social networks. *Social Network Analysis and Mining*, *6*(1), 107.

[62] Granger, C. W. J. (1969). "Investigating Causal Relations by Econometric Models and Cross-spectral Methods". *Econometrica*. **37** (3): 424–438.

[63] Gunderson, L., & Brown, D. (2000). Using a multi-agent model to predict both physical and cyber-criminal activity. In Systems, Man, and Cybernetics, 2000 IEEE International Conference on (Vol. 4, pp. 2338-2343). IEEE.

[64] Gutiérrez, V., Galache, J. A., Sánchez, L., Muñoz, L., Hernández-Muñoz, J. M., Fernandes, J., & Presser, M. (2013, May). SmartSantander: Internet of things research and innovation through citizen participation. In *The Future Internet Assembly* (pp. 173-186). Springer Berlin Heidelberg.

[65] Ha, J., Yoon, J., Heo, J., Han, Y., Jung, J., Yun, Y. S., & Eun, S. (2015, October). A perspective on the IoT services through a multi-dimensional analysis. In *Proceedings of the 2015 Conference on research in adaptive and convergent systems* (pp. 479-481). ACM.

[66] Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, *17*, 26-32.

[67] Hall, P. (2000). Creative cities and economic development. *Urban Studies*, *37*(4), 639-649.

[68] Halpern, J. Y. (1998). A logical approach to reasoning about uncertainty: a tutorial. In *Discourse, Interaction and Communication* (pp. 141-155). Springer Netherlands.

[69] Hancke, G. P., de Silva, B. de C., & Hancke, G. P. (2013). *The role of advanced sensing in smart cities*. *Sensors (Switzerland)* (Vol. 13). http://doi.org/10.3390/s130100393

[70] Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnanam, J., Paraszczak, J., & Williams, P. (2010). Foundations for smarter cities. *IBM Journal of Research and Development*, *54*(4), 1-16.

[71] Hasan, S., & Curry, E. (2014, December). Thematic event processing. In *Proceedings of the 15th International Middleware Conference* (pp. 109-120). ACM.

[72] Horváth, C. (2003). Dynamic analysis of marketing systems.

[73] Hromic, H., Le Phuoc, D., Serrano, M., Antonić, A., Žarko, I. P., Hayes, C., & Decker, S. (2015, June). Real time analysis of sensor data for the Internet of Things by means of clustering and event processing. In 2015 IEEE International Conference on Communications (ICC) (pp. 685-691). IEEE.

[74] Hvistendahl, M. (2016). Crime forecasters. Science, 353(6307), 1484-1487.

[75] Ilina, E., Hauff, C., Celik, I., Abel, F., & Houben, G. J. (2012, July). Social event detection on Twitter. In *International Conference on Web Engineering* (pp. 169-176). Springer Berlin Heidelberg

[76] Injadat, M., Salo, F., & Nassif, A. B. (2016). Data mining techniques in social media: A survey. *Neurocomputing*, *214*, 654-670.

[77] Jiang, Y., Xu, Z., Wang, X. (2014). The construction of ontology in the area of traffic violations. J.J. Park et al. (Eds). *Future Information Technology. Lecture Notes in Electrical Engineering 309*, 379-384.

[78] Jo, T., & Lee, M. (2007, June). The evaluation measure of text clustering for the variable number of clusters. In International Symposium on Neural Networks (pp. 871-879). Springer Berlin Heidelberg.

[79] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Springer.

[80] John R Boyd. The essence of winning and losing. *Unpublished lecture notes*, 1996.

[81] Kern-Isberner, G., & Lukasiewicz, T. (2017). Many Facets of Reasoning Under Uncertainty, Inconsistency, Vagueness, and Preferences: A Brief Survey. *KI-Künstliche Intelligenz*, 1-5.

[82] Kolozali, S., Bermudez-Edo, M., Puschmann, D., Ganz, F., & Barnaghi, P. (2014, September). A knowledge-based approach for real-time iot data stream annotation and processing. In *Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications*

(GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE (pp. 215-222). IEEE.

[83] Kyriazopoulou, C. (2015, May). Smart city technologies and architectures: A literature review. In *Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on* (pp. 1-12). IEEE.

[84] Leong, K., & Sung, A. (2015). A review of spatio-temporal pattern analysis approaches on crime analysis. International E-Journal of Criminal Sciences.

[85] Liao, R., Wang, X., Li, L., & Qin, Z. (2010, July). A novel serial crime prediction model based on bayesian learning theory. In 2010 International Conference on Machine Learning and Cybernetics (Vol. 4, pp. 1757-1762). IEEE.

[86] Lin, J. (2008). Notes on testing causality. *Institute of Economics, Academia Sinica, Department of Economics, National Chengchi University*.

[87] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture. *NIST special publication, 500*(2011), 292.

[88] Li, M.; Lou, W.; Ren, K. Data security and privacy in wireless body area networks. *IEEE Wirel. Commun.* 2010, *17*, 51–58.

[89] Li, X., Martinez, J. F., Eckert, M., & Rubio, G. (2017). Uncertainty Quantification in Mathematics-embedded Ontologies Using Stochastic Reduced Order Model. *IEEE Transactions on Knowledge and Data Engineering*.

[90] Liu, Q.; Zhang, X.; Chen, X.; Wang, L. The resource access authorization route problem in a collaborative manufacturing system. *J. Intell. Manuf.* 2014, *25*, 413–425.

[91] Luckham, D. (2002). *The power of events* (Vol. 204). Reading: Addison-Wesley.

[92] Luckham, D.C., Vera, J.: An Event-Based Architecture Definition Language. IEEE Trans. Software Eng. 21(9): 717-734 (1995).

[93] Nakamura, M., & Du Bousquet, L. (2015, July). Constructing execution and life-cycle models for smart city services with self-aware IoT. In *Autonomic Computing (ICAC), 2015 IEEE International Conference on* (pp. 289-294). IEEE.

[94] Ma, M., Wang, P., Yang, J., Li, C., (2015). OntoEvent: An ontology-based event description language for semantic complex event processing. J. Li and Y. Sun (Eds). *WAIM 2015, LNCS 9098*, 448-451.

[95] Ma, M., Wang, P., Chu, C. H., & Liu, L. (2015). Efficient Multipattern Event Processing Over High-Speed Train Data Streams. *IEEE Internet of Things Journal*, *2*(4), 295-309.

[96] Mangurian, C., Keenan, W., Newcomer, J. W., Vittinghoff, E., Creasman, J. M., & Schillinger, D. (2017). Diabetes Prevalence Among Racial-Ethnic Minority Group Members With Severe Mental Illness Taking Antipsychotics: Double Jeopardy?. *Psychiatric Services*, appi-ps.

[97] Mandel, B., Culotta, A., Boulahanis, J., Stark, D., Lewis, B., & Rodrigue, J. (2012, June). A demographic analysis of online sentiment during hurricane irene. In *Proceedings of the Second Workshop on Language in Social Media* (pp. 27-36). Association for Computational Linguistics.

[98] Mendes, M., Bizarro, P., & Marques, P. (2013, April). Towards a standard event processing benchmark. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering* (pp. 307-310). ACM.

[99] Mccord, M., & Chuah, M. (2011, September). Spam detection on twitter using traditional classifiers. In *International Conference on Autonomic and Trusted Computing* (pp. 175-186). Springer Berlin Heidelberg.

[100] Miaou, S. P. (1994). The relationship between truck accidents and geometric design of road sections: Poisson versus negative binomial regressions. Accident Analysis & Prevention, 26(4), 471-482.

[101] Michener, R., & Tighe, C. (1992). A Poisson regression model of highway fatalities. The American Economic Review, 82(2), 452-456.

[102] Molineaux, M., & Aha, D. W. (2014). *Learning unknown event models*. KNEXUS RESEARCH CORP SPRINGFIELD VA.

[103] Mookiah, L., Eberle, W., & Siraj, A. (2015, April). Survey of Crime Analysis and Prediction. In FLAIRS Conference (pp. 440-443).

[104] Morgan, R. T. (2014). EVENT MINING IN SOCIAL NETWORKS, *1*(12).

[105] Nonejad, N. (2017). Forecasting aggregate stock market volatility using financial and macroeconomic predictors: Which models forecast best, when and why. Journal of Empirical Finance, 42, 131-154.

[106] Osgood, D. W. (2000). Poisson-based regression analysis of aggregate crime rates. Journal of quantitative criminology, 16(1), 21-43.

[107] Papadopoulos, S., Scherp, A., Ireson, N., Tsampoulatidis, I., & Kompatsiaris, Y. (2010, October). Using event representation and semantic enrichment for managing and reviewing emergency incident logs. In *Proceedings of the 2nd ACM international workshop on Events in multimedia* (pp. 41-46). ACM.

[108] Paschke, A. (2008). Design patterns for complex event processing. *arXiv preprint arXiv:0806.1100*.

[109] Paschke, A. (2009). A Semantic Design Pattern Language for Complex Event Processing. In *AAAI Spring Symposium: Intelligent Event Processing* (pp. 54-60).

[110] Paul, M. J., & Dredze, M. (2012). A model for mining public health topics from Twitter. *Health*, *11*, 16-6.

[111] Paz, A., Veeramisti, N., & de la Fuente-Mella, H. (2015, September). Forecasting performance measures for traffic safety using deterministic and stochastic models. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on* (pp. 2965-2970). IEEE.

[112] Pongpaichet, S., Singh, V.K., Gao, M., Jain, R., (2013). EventShop: Recognizing situations in web data streams. *WWW2013 Companion, May 13-17, 2013, Rio de Janeiro, Brazil*, 1359-1367.

[113] Pravilovic, S., Bilancia, M., Appice, A., & Malerba, D. (2017). Using multiple time series analysis for geosensor data forecasting. Information Sciences, 380, 31-52.

[114] Presser, M., Vestergaard, L., & Ganea, S. (2014). Smart City Use Cases and Requirements. *CityPulse Project Deliverable D*, *2*.

[115] Radovanovic, M. and Ivanovic, M. (2008). Text mining: Approaches and applications. Novi Sad J. Math, 38(3):227–234.

[116] Raimond, Y. Abdallah, S., Sandler, M., Giasson, F., (2007). The music ontology. Austrian Computer Society, 6 p. http://raimond.me.uk/pubs/Raimond-ISMIR2007-Submitted.pdf

[117] Ranson, M. (2014). Crime, weather, and climate change. Journal of environmental economics and management, 67(3), 274-302.

[118] Rao, B. P., Saluia, P., Sharma, N., Mittal, A., & Sharma, S. V. (2012, December). Cloud computing for Internet of Things & sensing based applications. In *Sensing Technology (ICST), 2012 Sixth International Conference on* (pp. 374-380). IEEE.

[119] Roche, S., & Rajabifard, A. (2012, August). Sensing places' life to make the city smarter. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing* (pp. 41-46). ACM.

[120] Romsaiyud, W. (2013). Detecting emergency events and geo-location awareness from Twitter streams. In *The International Conference on E-Technologies and Business on the Web (EBW2013)* (pp. 22-27). The Society of Digital Information and Wireless Communication.

[121] Rosa, K.D, Shah, R., Lin, B., Gershman, A., Frederking, R. Topical clustering of tweets. Proceedings of the ACM SIGIR: SWSM 2011.

[122] Saif, H., He, Y., Alani, H. Semantic sentiment analysis of twitter: The Semantic Web–ISWC 2012. Springer; 2012, p. 508–524.

[123] Sakaki, T., Okazaki, M., Matsuo, Y. Earthquake shakes twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web; WWW '10. New York, USA: ACM; 2010, p. 851–860.

[124] Sarma, S., Venkatasubramanian, N., & Dutt, N. (2014, June). Sense-making from distributed and mobile sensing data: A middleware perspective. In *Proceedings of the 51st Annual Design Automation Conference* (pp. 1-6). ACM.

[125] Sagl, G., Resch, B., & Blaschke, T. (2015). Contextual Sensing: Integrating Contextual Information with Human and Technical Geo-Sensor Information for Smart Cities. *Sensors*, *15*(7), 17013–17035. http://doi.org/10.3390/s150717013

[126] Salas-Zárate, M. D. P., Medina-Moreira, J., Álvarez-Sagubay, P. J., Lagos-Ortiz, K., Paredes-Valverde, M. A., & Valencia-García, R. (2016). Sentiment Analysis and Trend Detection in Twitter. In *Technologies and Innovation: Second International Conference, CITI 2016, Guayaquil, Ecuador, November 23-25, 2016, Proceedings* (pp. 63-76). Springer International Publishing.

[127] Scherp, A., Franz, T. Saathoff, C. and Staab, S. (2009). F—A Model of Events based on the foundational ontology DOLCE+DnS Ultralite. *Knowledge Capture, '09, September 1-4, Redondo Beach, California*, 137-144.

[128] Schwartz, R., Naaman, M., & Matni, Z. (2013, June). Making sense of cities using social media: Requirements for hyper-local data aggregation tools. In *Proceedings of the International AAAI Conference on Weblogs and Social Media* (pp. 15-22).

[129] Schnizler, F., Liebig, T., Marmor, S., Souto, G., Bothe, S., & Stange, H. (2014, October). Heterogeneous stream processing for disaster detection and alarming. In *Big Data (Big Data), 2014 IEEE International Conference on* (pp. 914-923). IEEE.

[130] Sebastiani, F. (1999). A tutorial on automated text categorisation. In Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence (pp. 7–35). Buenos Aires, AR.

[131] Segnon, M., Lux, T., & Gupta, R. (2017). Modeling and forecasting the volatility of carbon dioxide emission allowance prices: A review and comparison of modern volatility models. Renewable and Sustainable Energy Reviews, 69, 692-704.

[132] Shanahan, M. (1999). The event calculus explained. In Artificial intelligence today (pp. 409-430). Springer Berlin Heidelberg.

[133] Sharon, G., & שרון גיא. (2007). *Event processing network-A conceptual model*. Technion-Israel Institute of Technology, Faculty of Industrial and Management Engineering.

[134] Shaw, R., Troncy, R., Hardman, L. (2009). LODE: Linking Open Descriptions of Events in *'The Semantic Web*', Springer Berlin / Heidelberg, pp. 153-167.

[135] Skarlatidis, A., Michelioudakis, E., Katzouris, N., Artikis, A., Paliouras, G., Alevizos, E., ... & Vlassopoulos, C. (2014). Event Recognition and Forecasting Technology (Part 2).

[136] Su, K., Li, J., & Fu, H. (2011, September). Smart city and the applications. In *Electronics, Communications, and Control (ICECC), 2011 International Conference on* (pp. 1028-1031). IEEE.

[137] Sokha, Y., Jeong, K., Lee, J., & Joe, W. (n.d.). A Complex Event Processing System Approach to Real-Time Road Traffic Event Detection.

[138] Starling, S. (2017). Genetic variation: Linear INSIGHTs into non-coding DNA. *Nature Reviews Genetics*.

[139] Tan, Y., Vuran, M. C., Goddard, S., Yu, Y., Song, M., & Ren, S. (2010, April). A concept lattice-based event model for Cyber-Physical Systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-physical Systems* (pp. 50-60). ACM.

[140] Taylor, K., & Leidinger, L. (2011, May). Ontology-driven complex event processing in heterogeneous sensor networks. In *Extended Semantic Web Conference* (pp. 285-299). Springer Berlin Heidelberg.

[141] Thelwall, M., Buckley, K., & Paltoglou, G. (2011). Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology*, *62*(2), 406-418.

[142] Teimourikia, M., & Fugini, M. (2017). Ontology development for run-time safety management methodology in Smart Work Environments using ambient knowledge. *Future Generation Computer Systems*, *68*, 428-441.

[143] Tokumitsu, M., Hasegawa, K., & Ishida, Y. (2015). Toward Resilient Sensor Networks with Spatiotemporal Interpolation of Missing Data: An Example of Space Weather Forecasting. Procedia Computer Science, 60, 1585–1594.

[144] Townsend, A. M. (2013). *Smart cities : big data, civic hackers, and the quest for a new utopia*. New

York NY: W.W. Norton & Company.
[145] Theodore, D. (2006). The sense of the City: An Alternative Approach to Urbanism-Edited by Mirko Zardini.
[146] Tsai, C. W., Lai, C. F., Chiang, M. C., & Yang, L. T. (2014). Data mining for Internet of Things: A survey. *IEEE Communications Surveys and Tutorials*, *16*(1), 77-97.
[147] Tsimelzon, M. (2006). Complex Event Processing: Ten Design Patterns.
[148] Uma, V., and Aghila, G. (2014). Event order generation using Reference Event based qualitative Temporal (REseT) relations in Time Event Ontology. *Central European Journal of Computer Science 4*(1): 12-29.
[149] Wang, W., Kennedy, R., Lazer, D., & Ramakrishnan, N. (2016). Growing pains for global monitoring of societal events. *Science*, *353*(6307), 1502-1503.
[150] Wang, Y., & Zhang, X. (2014). A Proactive Parallel Complex Event Processing Method for Large-Scale Intelligent Transportation Systems. *International Journal of Multimedia and Ubiquitous Engineering*, *9*(11), 111–122. http://doi.org/10.14257/ijmue.2014.9.11.11
[151] Wang, X, Mamadgi, S, Thekdi, A., Kelliher, A., & Sundaram, H. Eventory - an event based media repository. In Semantic Computing. IEEE, 2007.
[152] Wang, Y., & Kuang, L. (2015). A Traffic Prediction Method based on Complex Event Pprocessing and Adaptive Bayesian Networks.
[153] Wang, W., & Guo, D. (2012, October). Towards unified heterogeneous event processing for the Internet of Things. In Internet of Things (IOT), 2012 3rd International Conference on the (pp. 84-91). IEEE.
[154] Ward, J. H., Jr. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58, 236–244.
[155] Westermann, U, and Jain, R. (2007). Toward a common event model for multimedia applications. IEEE multimedia 14(1): 19-29.
[156] Widder, A., Ammon, R. V., Schaeffer, P., & Wolff, C. (2007, June). Identification of suspicious, unknown event patterns in an event cloud. In *Proceedings of the 2007 inaugural international conference on Distributed event-based systems* (pp. 164-170). ACM.
[157] Wu, H., Cao, J., Fan, X., Wu, H., Cao, J., & Fan, X. (n.d.). Dynamic, collaborative in-network event detection in wireless sensor networks. http://doi.org/10.1007/s11235-015-9981-0
[158] Zámečníková, E., & Kreslíková, J. (2015, November). Comparison of platforms for high-frequency data processing. In *Scientific Conference on Informatics, 2015 IEEE 13th International* (pp. 296-301). IEEE.
[159] Zhao, J., Wang, X., & Ma, Z. (2014). Towards events detection from microblog messages. *International Journal of Hybrid Information Technology*, *7*(1), 201-210.
[160] Zhou, T., Gao, L., & Ni, D. (2014, April). Road traffic prediction by incorporating online information. In *Proceedings of the 23rd International Conference on World Wide Web* (pp. 1235-1240). ACM.
[161] Zhu, X., Kui, F., & Wang, Y. (2013). Predictive analytics by using Bayesian model averaging for large-scale Internet of Things. *International Journal of Distributed Sensor Networks*, *2013*.
[162] Zhong, Z., Liu, Z., Li, C., & Guan, Y. (2012). Event ontology reasoning based on event class influence factors. International Journal of Machine Learning and Cybernetics, 3(2), 133-139.