Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Anna Haugsbø Hermansen

Machine Learning for Spatio-Temporal Forecasting of Ambulance Demand

A Norwegian Case Study

Master's thesis in Computer Science
Supervisor: Ole Jakob Mengshoel
June 2021

NTNU
Norwegian University of
Science and Technology

Anna Haugsbø Hermansen

Machine Learning for Spatio-Temporal Forecasting of Ambulance Demand

A Norwegian Case Study

Master's thesis in Computer Science
Supervisor: Ole Jakob Mengshoel
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

In Emergency Medical Services (EMS), time is of the essence. It is crucial to distribute available resources strategically so that they can reach the scene of an incident quickly and ensure timely life-saving assistance to people in need. In order to do that, we need to have good estimates of when and where incidents are likely to occur. This thesis investigates how to best forecast the EMS demand in and around the capital of Norway based on historical EMS data and, to some lesser extent, weather data. We use a fine spatio-temporal resolution of 1x1km spatial regions and 1-hr time intervals. The EMS demand is forecast directly and using a split approach that looks at the volume and distribution of the demand separately. We use Multi Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) models to forecast the EMS demand, in addition to some simple aggregation methods. The neural network models are trained with different input sets consisting of simple temporal data and weather data to investigate how the forecast quality varies with varying input feature sets. We conclude from our experiments that the split approach is better suited for modeling EMS demand as the complete methods tend to underestimate the demand volume. We also show how online learning tends to improve the performance of the models. Among the models tested in this study, we find that a split model consisting of a simple aggregation distribution model and an MLP volume model with simple temporal input features produces the best forecasts. This split model produces better volume, distribution, and complete forecasts than a common industry practice method and the complete MLP model proposed by Setzler et al. [2009].

Sammendrag

I akuttmedisinen opererer man ofte i en kamp mot klokken. Man må fordele tilgjengelige ressurser strategisk slik at man kan nå mennesker i nød på kortest mulig tid og redde liv. For å kunne posisjonere ambulanser strategisk må vi vite hvor og når det er stor sannsynlighet for at hendelser skjer. Denne oppgaven tar for seg predikering av den timesvise akuttmedisinske etterspørselen i 1x1km geografiske områder i Oslo og Akershus. Vi sammenlikner komplette og splittede modeller. De splittede modellene predikerer det totale antallet hendelser og distribusjonen av hendelsene hver for seg, mens de komplette modellene predikerer antallet hendelser i hvert område direkte. Vi bruker hovedsaklig nevrale nettverk for å predikere etterspørselen, samt noen enkle aggregeringsmodeller. Vi undersøker om været påvirker den akuttmedisinske etterspørselen ved å inkludere værdata i noen av input-settene til de nevrale nettene. Resultatene våre tyder på at de splittede modellene er bedre egnet til å predikere den akuttmedisinske etterspørselen enn de komplette modellene, ettersom de komplette modellene har en tendens til å underestimere volumet av hendelser. Vi viser også at online trening er et godt verktøy som forbedrer prediksjonene til modellene. Blant modellene vi tester slår vi fast at en splittet modell med en enkel distributisjonsmodell basert på aggregering og en flerlags perceptron (MLP) volummodell med enkle temporale inputter har mest nytteverdi i vårt tilfelle. Denne modellen produserer bedre komplette, volum og distribusjons prediksjoner enn en standard industrimodell samt MLP-modellen foreslått i Setzler et al. [2009].

Preface

This thesis was written and carried out by Anna Haugsbø Hermansen as the finalization of her Master of Science in Computer Science degree at the Department of Computer Science under the Faculty of Information Technology and Electrical Engineering at the Norwegian University of Science and Technology (NTNU). Professor Ole Jakob Mengshoel at the Department of Computer Science under the Faculty of Information Technology and Electrical Engineering at NTNU supervised the thesis.

I want to thank Professor Mengshoel for his advice and feedback throughout this project. I would also like to thank the Norwegian National Advisory Unit for Prehospital Emergency Medicine (NAKOS) and the Department of Emergency Medical Communication Centre (EMCC) Division of Prehospital Services at Oslo University Hospital for the dataset used in this thesis. Lastly, I would like to acknowledge Professor Jeff Orchard at the University of Waterloo for sharing his lecture notes on neural networks.

Anna Haugsbø Hermansen
Trondheim, June 14, 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Emergency Medical Services (EMS) are a crucial part of modern health care systems. They respond to emergency calls and are responsible for the pre-hospital care and transportation of patients.

Many strategic, tactical, and operational decisions affect the quality of an EMS system, such as the fleet size, personnel management, location of ambulance stations and hospitals, equipment investment, and location, dispatching, and routing of ambulance units. EMS systems are shaped by the high level of uncertainty they operate in. There is uncertainty in the volume, severity, and location of incidents, the availability of ambulances, and ambulance travel times. Further, the trade-off between cost, effect, and equity is a substantial concern for policymakers. Extra resources are necessary for being able to handle high workloads, but resources that are never used pose a high and unnecessary cost. It is more cost-effective to focus resources around locations with high demand, but that means people living outside of these areas will have less access to those critical resources. However, providing equal access to resources when the cost of doing so is higher in some areas implies that people are valued higher in those areas, as noted in Erkut et al. [2008].

This thesis is focused on modeling the EMS demand in Oslo and Akershus, Norway. Doing so involves several challenges. In addition to the challenges mentioned above, demand forecasts are only useful for positioning resources if the forecasts have high resolutions on a spatio-temporal scale. However, high resolutions result in very sparse data which is difficult to model accurately.

1.1 EMS Timeline

EMS systems are implemented differently throughout the world, but most of them include the steps illustrated in Figure 1.1. When an incident occurs it might take some time before the public calls the emergency number. After a (usually) short amount of time, an operator from the Emergency Medical Dispatch (EMD) center answers the call. The emergency operator is responsible for determining the location and priority of the incident (triage) and providing guidance to the caller. Next, an operator has to decide precisely which ambulance unit to dispatch to the incident and to which facility the patient should be transported. The selected unit is notified and then has to gather all needed equipment and get in the ambulance. It then travels to the scene. At the scene of the incident, the unit might take some time to locate and reach the patient. When the patient is located, the unit will perform medical care before transporting the patient to the appropriate health care facility. At the destination, the ambulance personnel has to hand the patient over to the facility staff. Once the patient has been safely transferred, the ambulance might need cleaning or replenishing equipment before it is ready for another mission.

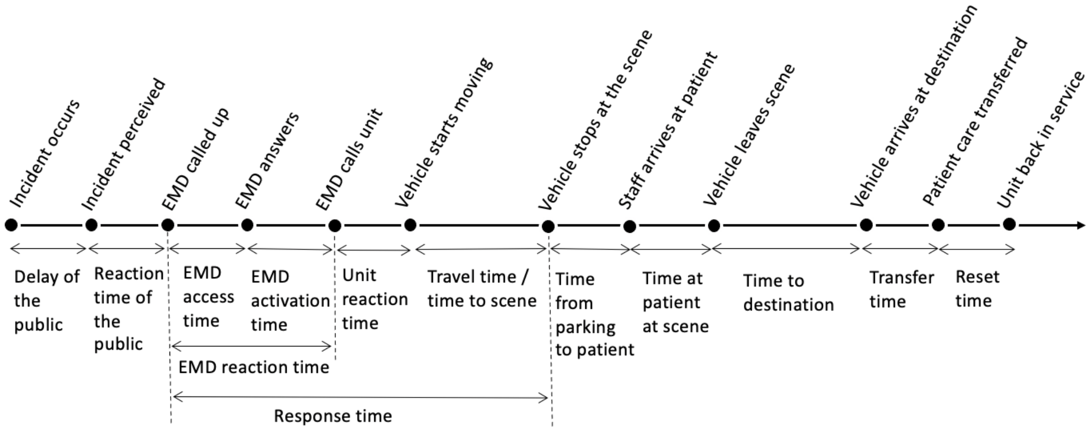


Figure 1.1: A general EMS timeline with named time points and intervals, adapted from Olsen et al. [2019].

1.2 Response Time

Receiving efficient treatment quickly is paramount for survival in certain acute incidents such as cardiac arrest, stroke, and serious trauma [The Norwegian Directory of Health, 2018; Haga et al., 1998]. In the case of out-of-hospital cardiac arrest (OHCA), studies have found that patient survival is inversely related to the time to defibrillation [Haga et al., 1998; Nolan et al., 2010; Larsen et al., 1993; O'Keeffe et al., 2011]. Nolan et al. [2010] found that each minute of delay before defibrillation reduces the likelihood of survival by 10-12%. Similarly, O'Keeffe et al. [2011] found that a one-minute reduction in time to treatment improves the odds of survival by 24%. In addition to the medical aspects, receiving treatment quickly is also important for the public's feeling of safety.

The response time in an EMS context is defined as the time between the notification of an incident and the arrival of an ambulance at the scene of the incidents, as illustrated in Figure 1.1. The response time as such consists of three independent parts: the EMD reaction time, the unit reaction time, and the travel time. All of these can be influenced by the EMS provider through efficient digital communication systems and triage protocols, and strategic location and dispatching of ambulance units. Note that the response time does not include the reaction time or delay of the public, nor the time it takes for the unit to reach the patient at the scene of the incident. Hence, the response time is a proxy for the time to treatment consisting of time intervals that we can measure and affect.

The most common system-wide performance indicator of an EMS system is response time statistics, such as the percentage of incidents that achieved a response time below some threshold. These percentages and thresholds are somewhat arbitrary and not directly connected to the medical outcomes of the patients. The fitness of such quality measures is questioned in Erkut et al. [2008] and Price [2006]. They emphasize the disparity between the fraction of missions reached within some threshold and the medical outcomes for the patients. The statistics, however, are easy to obtain and understand and are therefore still most widely used.

1.3 The Ambulance Location Problem

The ambulance location problem is about strategically choosing a set of standby sites and the number of ambulances that should be stationed at each of them to achieve some goal, such as minimizing the response time. Over the years, many mathematical models have been proposed to model this problem with varying assumptions and constraints. All such models need three things as input: 1) the demand that the ambulances should respond to (*i.e.* the emergency calls), 2) the time it takes to travel between different locations, and 3) the workload or busy time of the ambulances [Ingolfsson, 2013]. In a real-world application, we do not have access to the exact demand, travel times, or workload of the system, so we have to make do with forecasts. The location problem is computationally complex, so heuristics typically have to be used in practice to find a solution.

There is some variation in exactly what the different ambulance location models aim to optimize. Most of them involve the notion of *coverage*, in which an area is said to be covered if an ambulance is positioned such that it can travel to the area in question in less than some threshold time. Some models look at the minimum number of resources needed to provide some specified service level (typically coverage), such as the location set covering problem (LSCP) [Toregas et al., 1971] and the probabilistic model described by Ingolfsson et al. [2008]. Other models try to maximize the coverage given a certain amount of resources, such as the maximal covering location problem (MCLP) [Church and ReVelle, 1974]. In close relation to this, other models look at maximizing the coverage multiple times, such as the double standard model (DSM) [Gendreau et al., 1997] and the hierarchical objective set covering problem (HOSC) [Daskin and Stern, 1981]. Erkut et al. [2008] try to maximize survivability directly instead of focusing on the response/travel time. A different approach is presented by Chanta et al. [2011], which tries to minimize customer dissatisfaction or *envy* in their p-envy location problem formulation. Although the selection of a cost function has implications on the optimal locations of ambulances, different approaches can result in similar outcomes. McLay and Mayorga [2010] found that in the case of Hanover County, Virginia, locating ambulances to optimize seven and eight-minute response time thresholds were equivalent to optimizing patient survival, while nine and ten-minute thresholds improved survivability in rural areas, thus improving equity.

The models handle uncertainty in different ways. Most of the early models were deterministic, meaning they assumed that ambulances were always available and that the demand and travel times were constant. These unreasonable assumptions were relaxed in models such as the hypercube model [Larson, 1974] which uses a queueing theory approach to model ambulance unavailability. Daskin [1983] introduces a single "busy-factor," meaning the probability that an ambulance is available. Ingolfsson et al. [2008] address uncertainty in pre-travel delays and travel times as well as ambulance availability.

The models can also be characterized by how dynamic they are. Static models assume that when an ambulance is assigned to a site, it will always return to this once it has completed a mission. Such models do not account for the variability of travel time and demand over time. Multi-period models address such variations by dividing the problem into smaller time slots and solving a static model in each of them, such as the multi-period double standard model (mDSM) [Schmid and Doerner, 2010]. Dynamic models consider the state of the model, like how many ambulances are available and where they are located. This allows for dynamic relocation where units are moved in response to other units becoming unavailable/available. The relocation problem (RP^t) [Gendreau et al., 2001] is an example of such a dynamic model. It is built on top of the double standard model hence it maximizes the double coverage and minimizes relocation cost. Frequent relocation poses human resource problems as it is hard to satisfy the ambulance personnel's physical and social needs when they are in the ambulance for extended time periods.

For this reason, continuous relocation is rarely seen in practice.

For more thorough reviews and greater details of ambulance location models, we refer the reader to several good reviews on the topic. Bélanger et al. [2019] focus on models based on integer, stochastic, and dynamic programming. The paper includes mathematical formulations of many of the problems, and a handy taxonomy of the models can be found in the appendix. Another taxonomy with models from 57 papers can be found in Başar et al. [2012]. Aringhieri et al. [2017] discuss how the different ambulance location models handle uncertainty and equity. This review also includes models based on queueing theory, goal programming, and fuzzy programming.

Simulations have also been used widely to model EMS systems. Such simulations imitate the behavior of the system in question and serve as a natural way of validating solutions and testing the consequences of different strategies. They can handle many sources of uncertainty without being overwhelmed by time complexity because they are focused on evaluating a solution rather than finding a solution. The interested reader can find a review of the use of simulations in the EMS domain in Aboueljinane et al. [2013].

1.4 Goal and Research Questions

A minimal response time will benefit the medical outcome of a patient. We want to utilize the available resources more efficiently so that we can reduce the response time without incurring large expenses. More specifically, we focus on minimizing the travel time part of the response time by locating ambulances strategically. Hence the overall goal of this thesis is the following:

Goal *To minimize the EMS response time in Oslo and Akershus through strategic placement of ambulances.*

In order to position ambulances strategically, we need to forecast when and where incidents are likely to occur. In this thesis, we focus on such forecasting of EMS demand. We have defined three specific research questions that we try to answer in our work, as detailed below.

1.4.1 Research Question 1

We want to forecast the demand $\lambda(t) = \mathbf{u}^t \in \mathbb{R}_+^N$ such that u_k^t is the forecast number of incidents at time step t in region k for $k \in \{1, 2, ..., N\}$ where N is the number of spatial regions. The higher the spatio-temporal resolution of the EMS demand forecasts, *i.e.* the larger N is and the smaller each time step is, the more information we have for positioning ambulances strategically. The disadvantage of using such high resolutions is that the data becomes more sparse and stochastic, making forecasting the demand more difficult. This leads us to our first and most central research question:

Research question 1 *How can the EMS demand in Oslo and Akershus be forecast accurately at a fine spatio-temporal scale?*

In this thesis, we use a spatial resolution of 1x1km regions, which is the highest possible granularity in our case because of the limitations of our dataset. Our temporal resolution of 1 hour is the highest temporal granularity used in the literature.

We test various approaches, models, and input features to investigate how we can model the EMS demand as accurately as possible at these spatio-temporal resolutions.

1.4.2 Research Question 2

Most of the studies on EMS demand forecasting have tried to forecast the *complete* demand $\lambda(t)$ directly. However, it is also possible to use *split* approach that looks at the volume and the distribution of the demand separately. Let $\delta(t) \in \mathbb{R}_+$ be the aggregated volume of incidents at time step t such that $\delta(t) = \sum_{k=1}^{N} u_k^t$. Let $f(t) = \mathbf{v}^t \in \mathbb{R}_+^N$ be the spatial distribution of the incidents at time step t such that v_k^t is the fraction of the events in time step t that occurs in region k and $\sum_{k=1}^{N} v_k^t = 1$. A complete forecast at time step t can be obtained by combining a volume and distribution forecast: $\lambda(t) = \delta(t)f(t)$.

Almost all previous research has been on complete models, with the notable exception of Zhou and Matteson [2015]. We are interested in how the different approaches influence the forecasts and which is better for predicting EMS demand:

Research question 2 *Is a split model or a complete model better at forecasting EMS demand in Oslo and Akershus?*

We believe that the split approach can produce better results because it aggregates the data, which can combat data sparseness. It might also improve the interpretability of the forecasts.

1.4.3 Research Question 3

Weather has been shown to be related to the daily EMS demand volume in large cities [Wong and Lai, 2010, 2013; Thornes et al., 2014; Wong and Lin, 2020]. However, to the author's knowledge, no one has investigated whether the weather influences the EMS demand at a fine spatio-temporal scale.

Research question 3 *Does weather influence the spatial distribution of EMS demand in Oslo and Akershus?*

The EMS demand is often assumed to follow a non-homogeneous Poisson process in which the probability of an EMS incident occurring increases with the number of people gathered in one place [Channouf et al., 2007; Zhou, 2015; Steins et al., 2019; Huang et al., 2019; Matteson et al., 2011]. Because the weather influences what people do and where they are, the weather should, by extension, affect the spatial distribution of the EMS demand. However, this relationship between weather and EMS demand might be too weak to improve EMS demand forecasts.

1.4.4 Research Question 4

Online learning is a machine learning method commonly used when the data is constantly generated in time. When a machine learning model learns online, it is presented with a sample x_i at a time and makes its prediction $\hat{y}_i$ before knowing the correct output y_i. Then, the model learns from its error before being presented with the next sample x_{i+1}. The models continue learning like this one sample at a time without revisiting old samples. Online learning has, to the author's knowledge, not been used to forecast the EMS demand.

Research question 4 *Can online learning be used to improve EMS demand forecasts in Oslo and Akershus?*

Online learning makes a model capable of adapting to changes in the underlying function that the model is trying to estimate. Therefore, EMS demand forecasting models using online learning should perform better in the long run because populations change over time. For example, the population volume can increase or decrease over time as people reproduce more or less and people move to or from the area in question. The spatial density of the population can also change as the use of different areas changes. For example, suppose an old industrial complex is transformed into a large new office building. We could then see an increase in population density in this area, which again could affect the spatial distribution of the EMS demand. In addition, online learning leverages all of the available data, while offline learning only uses the data available at the start of the learning process.

1.5 Report Structure

The specifics of Oslo University Hospital (OUH)'s EMS system and dataset is described in Chapter 2. Chapter 3 details the underlying theory of the methods used in our experiments, while Chapter 4 reviews literature on EMS demand forecasting. In Chapter 5, we describe the scientific method used in our experiments. This chapter also provides implementation details of our models. The results of those experiments are presented in Chapter 6. Finally, in Chapter 7, we discuss our results and identify possible areas of future work.

Chapter 2

Background and Motivation

The Norwegian Ministry of Health and Care Services [2000] proposed response time goals for the Norwegian EMS; 90% of *acute* incidents should have a response time less than 12 minutes in densely populated areas and 25 minutes in sparsely populated areas. Further, 90% of *urgent* incidents should have a response time of less than 30 minutes in densely populated areas and 40 minutes in sparsely populated areas. As mentioned in Section 1.2, such performance indicators are somewhat arbitrary and not directly related to patient outcomes. However, the time goals were meant to serve as guidelines to give some idea about what the response times should be. Indeed, they were first proposed by Haga et al. [1998] because the lack of national guidelines had resulted in different practices across the country with varying standards and quality. Reaching those goals in Norwegian districts with few inhabitants, large distances, and varying weather and road conditions is impossible without a huge budget increase. Johansen et al. [2002] estimated that it would cost 224 million NOK yearly to fulfill the requirements proposed in Norwegian Ministry of Health and Care Services [2000] across Norway.

OUH has not been able to meet the response time goals and is looking to improve its response times by utilizing its limited resources more effectively. In particular, they are interested in dynamically distributing units based on factors such as time, weather, and historic caseload.

2.1 EMS at Oslo University Hospital

The Division of Prehospital Services at OUH is responsible for the prehospital critical care and transportation of patients in Oslo and what was previously Akershus and Østfold (now part of Viken). The Emergency Medical Communication Centre (EMCC) department answers calls to the emergency medical number and manages the ambulance fleet in Oslo, Akershus, and Østfold. This area is approximately 10,000 square kilometers and has around 1.5 million inhabitants. The medical operators at OUH are nurses or paramedics with additional training. They use a national triage system to classify a reported incident as acute, urgent, or regular. Regular incidents can be planned ahead of time, for example in the case of a patient transfer between two hospitals. A resource coordinator prioritizes between the EMCC's active missions and chooses which operative units to dispatch to which incident. The resource coordinator controls the fleet actively and can order a unit to relocate to cover temporarily unavailable units.

OUH's Ambulance Department is responsible for both emergency missions and regular patient transport in Oslo and Akershus. The department covers the area with 45 day units and 29 night units, staffed by ambulance workers or paramedics. In addition to the ambulances, the department has specialized resources such as physician-, paramedic- or supervisor-staffed vehicles,

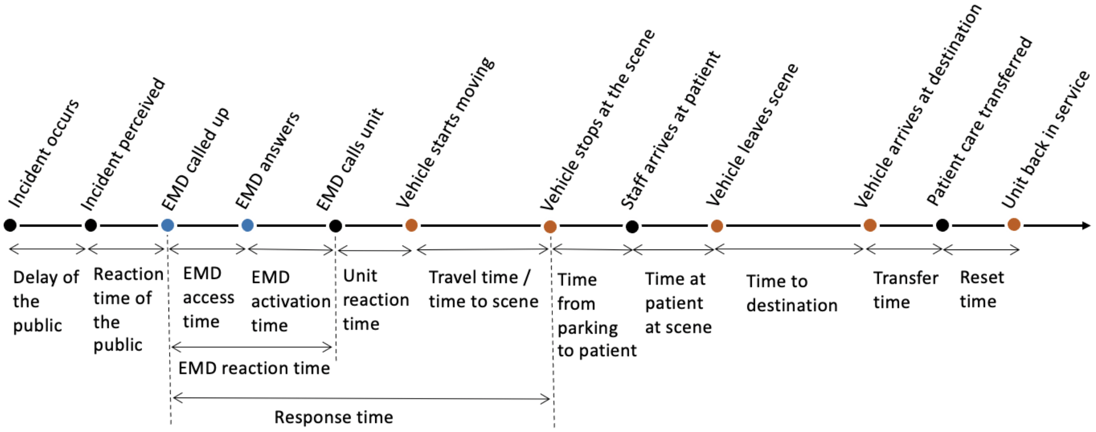


Figure 2.1: An EMS timeline with time points colored to indicate how they are tracked in OUH's systems. Red points are tracked manually by the ambulance personell, while blue points are tracked automatically in the EMCC's systems.

and rapid response units that utilize motorbikes, bicycles, and cars. These special resources generally do not have the capacity to transport patients. A newly introduced resource is the medical transport vehicles made for patients who need transport but no medical attention. Note that Østfold, although covered by the same EMCC as Oslo and Akershus, has its own ambulance fleet managed by Østfold Hospital.

Figure 2.1 illustrates the time points in the EMS timeline that are tracked in OUH's systems. The blue time points in the figure are tracked automatically in the EMCC systems, while the red time points are reported manually through a system in the ambulances.

OUH's ambulances are distributed over 15 ambulance stations. In 2013 a paramedic vehicle station was introduced at Nesodden to reduce the response time to this highly populated area. Between 2016 and 2017, the ambulance department gradually introduced strategic ambulance standby sites to further reduce the response time. These standby sites were predetermined strategic geographic locations near areas that had unsatisfactory response times. A standby site was typically a gas station parking lot near an intersection that could house one ambulance. The introduction of the Grorud standby site in 2016 resulted in a 20% increase in response time goal achievement in the nearby city boroughs Grorud, Alna and Stovner [Oftedahl, 2016]. No extra resources were introduced in this time period; only the distribution of resources was modified. These early standby sites decreased both the time-to-scene and the unit reaction time as the ambulance personnel waited inside the ambulances with the engines running, ready to leave at a moment's notice.

The introduction and use of the standby sites were somewhat problematic, as described in Kohlstrunk [2018]. There were issues around the working environment of the ambulance personnel, such as the lack of restroom and dining facilities. The number of standby sites peaked at a total of 10 in 2017. In 2019 the Norwegian labor inspection authority investigated the case and came to an agreement with OUH in which the use of standby sites without adequate facilities was to be terminated. In two standby sites (Grorud and Skedsmokorset), the necessary facilities were procured, and two other sites (Strand and Abildsø) had already been moved to locations with adequate facilities. Figure 2.2 shows the location of ambulance stations and active standby sites, as well as the standby sites that were closed in 2019.

Now, OUH is interested in knowing whether they can further reduce the response time by

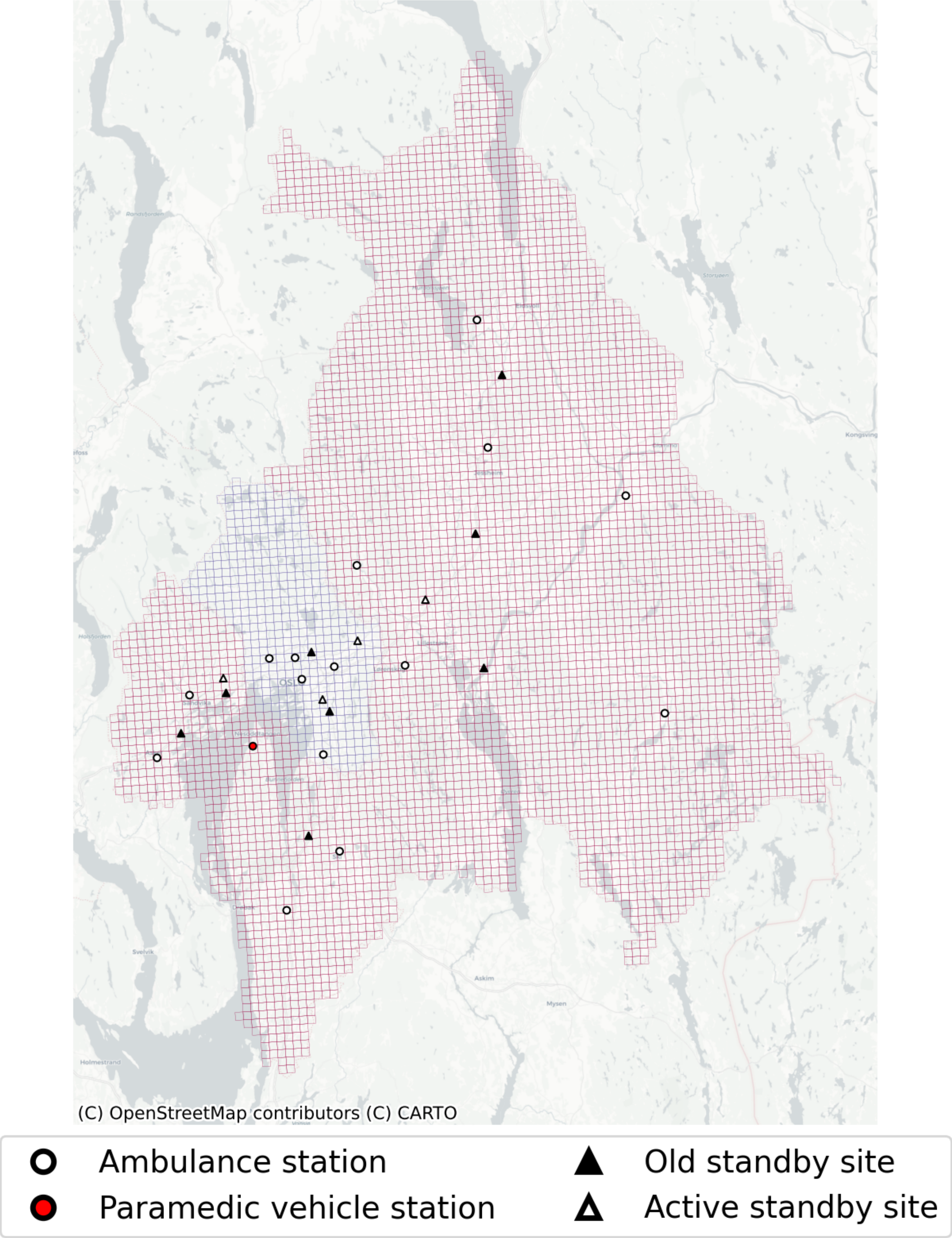


Figure 2.2: Illustration of the geographical areas of Oslo (blue grid) and Akershus (red grid) and the location of OUH's ambulance stations and standby sites. The dark areas are fjords, lakes, and rivers; the rest is land. The city center of Oslo is close to the Oslofjord, where the concentration of ambulance stations and standby sites is the highest. This area also has the highest population density.

positioning their ambulances strategically on a daily basis based on forecasts of the ambulance demand. They would like a complete system that makes daily recommendations for the number and location of ambulances for each hour of the following day. Such a system would need daily forecasts of the hourly ambulance demand in small geographical regions. The demand forecast can determine the number of ambulances needed and, together with a travel time and workload forecast, the optimal positioning of the ambulances. In this thesis, we focus on making such forecasts.

2.2 Datasets

The EMS *incident dataset* used in this thesis was provided by the EMCC department of OUH and the Norwegian National Advisory Unit for Prehospital Emergency Medicine (NAKOS). It includes the location and timestamps of missions completed by the ambulance department between January 1st, 2015, and February 11th, 2019. The timestamps recorded in the dataset are the red and blue time points depicted in Figure 2.1. The timestamps registered manually in the ambulances during a mission (red points in the figure) are often missing, especially for acute missions.

Because of privacy concerns, we only have access to an anonymized version of the data in which the exact incident locations have been mapped to a standard 1x1km grid as defined by Statistics Norway.[1] The grid map over Oslo and Akershus is illustrated in Figure 2.2.

Three main observations can be made from the statistics of the incident dataset presented in Table 2.1. Firstly, the number of incidents increases with each passing year. Secondly, there are more acute than urgent incidents and more urgent than regular incidents. Thirdly, that the ambulances are the most used units; they respond to 96% of the incidents in the dataset. In addition, we see that there are some incidents with unknown priority and some that occurred outside of the time range of the dataset. There are also a lot of incidents in the unfiltered incident dataset that lies outside of the geographical areas of Oslo and Akershus. These incidents are removed from the dataset before we use them for training forecasting models, as described in Section 5.1.1.

In addition to the incident dataset, we collect a *weather dataset* with precipitation and temperature data from the most central grid cell in Oslo at 3-hour intervals. The collection of this data is detailed in Section 5.1.2.

2.3 Initial Analysis of Dataset

In this section, we explore the *filtered incident dataset* to give ourselves and the reader an idea of how the EMS demand in Oslo and Akershus behaves in time and space. Refer to Section 5.1.1 for details on the filtered incident dataset.

There is a high degree of weekly seasonality in the EMS demand in Oslo and Akershus, as can be seen in Figure 2.3 and the autocorrelation plot in Figure 2.4. The demand is generally high during the day and low during the night. The demand is slightly shifted on weekends compared to weekdays as people sleep longer and stay up later. The planned regular incidents occur mostly in regular working hours, with almost no incident on evenings and few during weekends. From Figure 2.5 it seems like there is some annual seasonality as well. This becomes more apparent in the autocorrelation plot in Figure 2.6. Similar weekly and annual seasonality have been seen in

[1] More information about the grid can be found in Stand and Bloch [2009]. The grid can be downloaded from Statistics Norway at `www.ssb.no/natur-og-miljo/geodata`.

	Number of incidents
Total	754 811
In year: 2015	147 880
In year: 2016	185 976
In year: 2017	193 086
In year: 2018	201 675
In year: 2019	26 190
In year: other	4
Priority: acute	313 285
Priority: urgent	285 530
Priority: regular	155 987
Priority: unknown	9
Unit: ambulance	723 482
Unit: ambulance supervisor	19 718
Unit: physician-staffed vehicle	8 800
Unit: rapid response vehicle	2 251
Unit: medical transport	560

Table 2.1: Statistics of the original incident dataset.

multiple other case studies [Channouf et al., 2007; Steins et al., 2019; Zhou and Matteson, 2015; Matteson et al., 2011; Jones et al., 2002; Rezaei and Ingolfsson, 2021].

The EMS demand also exhibits an increasing trend, as can be seen in Figure 2.5, probably due to population growth or aging. A similar trend was present in Channouf et al. [2007].

There is a high degree of geographical locality of the demand, as can be seen in Figure 2.8. The most central grids have significantly more incidents than more rural areas. We can see from Figure 2.9 that the planned regular incidents constitute most of the incidents occurring in the cells with the highest EMS demand (the black cells in Figure 2.8). These high concentrations of regular events are connected to the transport of patients from hospitals.

Figure 2.7 illustrates the intense locality in a different way. The figure shows that over half of the grid cells in Oslo and Akershus have not had a single incident for over four years. Over the same period, only 581 of the 5569 grid cells have experienced 100 incidents or more.

In Figure 2.10 we illustrate the average distribution at different times of the day during weekdays. We are unable to see any apparent differences in the distributions, but it might seem like the day-distribution is *slightly* more spread out than the other two.

From Figure 2.9 and 2.3, it is clear that the planned regular incidents have a very different spatio-temporal distribution compared to the urgent and acute incidents, and to some lesser degree compared to the unplanned regular incidents. The planned regular incidents are concentrated temporally in regular working hours and spatially in grid cells with hospitals. The urgent and acute incidents have very similar spatial distributions. Their temporal distributions are also quite similar, but there are slightly more acute incidents. In addition, it seems like there are slightly more incidents during regular working hours relative to non-working hours for urgent and unplanned regular incidents. This pattern is not observed for acute incidents.

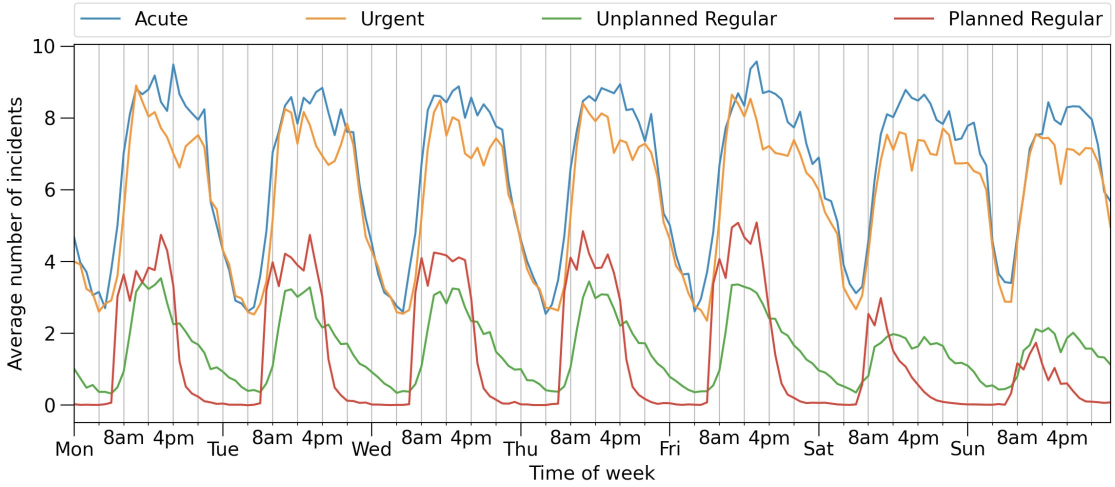


Figure 2.3: Average EMS demand per priority level for each hour of the week. The regular incidents are similar in volume, as are the acute and urgent incidents. The planned regular incidents have a distinct shape, while the volumes of the other priority incidents are more similar.

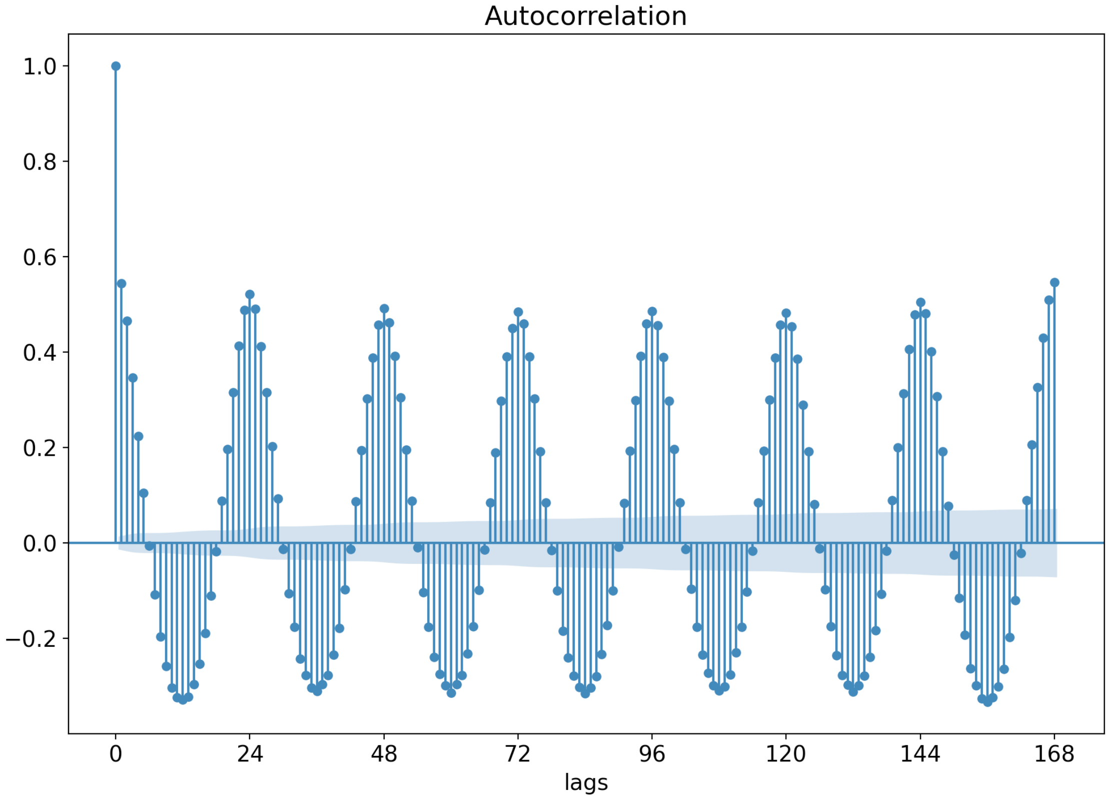


Figure 2.4: Autocorrelation plot of the hourly filtered demand, limited to one week. The highest autocorrelation is at one week lag.

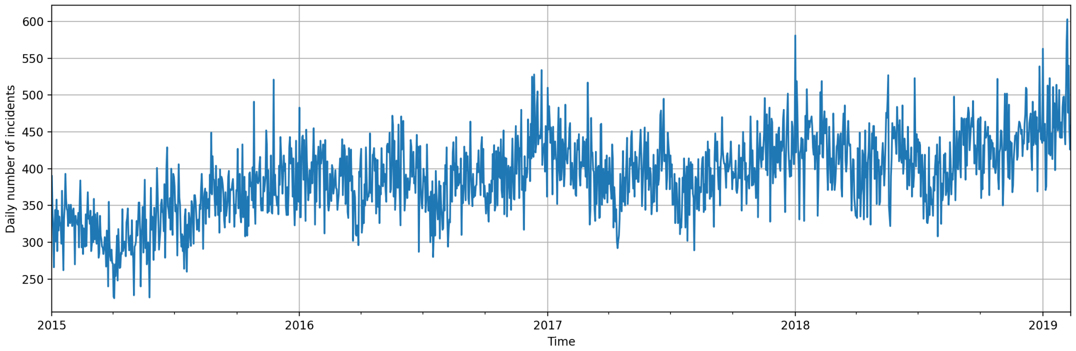


Figure 2.5: Number of incidents registered per day of the filtered incident dataset. The data appears to have an increasing trend and annual seasonality.

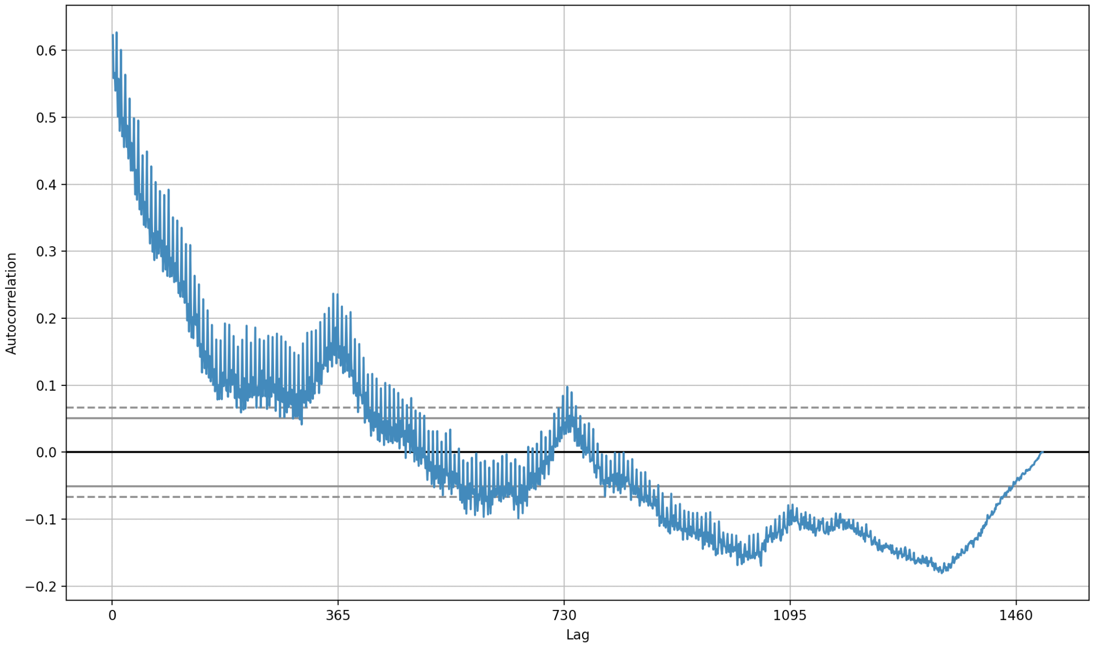


Figure 2.6: Autocorrelation plot of the daily demand of the filtered incident dataset. The plot indicates that the data has a trend and annual seasonality.

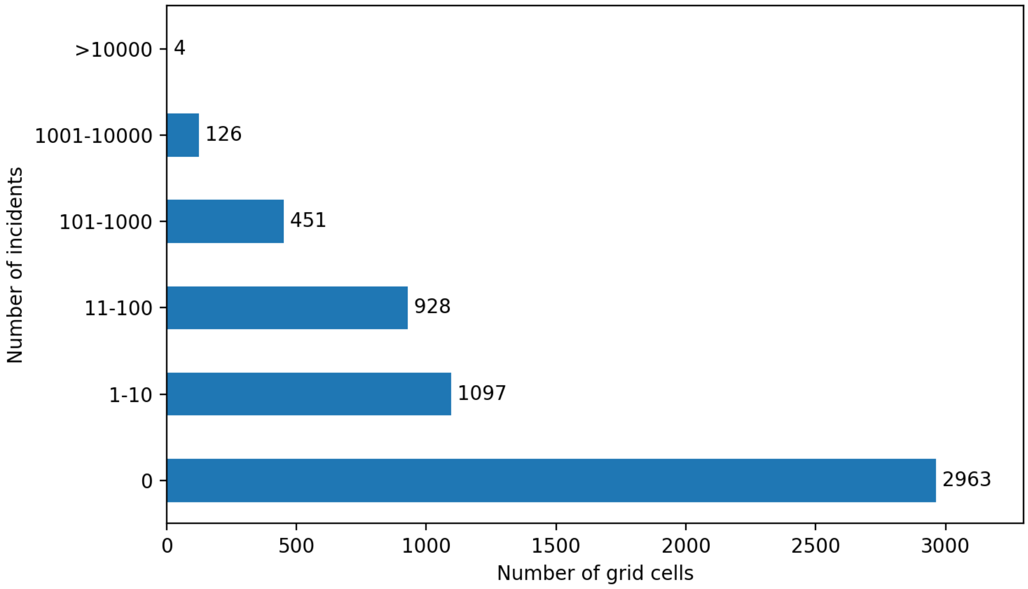
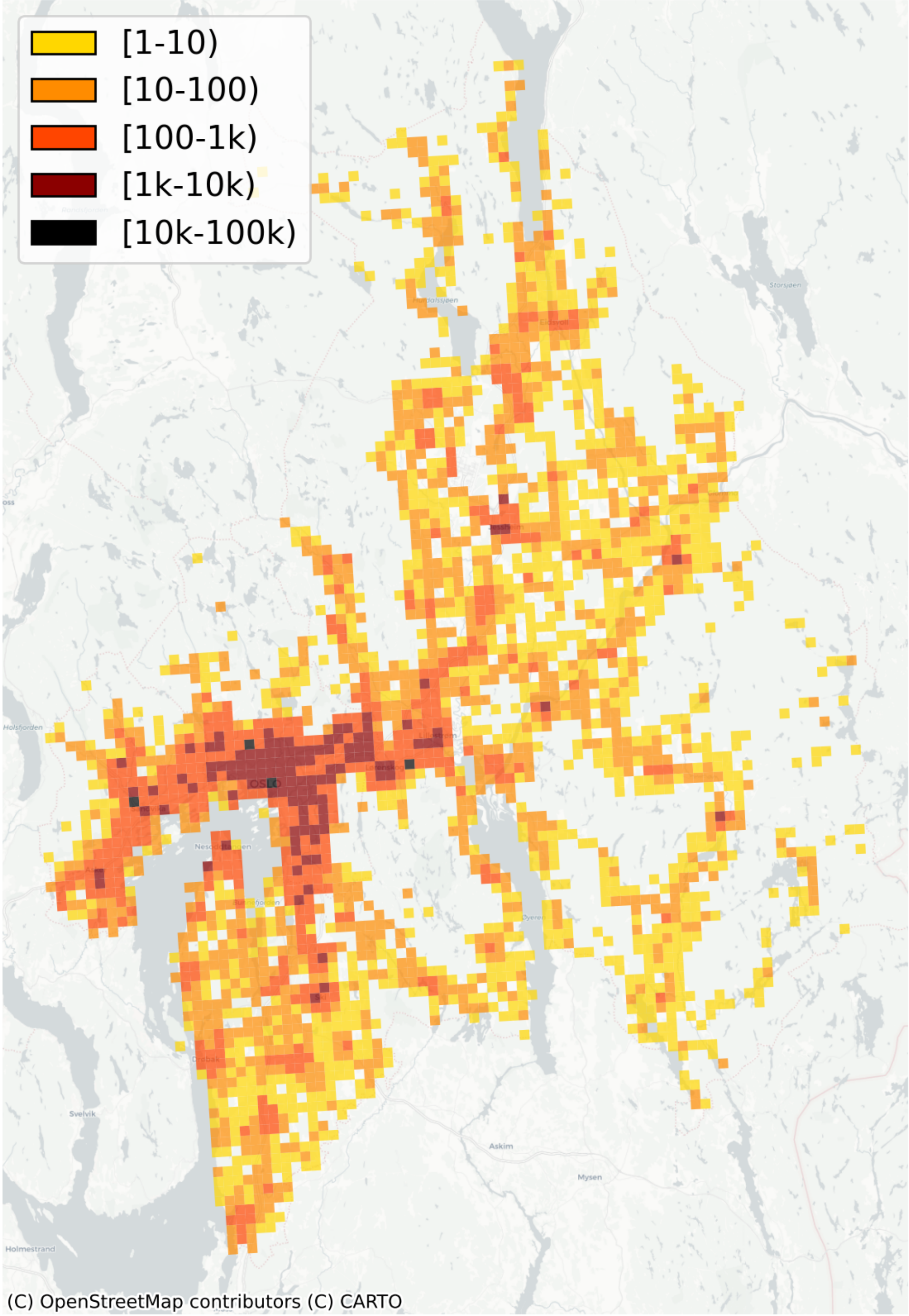
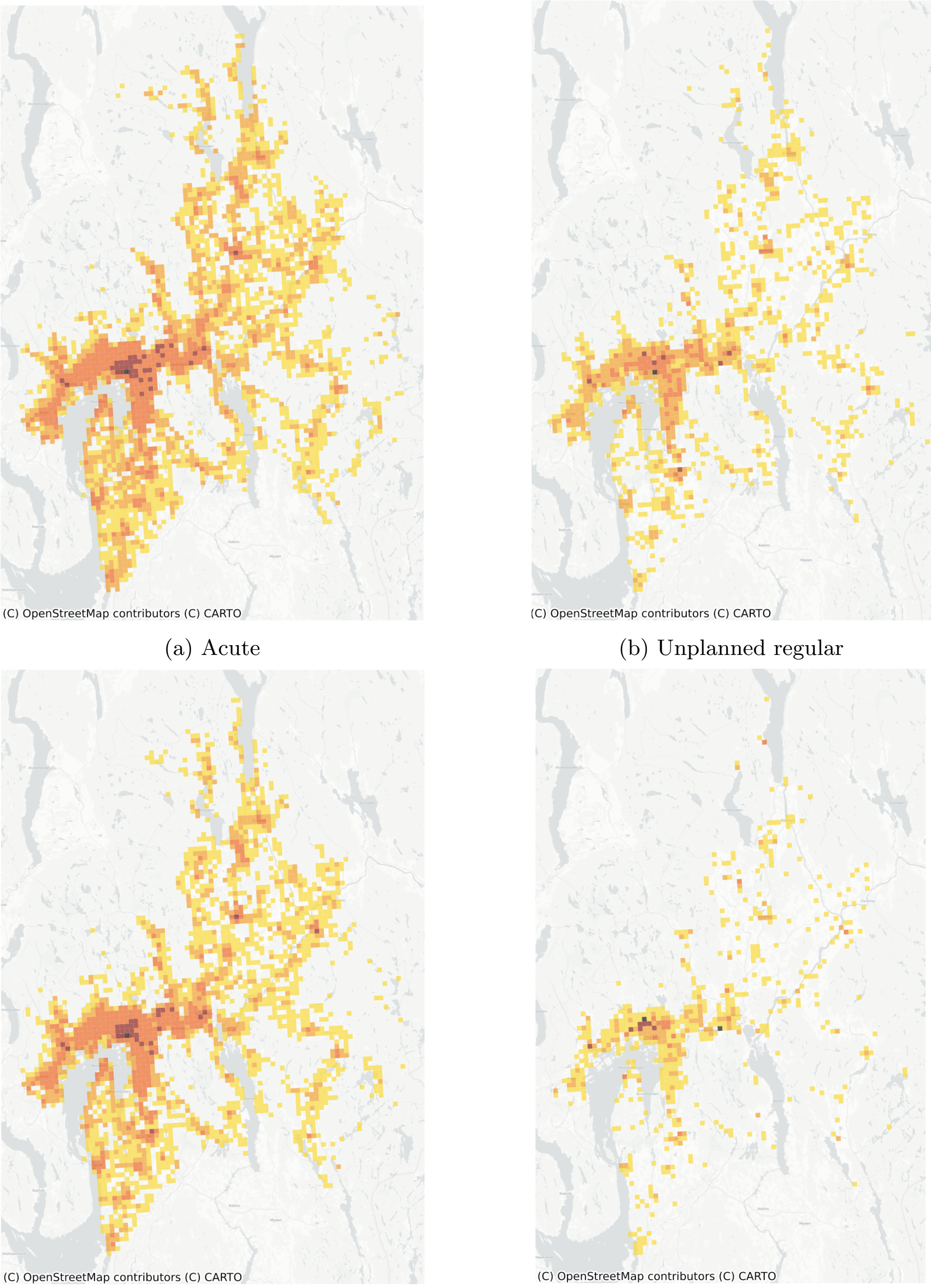


Figure 2.7: Distribution of grid cells in Oslo and Akershus on the total number of incidents they have in the filtered incident dataset. The data is skewed towards zero; 2963 of the 5569 grid cells have not experienced a single incidents over more than four years.



(C) OpenStreetMap contributors (C) CARTO

Figure 2.8: Illustration of the total number of incidents per grid cell in the filtered incident dataset. The demand exhibits extreme locality. See Figure 2.2 for a reference of the grid map of Oslo and Akershus.


(C) OpenStreetMap contributors (C) CARTO
(C) OpenStreetMap contributors (C) CARTO
(C) OpenStreetMap contributors (C) CARTO
(C) OpenStreetMap contributors (C) CARTO

(a) Acute

(b) Unplanned regular

(c) Urgent

(d) Planned regular

Figure 2.9: Illustration of the number of incidents per grid cell for each level of priority in the filtered incident dataset. The number of incidents indicated by the colors are of logarithmic scale, identical to the one showed in Figure 2.8

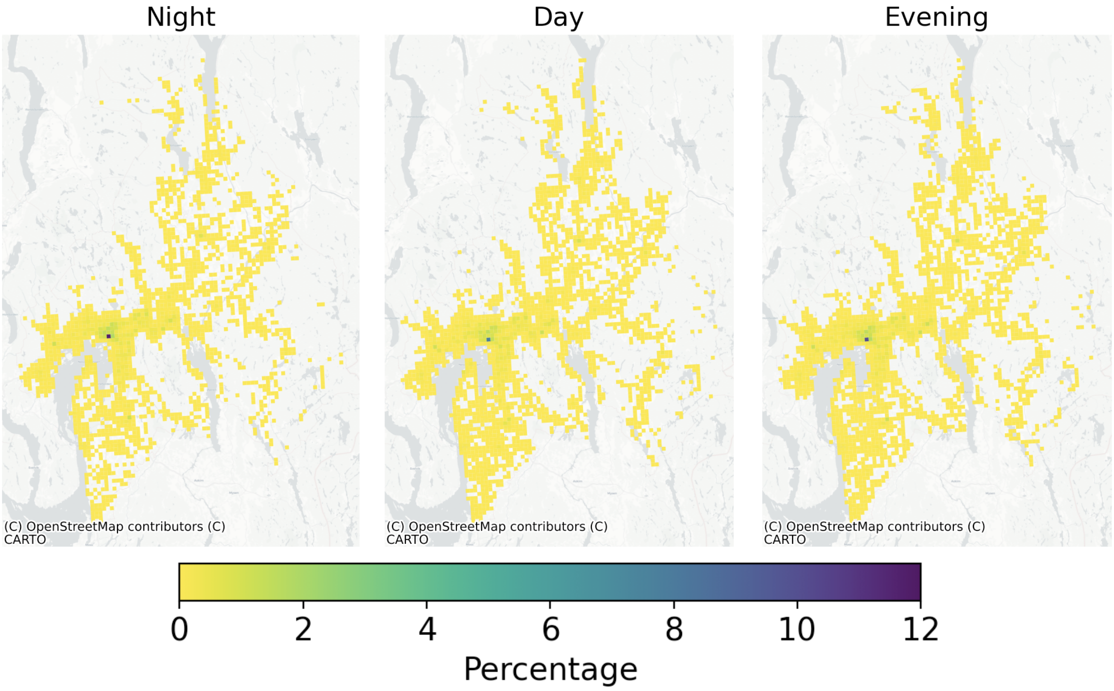

Night
Day
Evening
(C) OpenStreetMap contributors (C) CARTO
(C) OpenStreetMap contributors (C) CARTO
(C) OpenStreetMap contributors (C) CARTO
0
2
4
6
8
10
12
Percentage

Figure 2.10: The average distribution of weekday incidents at different time periods, excluding planned regular incidents. Night: 0-8, Day: 8-16, evening: 16-24. The distributions are strikingly similar.

Chapter 3

Theory

We propose a variety of models for predicting the EMS demand. This chapter is dedicated to explaining the fundamentals of time series and the models we propose for making forecasts of the EMS demand. We define time series and the simple moving average forecasting model in Section 3.1. Then we describe the fundamentals of neural networks and how they learn in Section 3.2. This section also details the architecture of the two types of neural networks used in this thesis. Finally, we define the metrics used to evaluate and compare the EMS demand forecasts of our proposed models in Section 3.3.

3.1 Time Series

A time series is a sequence of observations in time; $\mathbf{y} = [y_1, y_2, ..., y_n]$. Typically, the observations are made at regular periods of time, in which case the time interval between observations is the resolution of the time series. A time series can be univariate or multivariate. Univariate means that the time series consist of a single value, *i.e.* that $y_t \in \mathbb{R}$. An example of a univariate time series is the hourly number of EMS incidents in a spatial region. A multivariate time series on the other hand contains multiple variables in each time step, *i.e.* $y_t \in \mathbb{R}^m$ for some $m > 1$. An example of a multivariate time series is the hourly number of EMS incidents in three spatial regions, in which case $m = 3$.

3.1.1 Patterns

A time series can exhibit several common patterns. A *trend* in a time series means that the values are generally increasing or decreasing in the long term. EMS demand can exhibit an increasing trend if the area experiences an increase in population, as we saw in Figure 2.5. *Seasonality* in a time series means that there is a pattern in the series with a fixed period. We saw an example of weekly seasonality in Figure 2.3, where the EMS demand volume fluctuated regularly with the day and hour of the week.

3.1.2 Autocorrelation

Autocorrelation is the correlation between lagged values of a time series. There is one autocorrelation value for each level of lag k; r_k is the autocorrelation between y_t and y_{t-k}. The value of

r_k can be calculated according to Equation 3.1, where $\bar{y}$ is the average of the time series.

$$r_k = \frac{\sum_{t=k+1}^{n}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{n}(y_t - \bar{y})^2} \tag{3.1}$$

Plots of the autocorrelations of a time series can illustrate the potential trend and seasonality of the series. A trend will result in decreasing autocorrelation with increased lag, while seasonality will result in peaks in the autocorrelation at multiples of the seasonal frequency. The autocorrelation plot in Figure 2.4 showed that the hourly EMs demand volume had daily seasonality, but an even stronger weekly seasonality since $r_{168} > r_{24}$. Meanwhile, the autocorrelation plot in Figure 2.6 showed that the daily EMS volume had some trend and annual seasonality because we found peaks with each year and a generally decreasing autocorrelation.

3.1.3 Forecasting

Forecasting is about predicting the future as accurately as possible given all the information available, according to Hyndman and Athanasopoulos [2018]. The time horizon of a forecast specifies how many time steps into the future we want to predict. We denote the forecast of time series with horizon t as $\hat{y}_{n+t}$. Hence, the forecast of some hourly EMS demand two hours into the future will be $\hat{y}_{n+2}$. If we want to forecast every hour for the next 24 hours, we must make 24 separate forecasts with time horizons beginning at one and incrementally increasing up to 24.

We can make point forecasts or prediction interval forecasts. A point forecast $\hat{y}_{n+t}$ is expected future value of the relevant variable; $\hat{y}_{n+t} = \mathbb{E}[y_{n+t}]$. A prediction interval forecast $[\hat{l}_{n+t}, \hat{r}_{n+t}]$, on the other hand, is a range of values that contain the future value with some predetermined probability p such that $\mathrm{P}(\hat{l}_{n+t} < y_{n+t} < \hat{r}_{n+t}) = p$.

3.1.4 Moving Average

A moving average is a simple time series forecasting model that makes point forecasts by averaging a number of the previous values of the time series. There are several variations of the moving average model.

Simple Moving Average

A simple moving average (SMA) model uses the last k values to make forecasts. The forecast of an SMA$_k$ model for all time horizons $t > 1$ can be calculated with Equation 3.2.

$$\hat{y}_{n+t} = \frac{1}{k} \sum_{i=(n-k+1)}^{n} y_i \tag{3.2}$$

Cumulative Moving Average

A cumulative moving average (CMA) model uses all the available data when it forecasts. It is essentially a simple moving average where $k = n$. A CMA can be updated efficiently with a new sample y_{t+1}, as shown in Equation 3.3.

$$\hat{y}_{t+1} = \frac{y_{t+1} + t \cdot \hat{y}_t}{t+1}. \tag{3.3}$$

3.2 Artificial Neural Network

Artificial neural networks (ANNs) are learning systems inspired by biological brains. An artificial neural network consists of artificial neurons connected by directed links. A link from neuron a to neuron b serves to propagate the *activation* from neuron a to neuron b. Each link has a weight associated with it that determines the strength and sign of the connection. A neuron's activation is a function of its inputs. If the weighted sum of the inputs is above some (soft) threshold, the neuron "fires" by outputting a high activation value. Let w_{ij} be the weight of the link from node j to node i. Then the combination of the inputs and corresponding weights to node i is given by Equation 3.4.

$$z_i = \sum_j w_{ij} x_i + b_i. \tag{3.4}$$

Note that this is a simple linear function. A neural networks ability to capture non-linear functions comes from its *activation function.* A neuron's output or activation y_i is given by Equation 3.5, where σ is an activation function. [Russell and Norvig, 2009].

$$y_i = \sigma(z_i). \tag{3.5}$$

Neurons are often structured in *layers* in which the neurons in one layer are connected to the neurons of another layer. In this case, we can concisely represent the weights between the neurons of the two layers in a matrix $W \in \mathbb{R}^{mxn}$, where n and m are the numbers of neurons in the first and second layer, respectively. The biases of the neurons in a layer can be represented by a vector $\mathbf{b}$. Then we can express the output of the entire layer with matrix and vector operations as shown in Equation 3.6.

$$\mathbf{y} = \sigma(W\mathbf{x} + \mathbf{b}). \tag{3.6}$$

3.2.1 Supervised Learning

A neural network can be used to implement some function $\hat{\mathbf{y}} = f(\mathbf{x}; \Theta)$, where $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ are the values of the n input neurons, $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_m]^T$ are the values of the m output neurons of network after a forward pass, and Θ represents the weights and biases of the network. In a supervised learning setting, the desired output to a set of inputs are known, so we can compute the error of the network's outputs and use that to adjust the network's weights and biases. This can be formulated as an optimization problem. Let $L(\hat{\mathbf{y}}, \mathbf{y})$ be a loss function that measures the "distance" between an output $\hat{\mathbf{y}}$ and a target $\mathbf{y}$, and let D be the set of available input data. Then neural learning seeks to minimize the expected difference between the network's output and the target value by adjusting the network parameters, as described in Equation 3.7.

$$\min_\Theta E(\Theta) = \min_\Theta \mathbb{E}[L(f(\mathbf{x}; \Theta), \mathbf{y})]_{\mathbf{x} \in D} \tag{3.7}$$

The tuning of the network's weights is usually done through some variation of gradient descent learning - a simple optimization method. By calculating the gradient of the loss function ∇_Θ and taking a step in the opposite direction of the gradient, the function will be optimized to a local minimum. The backpropagation algorithm is used to efficiently propagate the error to the different weights and biases of the neural network.

Neural networks are usually implemented with some non-deterministic features. This causes different instances of a network to (usually) converge to different local minimums, resulting in the models having different performances. Examples of sources of randomness in neural networks are random initialization of weights and stochastic gradient descent optimization methods.

3.2.2 Online and Offline Learning

Supervised learning can be performed either offline or online.

In an offline learning setting, a training data set is collected and prepared in advance of the training. A machine learning model can train on the dataset for as long as it likes to extract as much information as possible from the data. When the model is finished with its training, it can make predictions, but it no longer learns; it will always produce the same output to the same input.

It is possible to retrain machine learning models offline periodically to leverage more data as it becomes available. This gives the offline models increasingly large datasets, which usually improves performance but can lead to storage and runtime issues. Each time the machine learning method is retrained, it estimates a single static function and weights all samples equally. Therefore, such a model will not be able to efficiently capture changes in the underlying function, even though it is presented with new samples.

In an online learning setting, the machine learning model does not have access to a complete training set with examples of inputs and outputs. Instead, it is presented with some inputs $\mathbf{x}_t$ at a time and has to predict the corresponding outputs $\hat{\mathbf{y}}_t$ before knowing the target values $\mathbf{y}_t$. When the target values are revealed, the model can calculate its errors and learn from them before being presented with the next sample $\mathbf{x}_{t+1}$. This is a natural setting for forecasting systems as they are made to make predictions without knowing the answer, but the answer is often known in hindsight. Online learning makes the model dynamic; the same input can result in different outputs at different times, even for a deterministic model. Sometimes, an online model is so eager to adapt to new information that it abruptly forgets previously learned information. This is know as the *catastrophic forgetting* problem [Losing et al., 2018].

It is possible to combine online and offline training. Typically, a data set is available that can be used for initial offline training of the model. Then, when the model is deployed in a real-world setting, it can continue learning using online learning.

3.2.3 Hyperparameter Tuning

Many parameters must be decided when implementing a neural network. Some choices, such as the activation function of the output layer and the number of output neurons, can be determined by the nature of the problem at hand. For example, if we try to solve a classification problem with ten potential classes, we would want to have ten output neurons and use the softmax activation function. Other parameters, such as the network architecture, are challenging to determine from the problem description. One of the best practices for making such decisions is to try out many different combinations using cross-validation and choose the one that works best.

We can evaluate the performance of a model without peeking on the test set by using a cross-validation method. A cross-validation method seeks to determine how well a model generalizes to an independent data set.

Holdout Cross-Validation

The simple holdout cross-validation technique splits the training data in two and uses one part for training and one part for validation. It is nontrivial to determine the sizes of the two sets. We want to use as much of the data as possible for training, but if the validation set is small, we will get a poor estimate of the model's accuracy. We denote the percentages used for the two sets as X/Y, where X is the percentage of the data in the training set, and Y is the percentage of the data used for the validation set. For example, 80/20 holdout means we use 80 percent of the data for training and 20 percent for validation.

K-Fold Cross-Validation

In k-fold cross-validation, we split the data into k subsets of equal size and use $k-1$ of them for training and the remaining subset for validation. We do this k times so that every subset is used for validation. This cross-validation technique utilizes more of the data to better estimate the accuracy at the cost of longer computation time.

3.2.4 Overfitting

Overfitting is a common problem in machine learning. It occurs when the model learns too much about the training data at the cost of generalization. This frequently happens when the model is complex and the amount of training data is relatively small. We can mitigate overfitting through different regularization techniques. *Early stopping* is a regularization technique based on monitoring the validation error and stop the learning process when the validation error stops improving. Sometimes, the validation error can go up temporarily before decreasing again. *Patience* can be used to avoid stopping the learning process prematurely. The patience number specifies how many learning iterations we allow with no improvement in validation error before we stop learning.

3.2.5 Classes of Artificial Neural Networks

Artificial neural networks come in many variations with different characteristics and capabilities. We can distinguish between different types of ANN models by looking at their topology. We describe the two types used in this study: the MLP and the LSTM.

MLP

A multilayered perceptron (MLP) is a fully connected feed-forward neural network structured in layers. An MLP has three or more layers: an input layer, one or more hidden layers, and an output layer. Each node in one layer is connected to every node in the next layer. The flow of information in an MLP is unidirectional; there are no loops in the connections of the neurons. An example of a simple MLP with one hidden layer is shown in Figure 3.1. According to the universal approximation theorem, an MLP can represent any continuous function within a specific range. We use MLPs to forecast the hourly volume, distribution, and complete EMS demand in Chapter 5.

LSTM

A Long Short-Term Memory (LSTM) is a recurrent neural network (RNN), meaning it connects its outputs to its inputs, allowing it to persist information over time. However, most RNNs struggle to persist information over extended periods because of vanishing or exploding gradients, as described in Bengio et al. [1994]. The LSTM was proposed in Hochreiter and Schmidhuber [1997] to combat these learning issues. Figure 3.2 shows the architecture of an LSTM. At time step t, the LSTM takes in inputs $\mathbf{x}_t \in \mathbb{R}^n$ and produces outputs $\mathbf{h}_t \in \mathbb{R}^m$. It uses its previous output $\mathbf{h}_{t-1}$ and the state of its memory cell $\mathbf{c}_{t-1}$ in addition to the new inputs to determine its next state and output. The squares in the figure represent neural network layers, while the

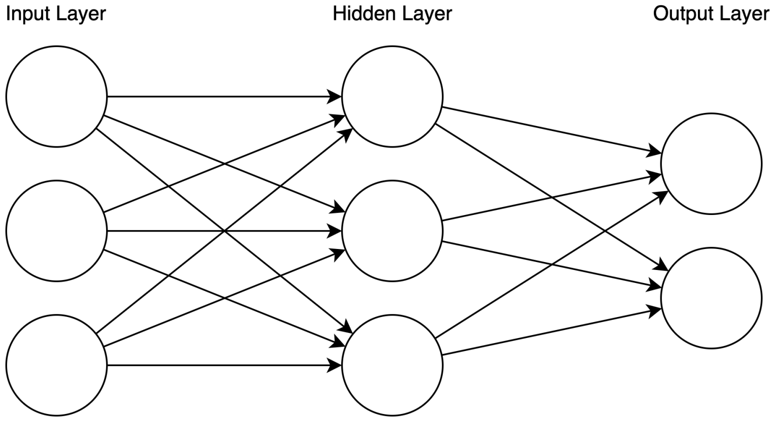


Figure 3.1: Topological illustration of an MLP with three input nodes, three nodes in the hidden layer and two output nodes.

circles represent element-wise operations. Each of these neural network layers has $m + n$ input nodes and m output nodes. The layer's activation function is either the sigmoid (σ) or tanh function, as illustrated in the figure. The memory cell $\mathbf{c}_t$ can persist through time which enables the LSTM network to capture long-term dependencies.

The LSTM has three specific "gates" for modifying its state. The *forget gate* determines which parts of the memory cell to remember. It is calculated according to Equation 3.8, where W_f and $\mathbf{b}_f$ are the weights and biases of the forget gate neural network layer, and $[\mathbf{x}, \mathbf{h}]$ denotes the concatenation of the two vectors.

$$\mathbf{f}_t = \sigma(W_f[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_f) \tag{3.8}$$

The *input gate layer* determines which parts of the new input are allowed to update the memory cell c. It is calculated according to Equation 3.9, where W_i and $\mathbf{b}_i$ are the weights and biases of the input gate layer.

$$\mathbf{i}_t = \sigma(W_i[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_i). \tag{3.9}$$

The value to (possibly) update the memory cell with is given by another neural network layer with a tanh activation function, as described in Equation 3.10.

$$\tilde{\mathbf{c}}_t = \tanh(W_c[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_c). \tag{3.10}$$

Together, the input gate and the forget gate determine how the memory cell is updated, as described by Equation 3.11.

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t. \tag{3.11}$$

Finally, the *output gate* determines which parts of the memory cell c_t to output. It is calculated similarly to the input and forget gate, as described in 3.12.

$$\mathbf{o}_t = \sigma(W_o[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_o). \tag{3.12}$$

The final output h_t of the LSTM is given by Equation 3.13.

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh \mathbf{c}_t. \tag{3.13}$$

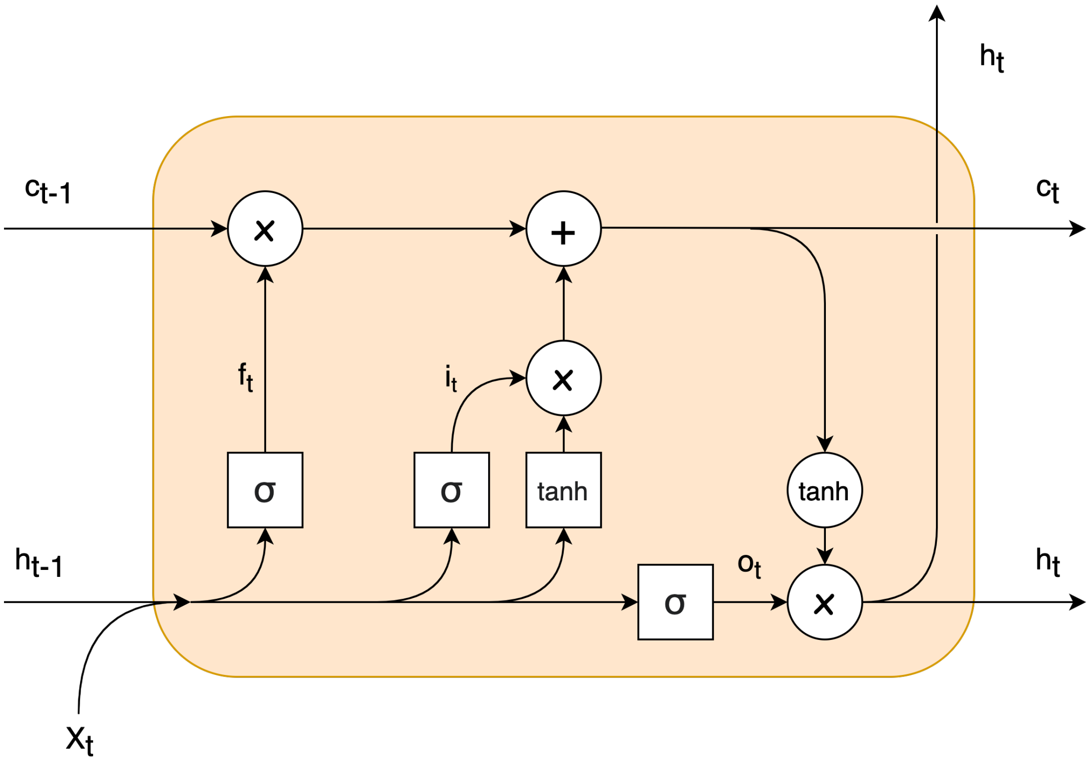


Figure 3.2: The architecture of an LSTM, adapted from Olah [2017].

The LSTM has been used with success as a time series forecasting model in many applications. We employ LSTM in our research to investigate whether there are patterns in time that can improve the forecasting of our EMS demand.

3.3 Error Metrics

Error metrics are used to quantitatively evaluate the performance of prediction models.

3.3.1 Mean Absolute Error

The mean absolute error (MAE) is a popular metric for regression problems. It avoids the cancellation of negative and positive errors by taking the absolute value of each error. For predictions $\hat{\mathbf{y}} \in \mathbb{R}^m$ and targets $\mathbf{y} \in \mathbb{R}^m$, the mean absolute error is defined as:

$$\mathrm{MAE}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = \frac{1}{m}\sum_{i=1}^{m}|\hat{y}_i - y_i|. \tag{3.14}$$

3.3.2 Mean Squared Error

The mean squared error (MSE) is also a popular metric for regression problems. Similarly to the MAE, it avoids cancellation of negative and positive errors. It squares the error instead of taking the absolute value, which makes it emphasize large errors. For predictions $\hat{\mathbf{y}} \in \mathbb{R}^m$ and targets $\mathbf{y} \in \mathbb{R}^m$, the mean squared error is defined as:

$$\mathrm{MSE}\left(\hat{\mathbf{y}}, \mathbf{y}\right) = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2. \tag{3.15}$$

.

MSE is often used as a loss function for neural networks because of the efficient calculation of its gradient.

3.3.3 Categorical Cross-Entropy

Categorical cross-entropy (CCE) is a measure of the distance between two probability distributions. It is based on cross entropy which is a measure of the distance between two vectors in information theory. For a prediction $\hat{\mathbf{y}} \in \mathbb{R}^m, \sum_{i=1}^m \hat{y}_i = 1$ and a target distribution $\mathbf{y} \in \mathbb{R}^m, \sum_{i=1}^m y_i = 1$, the categorical cross-entropy is calculated according to Equation 3.16.

$$\mathrm{CCE}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{i=1}^{m} y_i \log \hat{y}_i. \tag{3.16}$$

For many predictions and targets, we take the average of the cross-entropies between each prediction-target pair.

Chapter 4

Related Work

EMS demand forecasting is a crucial part of any ambulance location model. A variety of technical approaches have been proposed to forecast the EMS demand in previous research. One of the many choices an analyst must make when creating such a forecasting model is their forecasts' time and space granularity. Larger granularity makes the data less sparse, making it easier to train models. On the other hand, smaller granularity conveys more useful information for optimal unit location but is prone to being highly skewed towards zero demand. Forecasting the exact time and location of future events is impossible as the incidents are stochastic. This raises the question of how well we can estimate the demand and how much gain there is in trying to supersede rudimentary predictors.

In this chapter, we will discuss how others have modeled EMS demand in previous case studies. We group these cases by their spatial and temporal granularity, starting with the low resolution methods in Section 4.1 and ending with the high resolution methods in Section 4.4. Daily forecasts are considered a low temporal resolution, while a time interval of one or a few hours is considered as high resolution. For the spatial resolutions, we consider entire cities to be low resolution while spatial regions of a few square kilometers are considered high resolution. The work concerned with the highest spatio-temporal resolutions, discussed in Section 4.4, is given more attention as they are most relevant for this study.

4.1 Low Spatial and Temporal Resolution

The research detailed here study the daily EMS demand in large cities.

4.1.1 Wong et al.'s Research

Wong and Lai [2010] used regression analysis to determine the correlation between weather and daily ambulance demand in Hong Kong using EMS data from 2006-2009. The effects were studied for different target groups based on triage level, age, gender, and hospital admission status. They found a statically significant relation between the weather and the EMS demand among older people and people with pre-existing conditions. Further, they tested different amounts of time-lag to ascertain when the weather affects the EMS demand the most. They found that a time lag of four days produces the best results, meaning that the weather today affects the EMS demand in four days the most.

In Wong and Lai [2013], they show a significant relationship between weather forecasts and the daily ambulance demand using an ARIMA model. Their findings indicate that weather forecasts can improve demand forecasts, though to a lesser degree than historical weather data. Both Wong and Lai [2010] and Wong and Lai [2013] found the average temperature to be the most influential weather parameter.

Wong and Lin [2020] looked at the daily EMS demand of Taipei City, Taiwan, using EMS data from 2010-2012. They used multivariate regression with average, minimum, and maximum temperature, cloud amount, relative humidity, wind speed, barometric pressure, precipitation, and visibility inputs to model the demand. Also here, they found that a time lag of four days produced the best results and that older people and those with pre-existing conditions are most sensitive to the weather.

4.1.2 Thornes et al.'s Research

Thornes et al. [2014] study the effect of air temperatures on the daily EMS demand in Birmingham, UK, using data from 2007-2011. They found that the demand is affected by extremely hot and cold weather. Although they did not implement any forecasting models, they indicate that weather forecasting models should be used in EMS planning.

4.1.3 Huang et al.'s Research

Huang et al. [2019] use a Poisson Neural Network (PNN) to estimate the daily number of emergency calls in Ningbo, China. The result of the PNN is adjusted by fitting a multiple linear regression (MLR), an autoregressive integrated moving average (ARIMA), and a multi-gray model (GM) to the residual error of the PNN and adding this to the PNN's forecast. The model uses temporal features, weather, and incidents from the six previous days as inputs. The proposed combined model outperforms the other models tested, including an ANN, Poisson regression, GM, ARIMA, and MLR.

4.2 Low Spatial and High Temporal Resolution

The methods described in this section forecast the hourly demand in large cities, though the work described in Section 4.2.2 includes daily forecasts as well.

4.2.1 Matteson et al.'s Research

Matteson et al. [2011] combined a time series model with a dynamic latent factor model to forecast hourly emergency calls in Toronto, Canada. The factor model includes day-of-week and week-of-year effects via constraints on the factor loadings. Smoothing splines were used to enforcing a smooth evolution of factor loadings and levels. EMS data from 2007-2008 was used to train and validate the models. The authors found that the proposed model outperformed a moving average model.

4.2.2 Channouf et al.'s Research

Channouf et al. [2007] look at both daily and hourly forecasts of emergency calls in Calgary, Canada. They compare five different time series models for daily forecasts where three of the models build upon their predecessor. The basis for these improved models is a linear regression model with temporal inputs and special-day effects. The first extended model adds an autoregressive process for the errors to the basic model. The next extended model adds interaction between day-of-week and month-of-year of the first extended model. The third extended model is a stripped-down version of the second extended model. The last model is a doubly seasonal ARIMA model with special-day effects. They find that the third extended model (the stripped-down linear model with special-day effects, day-month interactions, and autoregressive residual estimation) makes the best daily forecasts for their data.

For hourly forecasts, the authors extend the best daily model in two ways: hourly conditional distribution of the daily call volume and an autoregressive hour-of-day effect. The first of these two models produce the best forecasts. In addition to this, the authors show that it is possible to improve the hourly forecast with a kind of online learning approach in which they update the hourly forecast for the later hours of a day with correct data for the early parts of the day.

4.3 High Spatial and Low Temporal Resolution

This section describes two studies that forecast the daily demand in small spatial regions of some square kilometers or on a continuous spatial domain.

4.3.1 Lin et al.'s Research

In Lin et al. [2020], six different models are trained on nine years worth of data to forecast the daily EMS demand in each of Singapore's regions. The six models tested are moving average, linear regression, support vector regression (SVR), MLP, radial basis function network (RBFN), and light gradient boosting machine (LightGBM). Among these, the LightGBM was chosen as the most appropriate model. The LightGMB model was then trained on two other datasets, one with more demographic data and one with composite spatio-temporal features. They found that the additional data did not improve the forecasts; the composite features made the forecasts worse.

4.3.2 Grekousis and Liu's Research

Grekousis and Liu [2019] try to forecast the exact location of EMS events on a weekly basis in Athens, Greece. They use a novel notion of "poly events" in which they match incidents in successive time steps (weeks) to form paths in time. To make forecasts based on the poly events, they use an artificial neural network (ANN) which is tuned with evolutionary hyper-parameter optimization. They use 23 weeks of data to calibrate the model and test it by forecasting the 24th week. They find that they reduce the total distance between ambulances and incidents by over 1km when distributing the units according to a location-allocation model with their forecast demand as input, compared to the current location of ambulances in Athens.

4.4 High Spatial and Temporal Resolution

This Section is focused on research done on hourly forecasts in either small spatial regions of some square kilometers or on a continuous spatial domain. Using such high spatio-temporal scales is one of the main challenges of our work; therefore, this research is reviewed in greater detail than previous sections.

4.4.1 Steins et al.'s Research

Steins et al. [2019] use Zero-Inflated Poisson (ZIP) regression to model the hourly EMS call volume in 6x6km, 4x4km, and 3x3km spatial regions of three different Swedish counties. The

ZIP regression combines two models: one that generates zeroes and a Poisson process that generates counts. This makes it suitable for modeling point processes with excess zero-count data. Steins et al. [2019] create one model for each county, with socioeconomic, geographic, and temporal features as independent factors. The ZIP model performs slightly better than the existing model, which calculates the total average demand in the county per hour of the week using data from the previous year and then assigning a fraction of this demand to each grid based on their day and night population.

4.4.2 Setzler et al.'s Research

In Setzler et al. [2009], the authors propose an MLP to forecast the emergency call volumes in Charlotte-Mecklenburg, North Carolina (USA) for a variety of grid cell sizes and time intervals. The provided EMS data from 2002-2004 was mapped to different combinations of grid cell sizes (2x2mile and 4x4mile) and time intervals (1hr and 3hrs). The MLP has four temporal inputs (hour, day of week, month, and season), a hidden layer with four nodes, and one output value for each grid cell. Setzler et al. compared their proposed MLP model to the *MEDIC* forecasting method used by the EMS agency in Charlotte-Mecklenburg.

The MEDIC method is a simple moving average model, which makes forecasts by averaging the call volumes at the same time point for the previous four weeks over the past five years. Let $A_{h,d,w,y}$ be the actual call number for a given hour, day, week, and year. Let $Y = 5$ be the number of years and $W = 4$ be the number of weeks to average over. Then the MEDIC forecast $F_{h,d,w,y}$ for a given hour, day, week, and year is given by Equation 4.1.

$$F_{h,d,w,y} = \frac{\sum_{i=1}^{Y} \sum_{j=1}^{W} A_{h,d,w-j,y-(i-1)}}{YW} \tag{4.1}$$

The authors found that the new model performs slightly better than MEDIC for 4x4mile grids (both 1hr and 3hr time intervals). For the 2x2mile grid with 3hr intervals configuration, there is no statistical difference between the two methods. Interestingly, MEDIC was better for the 2x2mile grid with 1hr time intervals. Upon further inspection, the authors found that forecasting only zeros outperformed the two other approaches for this configuration.

4.4.3 Chen et al.'s Research

Chen and Lu have studied the EMS demand in 3x3km spatial regions in New Taipei City, Taiwan using data from 2010 to 2012. Chen and Lu [2014] forecast the daily demand using moving average, neural network, linear regression, and support vector machine models. They train one model for each of the spatial regions for each of the four model types. The SVM and ANN models are trained with month, day of the week, season, hour, and year inputs. They retrain their models with increased data after each forecast they make on the test set. They found that the ANN model was best overall but that the SVM models were better for regions with high EMS demand. They propose a model selection phase in which the best of the four proposed models is selected for each region.

In the follow-up study, Chen et al. [2016] adapt this two-phased model-selection approach to make 3hr forecasts in 3x3km spatial regions of three districts in New Taipei City. In the first phase, they use cross-validation to select the best input parameters and potential hyperparameters for the four different models. Then, in the second phase, the best of the four proposed models is selected for each grid. The authors tested seven different input sets for the SVM and ANN models. The features included in the input sets are described in Table 4.1. Many different models with different input sets were chosen for the regions. The district with the highest EMS

Input set	**Input Features**
Input set 1	Month, day, day of week, time bucket, weekend, rush hour, year
Input set 2	Month, day, day of week, time bucket, weekend, rush hour
Input set 3	EMS demand of previous time bucket, weekend
Input set 4	EMS demand of previous time bucket, rush hour
Input set 5	EMS demand of previous time bucket, weekend, rush hour
Input set 6	EMS demand of previous time bucket, weekend, rush hour, year, month, day of week
Input set 7	Season, month, day of week, time bucket

Table 4.1: The input sets tested in Chen et al. [2016], adapted from the original paper. Note that input set 7 is identical to the one used in Setzler et al. [2009].

demand chose the ANN model in 4 out of 7 regions, while the districts with low demand preferred the linear regression models, choosing those in 16 out of 27 regions. All of the input sets were chosen one to three times each.

The paper also includes a smaller study on forecasting the daily demand in 2x2km regions in Banqiao. Daily rainfall data was included in the inputs of the ANN and SVR models, but the initial input set used is unspecified in the paper. During the validation phase for this configuration, the following models were chosen: 5 ANN models, 5 SVR models, and 4 MA models, where 4 of the SVR and 2 of the ANN models used the additional rainfall data.

In the 2016 study, Chen et al. found that the two-fold model selection approach produced better results than the single model approach of Chen and Lu [2014].

4.4.4 Zhou's Research

In her PhD thesis [Zhou, 2015] and related publications [Zhou et al., 2015; Zhou and Matteson, 2015, 2016; Zhou, 2016], Zhou proposes several novel approaches for forecasting the ambulance demand on a discrete time and continuous space domain. She assumes that the ambulance demand in each 1- or 2-hr time period independently follows a non-homogeneous Poisson process whose expected value can be represented as a positive intensity function $\lambda_t(s)$, where $s \in S \subseteq \mathbb{R}^2$ is a spatial location in the relevant geographic area S. The intensity function can be decomposed into an aggregate demand intensity $\delta_t = \int_s \lambda_t(s)\,\mathrm{d}s$ and spatial density $f_t(s)$ such that $\int_S f_t(s)\,\mathrm{d}s = 1$. Then, $\lambda_t(s) = \delta_t f_t(s)$. The aggregate demand intensity δ_t is simply the total demand of the entire geographic area of interest in time interval t, which has been well studied by others. Hence, Zhou focuses on forecasting the spatial density $f_t(s)$. We call this approach of forecasting the aggregated demand and its spatial density separately as a “split” approach.

Zhou [2015] includes case studies from Toronto, Canada, and Melbourne, Australia, with data from 2007-2008 and 2011-2012, respectively. The Toronto study is described in Zhou and Matteson [2015] and Zhou et al. [2015], while Zhou and Matteson [2016] describes the Melbourne case, and Zhou [2016] contains a summary and comparison of the methods of both case studies. Three main models are proposed for modeling the spatial density $f_t(s)$:

1. A Gaussian Mixture Model (GMM) with weekly seasonality constraints and conditional autoregressive (CAR) priors to capture daily seasonality.

2. A spatio-temporal kernel density estimation (stKDE) approach.

3. An extended stKDE with kernel warping.

These methods are compared against each other and the benchmark MEDIC approach adapted from Setzler et al. [2009] using the Average Logarithmic Score (ALS).

The stKDEs are found to produce equally accurate forecasts as the GMM while having considerably lower time complexity. The warped stKDE model outperforms the other models in the Melbourne case as the warping adapts the stKDE to Melbourne's complex geographical boundaries.

4.5 Discussion

We are inspired by the work of Zhou and want to investigate how her split approach performs compared to the more common complete method. We are also inspired by Setzler et al. [2009]'s use of neural networks. We believe neural networks have a lot of potential in modeling the EMS demand, and we will make an effort to find the best possible architecture for such a model. In addition, we test the LSTM model, which, to our knowledge, has not been tested in this domain before. We use the MEDIC benchmark and the neural network model proposed in Setzler et al. [2009] as benchmarks. Channouf et al. [2007] showed that the use of special day effect flags for New Year's Eve and a local festival improved demand volume forecasts. We try to capture such special days implicitly by adding a day of the month variable to our input set. Combined with the month of the year, this should make it possible for a neural network to capture relevant special day effects. We also use weather inputs as suggested by Setzler et al. [2009] and Zhou [2015]. Our study in Chapters 5, 6, and 7 is an extension of the work done by Wong and Lai as we use a much higher spatio-temporal resolution. We also employ an online learning regime as suggested in Chen and Lu [2014] and Chen et al. [2016] and compare it to traditional batch learning.

Chapter 5

Experimental Method

We want to make accurate hourly forecasts of the EMS demand in 1x1km spatial regions in Oslo and Akershus. Such a forecasting model must be able to produce forecasts for the next 24 hours in a reasonable time.

We propose and implement a variety of forecasting models and compare them against each other and the MEDIC, MLP, and All 0s methods discussed in Setzler et al. [2009].

Section 5.1 details how we preprocess the incident dataset and transform it to a time series, how we collect weather data for our weather dataset, and what the different input sets for our neural network models contain.

We explain how we divide our proposed models into three main categories (distribution, volume, and complete) in Section 5.2. In Subsection 5.3, we describe how we structure our experiments in three main phases and how the different models are trained and compared against each other in each phase.

Section 5.4 describes the implementation details of our proposed and benchmark offline methods. Finally, in Section 5.5 we detail how we augment our proposed models to use online learning.

All the programs created in this thesis are run on a 2016 MacBook Pro with a 2.9GHz Intel Core i5-6267U processor and 8GB memory. We have not comprehensively timed the training or testing of the different models as none of the models have had any issues producing 24 forecasts in a matter of seconds.

5.1 Data Collection and Preprocessing

Before training our models, we preprocess the dataset. This includes removing incorrect and uninteresting data and structuring it as a time series. We also collect weather data to be used as inputs to our proposed neural networks.

5.1.1 Preprocessing

The original incident dataset described in Section 2.2 contained many missions outside of the area officially covered by the ambulance department of OUH, most prominently in Østfold because OUH's EMCC department covers this area. These incidents were filtered out by doing an inner merge on the id of the incident grid with those of Oslo and Akershus (collected from SSB), reducing the dataset by 169 436 incidents. A total of 27 201 duplicate rows were also removed. In addition, nine incidents with invalid priority levels and four with invalid time points were filtered out. Some statistics on the resulting *filtered incident dataset* can be found in Table

5.1. We see that the filtered dataset has the same characteristics as the unfiltered dataset, as described in Section 2.2.

	Number of incidents
Total	558 161
In year: 2015	118 385
In year: 2016	135 751
In year: 2017	139 081
In year: 2018	146 424
In year: 2019	18 520
Priority: acute	237 732
Priority: urgent	213 520
Priority: unplanned regular	56 505
Priority: planned regular	50 404
Unit: ambulance	532 925
Unit: ambulance supervisor	17 179
Unit: physician-staffed vehicle	7 435
Unit: rapid response vehicle	93
Unit: medical transport	529

Table 5.1: Statistics of the filtered incident dataset.

To create the *incident time series* to be used in model fitting and testing, we first remove the planned regular events from the filtered incident dataset. We do not need to forecast the planned regular incidents because they will be known in advance. Then, we sort the remaining incidents by grid cell id and resample them into hourly intervals starting at 00:00 on the 1st of January 2015 up to and including 23:00 on the 11th of February 2019. The result is a multivariate time series of hourly resolution with $N = 5569$ variables - one for each spatial region. We denote this time series as $\mathbf{y}$, such that $\mathbf{y}_t \in \mathbb{R}^M$ is a vector with the incidents in the $M = 5569$ regions at time step t, and $(y_t)_k \in \mathbb{R}$ is the number of incidents in region k at time step t.

5.1.2 Weather Data Collection

Weather data has been collected through the Grid Time Series Data API provided by The Norwegian Water Resources and Energy Directorate (NVE).[1]. The API provides access to time series of weather data for grid cells. Several time series are available, such as daily temperature, precipitation, snow depth, and wind speed. We collected temperature and precipitation data at 3-hour intervals from 01.01.2015 00:00 up to and including 11.02.2019 21:00 for the grid cell with the most incidents (SSB id 22620006649000). We then normalize the temperature and precipitation data using Kera's MinMaxScaler fitted on the data belonging to the training set.

5.1.3 Input Sets

The neural networks are trained and tested with four different sets of inputs. The *basic* input set includes the hour of the day, day of the week, and month. These features are all one-hot encoded, making a total of $24 + 7 + 12 = 43$ input neurons. The other input sets are extensions of this basic set.

[1] The API is available at `http://api.nve.no/doc/gridtimeseries-data-gts` There is also an interactive application built on top of the API, which can be used to explore the data more intuitively at `http://www.xgeo.no/`.

Input set name	Identifier	Features	Nodes
Basic	b	Hour, day of week, month	43
Weather	w	Hour, day of week, month, precipitation, temperature	59
Date	d	Hour, day of week, month, day	74
Weather and date	w&d	Hour, day of week, month, day, precipitation, temperature	90

Table 5.2: Overview of the four different input sets. The identifier of the input set is used in the tables in the results (Chapter 6). The nodes column states the number of input nodes each input set has.

The *weather* input set includes rainfall and temperature measurements in addition to the basic inputs. For each hourly forecast, the entire day's worth of weather data is included. As mentioned in Section 5.1.2, the weather data is collected at 3-hour intervals, meaning each day has eight measurements of each weather type. Hence, the weather set has a total of $43+8\cdot2=59$ input nodes.

The *date* input set includes the day of the month, in addition to the basic set. The day of the month is one-hot encoded with 31 classes, making a total of $43+31=74$ input nodes. We hope that the inclusion of the day of the month will allow the neural network to leverage special day effects.

We also test a combined *weather and date* input set, which has $43+8\cdot2+31=90$ input nodes.

An overview of the inputs and their identifiers used in the result tables can be found in Table 5.2.

5.2 Forecasting Model Categories

Our proposed models are split into three main categories: distribution, volume, and complete models. The volume models forecast the aggregated hourly demand, while the distribution models forecast how this demand is distributed spatially across the grid cells. The forecasts of the volume and distribution models can be combined into a complete forecast which predicts the number of incidents in each grid cell. The complete models make such complete forecasts directly. We evaluate the volume and complete models using MSE and MAE, while the distribution models are evaluated using categorical cross-entropy.

The complete models have 5569 positive output values - one for each grid cell. The target values for the complete models are simply the number of incidents in each of the M regions at each time step t; $\mathbf{y}_t \in \mathbb{R}^M$.

The volume models have a single positive output. The target values for the volume models are the sum of all events across all grid cells in a time step; ${y_t}^{vol} = \delta_t = \sum_{i=1}^{M} (y_t)_i$.

The distribution models have $M = 5569$ output values - one for each grid cell. The output values must be positive and sum to one. The distribution target values are calculated by normalizing the number of incidents across a time step; ${\mathbf{y}_t}^{dist} = \frac{1}{\delta_t}\mathbf{y}_t$. If there are no incidents ($\delta_t = 0$) in a time step, an even distribution is used as the target; ${\mathbf{y}_t}^{dist} = [\frac{1}{M}, \frac{1}{M}, ..., \frac{1}{M}]$.

For example, suppose we have only 5 grid cells with the following number of incidents at time step t: $\mathbf{y}_t = [1, 2, 0, 0, 1]$. Then, the target distribution will be: ${\mathbf{y}_t}^{dist} = [0.25, 0.5, 0, 0, 0.25]$, the target volume will be ${y_t}^{vol} = 4$, and the complete target will be ${\mathbf{y}_t}^{comp} = [1, 2, 0, 0, 1]$.

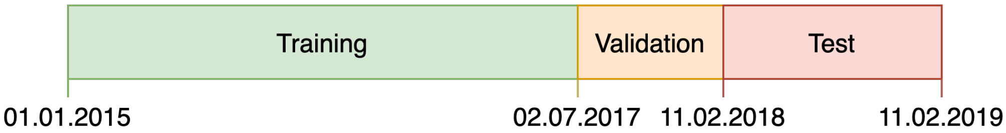


Figure 5.1: Illustration of how the data set is split into a training, validation and test set.

Next, suppose that for a different time step $t + k$ there are no incidents: $\mathbf{y}_{t+k} = [0, 0, 0, 0, 0]$. Then the target distribution will be $\mathbf{y}_{t+k}{}^{dist} = [0.2, 0.2, 0.2, 0.2, 0.2]$, the target volume is $y_{t+k}{}^{vol} = 0$, and the complete target will be $\mathbf{y}_{t+1}{}^{comp} = [0, 0, 0, 0, 0]$. Note that $\forall t\ (y_t{}^{vol} \cdot \mathbf{y}_t{}^{dist} = \mathbf{y}_t{}^{comp} = \mathbf{y}_t)$.

5.3 Experimental Phases

The forecasting models are trained, validated, and tested in different phases with different data. Common for all phases is the use of early stopping with patience 5 in the training of the neural network models. The maximum number of iterations is set to 100, but this limit is never reached.

First, in the *architecture selection phase*, we select the best architecture for each of our proposed neural networks by using cross-validation on the training set. We use 5-fold cross-validation for the MLP models and hold-out 80/20 hold-out cross-validation for the LSTM models, as they are sensitive to the sequential order of the time series. We test 14 different architectures for each model, varying between 2 and 3 hidden layers with 2, 4, 8, 16, 32, 64, or 128 nodes in each layer.

Next, in the *validation phase*, we train each of the proposed models on the training set and evaluate them on the validation set, with and without online training. Because neural networks have random initialization, we create five instances of each neural network model and train them 80/20 hold-out cross-validation error. The instance with the lowest cross-validation error is selected for the model, and only this instance is evaluated on the validation set. We select the two best models within each category to proceed to the testing phase based on their performance on the validation set.

Finally, in the *testing phase*, the six models selected in the validation phase and the three benchmark models (MEDIC, MLP, and All 0s from Setzler et al. [2009]) are trained anew on the combined training and validation set and tested on the test set, producing our final results. Also here, each neural network model is trained five times with 80/20 hold-out cross-validation, and only the instance with the best cross-validation error is evaluated on the test set.

Figure 5.1 shows how the incident time series is split into the training, validation, and test set. The *training set* is the largest, with just over 2.5 years of data (1st of January 2015 - 1st of July 2017). The *validation set* is the smallest, with just over 0.5 years of data (2nd of July 2017 - 10th of February 2018). The *test set* consists of 1 year's worth of data (11th of February 2018 - 11th of February 2019).

5.4 Offline Forecasting Methods

We implement a variety of offline forecasting methods for forecasting the hourly EMS demand in Oslo and Akershus. These include several neural network models and some simple baseline models based on averages of the historical EMS demand. The implementations of these models are detailed in Section 5.4.1. We also implement 3 benchmark models adapted from Setzler et al. [2009], as described in Section 5.4.2

The models were implemented in Python 3.6.7 with the Keras 2.4.3, Pandas 1.1.2, GeoPandas 0.8.1, and Numpy 1.19.2 libraries. Default values have been used unless otherwise specified.

5.4.1 Proposed models

We choose neural networks because of their ability to identify and leverage complex and non-linear patterns in data. The LSTM model can also leverage the sequential nature of the problem and may be able to pick up patterns over time. The baseline models are created to capture the simple patterns in time and space we saw in Section 2.3.

Neural Networks

We propose both MLP and LSTM neural networks (discussed in Chapter 3) to forecast EMS demand. We use Keras to implement a total of 24 neural different network models: one MLP and LSTM model for each of the four input sets for each of the three categories. As mentioned in Section 5.3, we test each neural network model with 2-3 hidden layers and 2, 4, 8, 16, 32, 64, and 128 nodes in each layer during the architecture selection phase.

We the RMSProp optimizer for all of the ANNs and the ReLU activation function in the hidden layers of the MLPs. For the LSTM volume and complete models, we normalize the output values of the training set using Keras' MinMaxScaler to ensure efficient training.

The distribution models are optimized using categorical cross-entropy loss and use the softmax activation function in the output layer to ensure that the sum of the outputs is one. Meanwhile, the volume and complete models are optimized using the MSE loss and have no activation function in the output layer.

Volume Baseline

The volume baseline, α_{1hr}, makes forecasts based on the average number of incidents in each hour of the week. Thus it captures the weekly seasonality of the demand found in the initial analysis (Section 2.3). It calculates these averages by grouping the incidents in the training data by day of week and hour of the day and then averaging the number of events within each group. Table 5.3 shows some of the forecast values created by α_{1hr} after it has "trained" on the training set. When making forecasts for the validation data, α_{1hr} outputs the forecast value of the row corresponding to the weekday and hour of the time point for each of the time steps in the validation set. For example, if we want to forecast the number of incidents at some Monday at 1 am, α_{1hr} forecasts 8.52. We can easily interpret this forecast; there was an average of 8.52 incidents occurring on Mondays at 1 am in the training data.

Weekday	**Hour**	**Forecast**
1	0	8.94
1	1	8.52
1	2	7.00
...	...	...
4	16	17.44
4	17	16.76
...	...	...
7	23	11.21

Table 5.3: Example of a lookup table representing the volume baseline α_{1hr}. These values correspond to the forecasting values of α_{1hr} after being "trained" on the training set.

The pseudo-code below shows how we create the volume baseline and how it makes predicitons.

```
def create_volume_baseline(incidents):
    # Group incidents on day of week and hour.
    groups = incidents.groupby(
        [incidents.dayofweek, incidents.hour]
    )
    # Calculate average within each group.
    groups = groups.sum() / groups.count()
    return groups

def baseline_forecast(baseline, incidents):
    forecasts = []
    for incident in incidents:
        pred = baseline[incident.dayofweek, incident.hour]
        forecasts.append(pred)
    return forecasts
```

Distribution Baseline 1

The first distribution baseline, β_{total}, makes forecasts based on the spatial distribution of the incidents in the training set across all time steps. It calculates this distribution by summing the number of events that have occurred in each grid cell and then dividing by the total number of events. This calculation is expressed mathematically in Equation 5.1, where n is the last time point in the training set, and $t > 0$ is the time horizon of forecast. As usual, we let $\mathbf{y}_t \in \mathbb{R}^M$ be a vector of the number of incidents occurring in the M spatial regions at time point t.

$$\beta_{total}(n+t) = \frac{1}{\sum_{\tau=1}^{n} \sum_{i=1}^{M} \left(y_\tau\right)_i} \sum_{\tau=1}^{n} \mathbf{y}_\tau. \tag{5.1}$$

Note that the forecast of $\beta_{total}(n+t)$ does not depend on the time horizon t; β_{total} forecasts the same distribution vector for all time steps.

Distribution Baseline 2

The second distribution baseline, β_{8hr}, creates 21 different distribution predictions: one for each 8hr bucket of the week. These are created much in the same way as β_{total}, but we group the training set by day of the week and 8hr bucket before averaging within each group. Similarly to α_{1hr}, it makes forecasts by referring its “lookup”-table of distributions based on the time bucket of the time step to forecast.

Distribution Baseline 3

The third distribution baseline, β_{1hr}, creates a distribution forecast for each hour of the week, similarly to β_{8hr} but with 168 different values.

5.4.2 Benchmarks

We use the MEDIC, All 0s, and MLP methods discussed in Setzler et al. [2009] as forecasting model benchmarks. All of these models are in the complete category.

MEDIC

We implement the MEDIC method as described in Equation 4.1. However, we only have over four years' worth of data available, while the original MEDIC method uses five years of data. We adapt the method by reducing the number of points included in the model to what we have available. For the test set, this means that the forecasts before 07.01.2019 are made with 16 data points, while the forecasts for the next three weeks are made with 17, 18, and 19 points, respectively. The remaining forecasts are made with the complete MEDIC method with 20 data points. For simplicity, we subtract 52 weeks to go one year back.

All Zeroes

The *All 0s* model forecasts zeros for all grid cells at all times.

Setzler

We implement the complete MLP model proposed in Setzler et al. [2009] and dub it the Setzler model. It has a single hidden layer with four neurons that use the sigmoid activation function. The inputs consist of the one-hot encoded hour (0-23), day of the week (1-7), month (1-12), and season (0-3), making a total of 47 binary input nodes. The output layer has no activation function.

5.5 Online Forecasting Methods

We create online versions of our proposed offline forecasting methods to make the models dynamic and get the most out of the available data. In practice, we use a hybrid approach for the online models by first training them offline on the training set and then continue with online learning on the validation/test set. Our online models are direct extensions of their offline counterparts: we perform online learning after their initial offline learning.

5.5.1 Neural Networks

For our online neural network models, we start off with the trained offline version of the models. Then, we lower the learning rate of the optimizer (from 0.001 to 0.0001) to mitigate the catastrophic forgetting problem mentioned in Section 3.2.2 by avoiding putting too much weight on new samples. Then we make forecasts for each hour of the first day of the validation or test set and store those forecasts in a vector. Then we train the model one epoch on those 24 samples. We continue forecasting and training one day at a time until we have made forecasts for the entire validation set. The pseudo-code below details this online forecasting process.

```
def forecast_online(model, inputs, targets):
    all_forecasts = []
    # Loop over 24 inputs at a time.
    for i in range(0, len(inputs), 24):
        x = inputs[i:i+24]
        y = targets[i:i+24]
        forecasts = model.predict(x)
        all_forecasts.append_all(forecast)
        model.train(x, y)  # Train on new samples.
    return all_forecasts
```

Because the LSTM volume and distribution models are trained on normalized outputs, we have to normalize the validation/test targets before feeding them into the online forecasting function. We also have to inverse the normalization on the returned forecasts to transform the raw forecasts back to the original scale.

5.5.2 Baselines

The online versions of our baseline methods are essentially cumulative moving averages versions of the offline average baselines. The resulting learning approach produces outputs as if we "retrain" the models from scratch with each new data received, but the computations efficient as described in Section 3.1.4. Hence, these methods are safe from the catastrophic forgetting problem and storage or runtime issues. However, because all samples are weighted equally, these models are not good at adapting to changes over time.

The pseudo-code below shows how we implement online learning for our baselines. The update_baseline function implements the CMA update calculation described in Equation 3.3.

```
# T is an index representing the start of the test set.
# t is the number of time steps to predict at a time.
def baseline_forecast_online(baseline, incidents, T, t):
    # Create initial baseline based on training data.
    training = incidents[:T]
    baseline = create_baseline(training)
    all_forecasts = []
    for T in range(test_start, len(incidents), t):
        # Make forecasts for time period.
        targets = incidents[T:T+t]
        forecasts = baseline_predict(baseline, targets)
        all_forecasts.append_all(forecasts)
        # Updated baseline.
        baseline = update_baseline(baseline,targets, T, t)
    return all_forecasts

def update_baseline(baseline, new_samples,T,t):
    updated_baseline = (new_samples + T*baseline)/(T+t)
    return updated_baseline
```

The online version of the simplest distribution baseline, β_{total}, is implemented as shown by the pseudo-code above, with $t = 24$. Our "grouped" baselines (β_{1hr}, β_{8hr} and α_{1hr}) has to maintain one CMA for each group which leads to slightly more logic than what shown in the pseudo-code above, but the fundamentals are the same.

Chapter 6

Experimental Results

The results from the architecture selection, validation, and test phases are presented in this chapter.

6.1 Architecture Selection Results

The purpose of the architecture selection phase is to find the best architecture for each neural network model. The results of the architecture selection phase for the distribution, volume, and complete models are shown in Tables 6.1, 6.2, and 6.3 respectively. The selected architecture is denoted as LxN, where L is the number of layers and N is the number of neurons in each layer. The postfix of the method names specifies what input set of the model as specified in Table 5.2.

Note that the training errors of the MLP and LSTM models are not directly comparable because they use different cross-validation approaches (the MLP models use 5-fold while the LSTM models use 80/20 hold-out cross-validation, as explained in Section 5.3). In addition, the training errors of the volume and complete LSTM models are on a different scale because the models are trained with normalized outputs. We are not interested in comparing the different models in this phase because our focus is on selecting the best architecture for each model.

From the results of the distribution models in Table 6.1, we make three observations. Firstly, we see that all of the models chose three hidden layers. Secondly, we observe that the LSTM models prefer considerably more neurons in each layer compared to the MLP models. Indeed, the LSTM models consistently chose the most complex architecture possible (3x128 neurons), while the MLP models chose only 16 or 32 neurons in each layer. Lastly, we note that the weather set achieved the best training error among both the MLP and LSTM models.

The architectures chosen for the volume models are more mixed. We note two matters of interest from the results presented in 6.2. Firstly, it might seem like the input sets affect the architecture more than the model type. For example, the models with the date input set chose 128 neurons in each layer, while the models with combined weather and date inputs had relatively simple architectures with 3x16 and 2x32 neurons. Secondly, we see that the weather input set achieved the lowest training error for both model types, similarly to the distribution models.

The architectures chosen for the complete models are presented in Table 6.3. We make three main observations from these results. Firstly, we note that the complete models generally chose simpler architectures than the models of the two other categories. All of the complete models chose two hidden layers with 4-64 neurons. Secondly, we see that the LSTM models prefer more complex architectures than the MLP models, with 32-64 versus 4-8 neurons in each layer. This is similar to what we saw in the results of the distribution models. Lastly, we note that the basic input set achieved the lowest training error among the MLP models, while the weather set did the same among the LSTM models.

Method	Nodes	Average training CCE
MLP_b	3x16	5.8054
MLP_d	3x8	5.8052
MLP_w	3x16	**5.8037**
$\text{MLP}_{w\&d}$	3x8	5.8047
LSTM_b	3x128	5.8494
LSTM_d	3x128	5.8495
LSTM_w	3x128	**5.8490**
$\text{LSTM}_{w\&d}$	3x128	5.8495

Table 6.1: The best architecture for each proposed *distribution* neural network model and its average training error during the architecture selection phase.

Method	Nodes	Average training MSE
MLP_b	2x32	22.3580
MLP_d	3x128	22.4657
MLP_w	3x64	**22.0083**
$\text{MLP}_{w\&d}$	3x16	22.2789
LSTM_b	3x64	0.009466
LSTM_d	2x128	0.009753
LSTM_w	3x64	**0.009444**
$\text{LSTM}_{w\&d}$	2x32	0.009663

Table 6.2: The selected architecture for each proposed neural network *volume* model and its average training error during the architecture selection phase.

Method	Nodes	Average training MSE
MLP_b	2x8	**0.003633**
MLP_d	2x4	0.003636
MLP_w	2x8	0.003635
$\text{MLP}_{w\&d}$	2x8	0.003638
LSTM_b	2x32	1.43318e-05
LSTM_d	2x64	1.43303e-05
LSTM_w	2x64	**1.43299e-05**
$\text{LSTM}_{w\&d}$	2x64	1.44027e-05

Table 6.3: The best architecture and its average training error of the *complete* models during the architecture selection phase.

6.2 Validation Results

The validation results determine which methods are selected to proceed to the test phase. We choose the methods with the lowest error in each category. We use the MSE to select the volume and complete models, but we list the MAE in the results as well. We select distribution models based on their categorical cross-entropy. The validation results of all the proposed distribution, volume, and complete models, with and without online learning, can be found in Tables 6.4, 6.5, and 6.6 respectively. We highlight the best offline and online results in each category.

We make four observations on the validation results of the distribution models shown in Table 6.4. Firstly, we see that the simplest distribution model, β_{total}, has the lowest categorical cross-entropy both online and offline. In contrast, the other two distribution baselines, β_{8hr} and β_{1hr}, are considerably worse. Secondly, we observe that all of the LSTM models perform better than all of the MLP models. Thirdly, it seems like the input sets of the neural network models do not affect the performance much; the results within the MLP or LSTM models are very close. Finally, we see that all of the models improve with online learning.

From the validation results of the volume models shown in Table 6.5, we make three remarks. Firstly we see that all of the models improve with online learning, just like the distribution models. Secondly, we note that the input sets have more influence among the volume models than the distribution models. For example, the date input set results in high errors for both the MLP and LSTM models, while the weather and basic input sets produce good forecasts for both model types. Finally, we see that all of the neural network models outperform the volume baseline α_{1hr}.

The validation results of the complete models presented in Table 6.6 also show three interesting patterns. Firstly, we see that all of the models improve with online learning, just like the distribution and volume models. Secondly, we observe that the model type influences the performance more than the input sets and that the MLP models are generally better than the LSTM models. Thirdly, we see that the LSTM models improve more with online learning than the MLP methods, reducing the performance gap between the two model types.

We select the two best models from each category to proceed to the testing phase. The chosen distribution models are the online β_{total} and online LSTM$_w$ models, while the selected volume models are the online MLP$_b$ and online LSTM$_w$. The two best complete are the online MLP$_d$ and online MLP$_w$.

Method	CCE	Online CCE
β_{total}	**5.8518**	**5.8479**
β_{8hr}	6.0776	6.0337
β_{1hr}	7.1632	6.8343
MLP$_b$	5.8607	5.8597
MLP$_d$	5.8594	5.8586
MLP$_w$	5.8611	5.8595
MLP$_{w\&d}$	5.8590	5.8578
LSTM$_b$	5.8545	5.8527
LSTM$_d$	5.8552	5.8528
LSTM$_w$	5.8548	5.8525
LSTM$_{w\&d}$	5.8558	5.8533

Table 6.4: Validation errors of the proposed distribution methods.

Method	MSE	Online MSE	MAE	Online MAE
α_{1hr}	24.8293	24.4766	3.7546	3.7319
MLP$_b$	23.6448	**22.7188**	**3.6935**	**3.6528**
MLP$_d$	24.2637	23.8844	3.8229	3.7842
MLP$_w$	23.6387	23.0511	3.7661	3.6954
MLP$_{w\&d}$	23.4240	23.1589	3.7387	3.7223
LSTM$_b$	23.4473	23.0577	3.7291	3.6957
LSTM$_d$	24.3394	23.5889	3.8800	3.7703
LSTM$_w$	**23.1635**	22.7410	3.7052	3.6752
LSTM$_{w\&d}$	23.5593	23.2741	3.7417	3.7179

Table 6.5: Validation errors of the proposed volume methods.

Method	MSE	Online MSE	MAE	Online MAE
MLP$_b$	0.0037455	0.0037428	0.0047374	0.0043507
MLP$_d$	0.0037461	0.0037432	0.0045820	**0.0043154**
MLP$_w$	**0.0037448**	**0.0037414**	**0.0045455**	0.0043318
MLP$_{w\&d}$	0.0037485	0.0037452	0.0046975	0.0043414
LSTM$_b$	0.0037822	0.0037498	0.0074629	0.0052450
LSTM$_d$	0.0037802	0.0037478	0.0076093	0.0053806
LSTM$_w$	0.0037818	0.0037495	0.0078251	0.0054976
LSTM$_{w\&d}$	0.0037811	0.0037492	0.0074902	0.0054961

Table 6.6: Validation errors of the proposed complete methods.

6.3 Test Results

We select the two best distribution, volume, and complete models from the validation phase to retrain and run on the test set according to the method outlined in Section 5.3. Next, the two distribution models are combined with each of the two volume models, making a total of four split models that are tested. As only online models were selected, we omit the online prefix of the model names for the rest of this chapter for brevity.

The final complete test results can be found in Table 6.7. We see that the complete MLP$_w$ model achieves the lowest MSE, closely followed by the two split models with the internal LSTM$_w$ distribution model. The *All 0s* model has the lowest MAE.

It is also worth mentioning that the Setzler method achieves a better MSE than the MEDIC method in our case, in contrast to the original study [Setzler et al., 2009]. In the original study, the MEDIC method outperformed their proposed ANN method at their smallest spatial and temporal levels of granularity (grid-size of 2x2 miles ≈ 3.2x3.2 km and 1-hr time buckets).

In addition to calculating the MSE and MAE of the complete forecasts of the models, we also calculate the errors of their volume and distribution forecasts as described in Section 5.2.

The errors of the volume forecasts can be found in Table 6.8. We see that the dedicated volume models produce the best volume forecasts, followed by MEDIC, the complete neural networks, and finally, *All 0s*. The volume forecasts made by the models on the first and last week of the test set are illustrated in Figure 6.1 and Figure 6.2. We see that the dedicated volume models and the MEDIC models produce better volume forecasts than the complete neural network models, which tend to be too conservative. The complete neural networks do not capture the weekly seasonality well; their volume predictions look very similar every day and do not capture the increase in incidents during working hours or Friday and Saturday nights.

Method	**MSE**	**MAE**
All 0s	0.0043334	**0.0027283**
MEDIC	0.0040064	0.0046825
Setzler	0.0038105	0.0049510
dist β_{total} + vol MLP$_b$	0.0038172	0.0048571
dist β_{total} + vol LSTM$_w$	0.0038179	0.0048555
dist LSTM$_w$ + vol LSTM$_w$	0.0037996	0.0048874
dist LSTM$_w$ + vol MLP$_b$	0.0037995	0.0048889
complete MLP$_b$	0.0038011	0.0044522
complete MLP$_w$	**0.0037983**	0.0044777

Table 6.7: Test errors of the complete forecasts of the proposed and benchmark methods. Note that all models except *All 0s*, *MEDIC* and *Setzler* use online learning.

Method	**MSE**	**MAE**
All 0s	280.6049	15.1939
MEDIC	25.2394	3.8183
Setzler	33.8975	4.3933
complete MLP$_b$	34.5490	4.4373
complete MLP$_w$	35.2900	4.5135
volume LSTM$_w$	22.9890	3.7027
volume MLP$_b$	**22.9108**	**3.6988**

Table 6.8: Test errors of the volume forecasts of the proposed and benchmark methods. Note that all models except *All 0s*, *MEDIC* and *Setzler* use online learning.

The errors of the distribution forecasts are shown in Table 6.8. We see that the dedicated distribution models produce the best forecasts, followed by the complete neural network models, the *All 0s* method, and finally the MEDIC method. Illustrations of the distribution forecasts at 12pm on the first day of the test set and 9pm on the last day of the test set can be found in Figures 6.3 and 6.4 and Figures 6.5 and 6.6 respectively. We see that the forecasts of the neural network models are very similar, while the MEDIC model has a significantly wider spatial spread. All the forecasts of the models (except for the All 0s model) are concentrated around the city center of Oslo.

Method	**CCE**
All 0s (even dist)	8.6249930
MEDIC	10.487533
Setzler	6.2471037
complete MLP$_b$	6.0354133
complete MLP$_w$	6.0745206
dist β_{total}	**5.8713098**
dist LSTM$_w$	5.8967190

Table 6.9: Categorical cross-entropy errors of the distribution forecasts on the test set. A lower error is favorable. Note that all models except *All 0s*, *MEDIC* and *Setzler* use online learning.

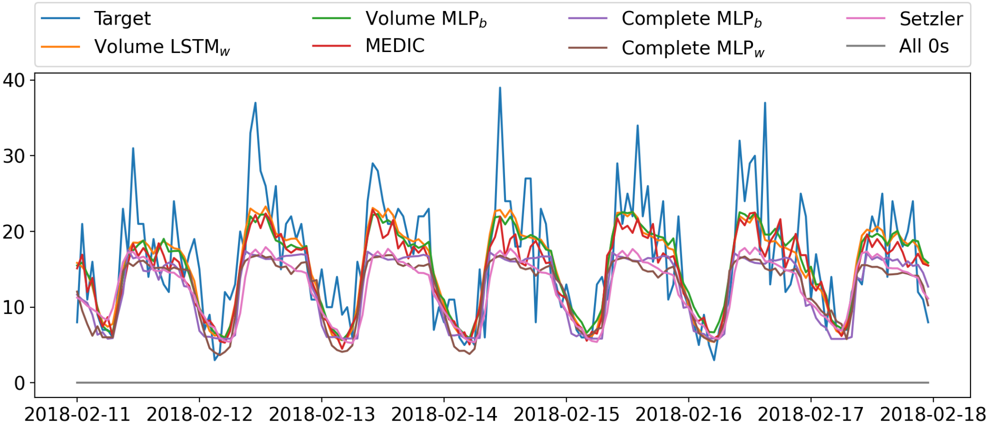


Figure 6.1: Volume forecasts on the first week of the test set, starting on Sunday the 11th of February 2018. The complete neural network methods tend to underestimate the demand volume, especially during working hours and Friday and Saturday night.

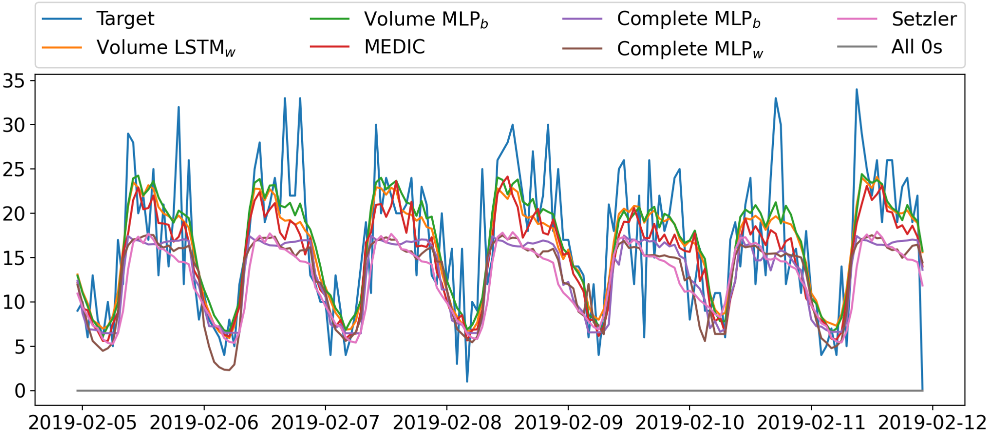


Figure 6.2: Volume forecasts on the last week of the test set, starting on Tuesday the 5th of February 2019. The complete neural network methods tend to underestimate the demand volume, especially during working hours and Friday and Saturday night.

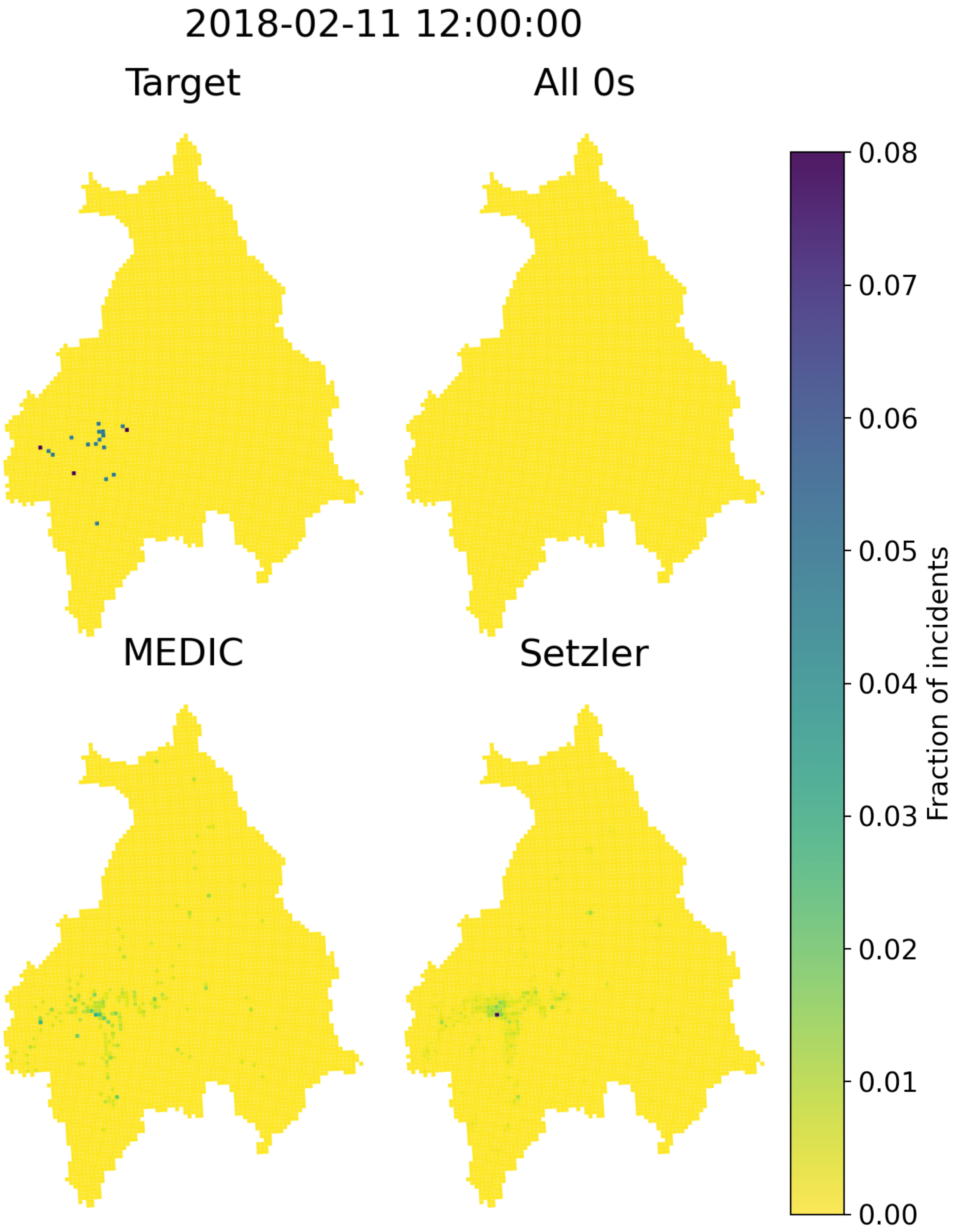


Figure 6.3: The actual EMS demand distribution and the distribution forecasts of the benchmark models for Sunday 11th of February 2018 (the first day of the test set) at 12pm. Refer Figure 2.2 for a geographical reference of the grid.

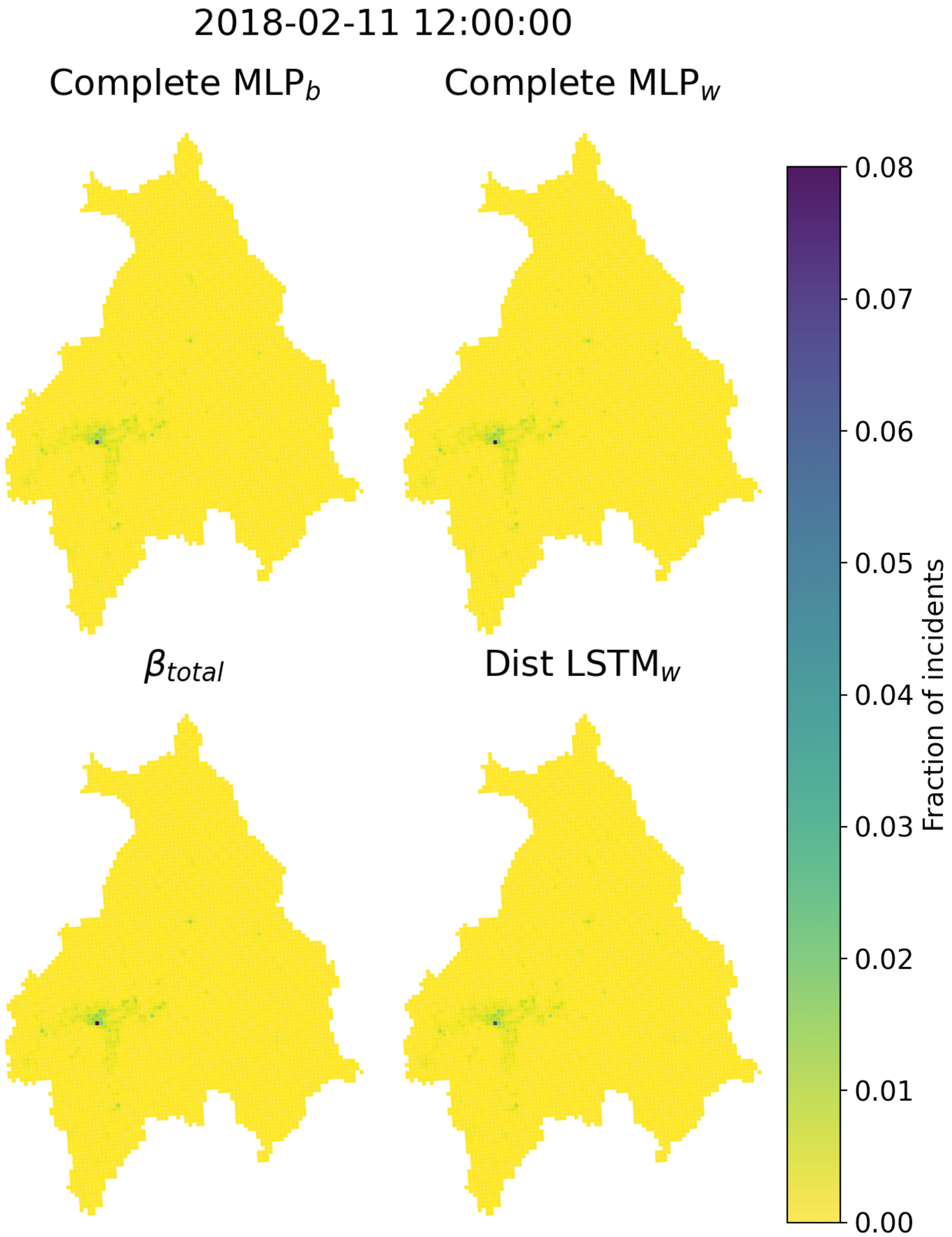


Figure 6.4: The distribution forecasts of the proposed models for Sunday 11th of February 2018 (the first day of the test set) at 12pm. Refer Figure 2.2 for a geographical reference of the grid.

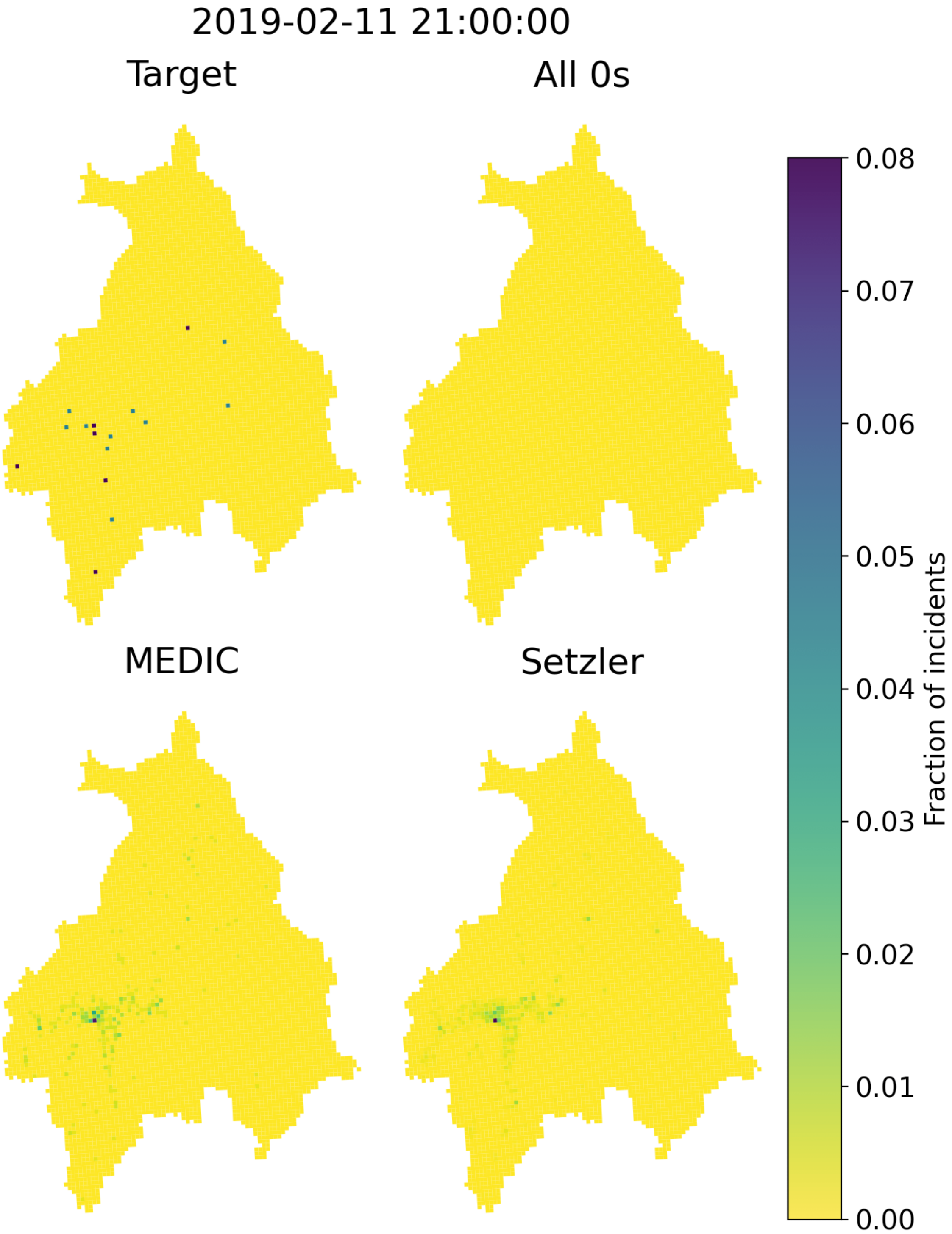


Figure 6.5: The actual EMS demand distribution and the distribution forecasts of the baseline models for Monday 11th of Febrary 2019 (the last day of the test set) at 9pm. Refer Figure 2.2 for a geographical reference of the grid.

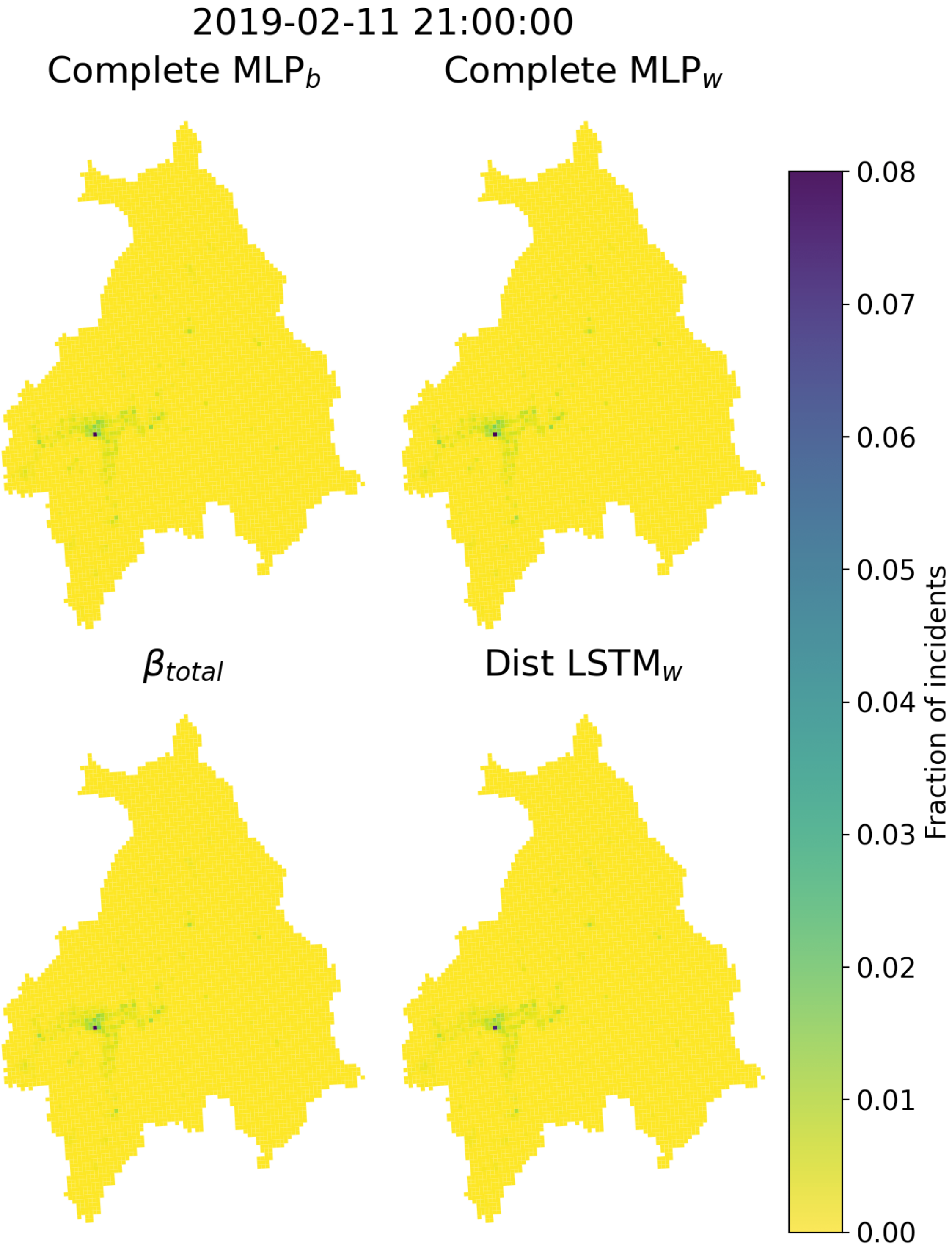


Figure 6.6: The distribution forecasts of the proposed models for Monday 11th of Febrary 2019 (the last day of the test set) at 9pm. Refer Figure 2.2 for a geographical reference of the grid.

Chapter 7

Discussion and Conclusion

In Section 1.4 we defined our research goal and questions. Now, in Section 7.1, we revisit those research questions and discuss them in light of our results from Chapter 6. We conclude our research in Section 7.2 and identify possible areas of future work in Section 7.3.

7.1 Discussion

7.1.1 How can the EMS demand in Oslo and Akershus be Forecast Accurately at a Fine Spatio-Temporal Scale?

We have tested a variety of input sets, model types, and training regimes for predicting the EMS demand in Oslo and Akershus at a fine spatial and temporal scale of 1x1km spatial regions and 1-hr time intervals. In this section, we discuss what we have found to be the best approaches for making accurate forecasts of the EMS demand.

Model Type

The β_{total} model made the best distribution forecasts in all the phases, as evident in Tables 6.4 and 6.9. All of the neural network models also made fairly good spatial forecasts, as evident in Table 6.9. We can see from Figures 6.3, 6.4, 6.5, and 6.6, that the neural network models make forecasts almost identical to those of β_{total}. MEDIC, on the other hand, makes terrible distribution forecasts. This is caused by the fact that MEDIC makes forecasts as if an incident occurring increases the likelihood that an incident will occur at the same area again at the same time point in later weeks. However, the EMS incidents are mostly independent and stochastic, which means MEDIC often forecasts incidents that are unlikely to occur.

We find it surprising that β_{total}, being the most stationery and the simplest of the distribution models tested, made the most accurate distribution forecasts. Intuitively, we expected the distribution of demand to show some pattern in time. For example, there might have been a more centralized distribution in working hours, while people might be drawn toward the fjord when on warm and sunny Sunday. However, we saw few hints of such differences in the distributions in Figure 2.10, which show the distribution of the EMS demand in Oslo and Akershus at different times on weekdays. Seeing how our best distribution model has captured no such variations, there must either be no variations in our available data, or our proposed models are unable to detect and leverage them.

In contrast, Zhou [2015] found significant variations in the density of ambulance demand with patterns in time and space for both Toronto and Melbourne, despite having half of the amount of data that we do (four years vs. two years). This might be due to Melbourne and Toronto having a higher population density than Oslo and Akershus (Toronto, Melbourne, and Oslo and Akershus have population densities of approximately 5 000 people/km^2, 500 people/km^2, and 200 people/km respectively). Indeed, Zhou noted that the patterns were most prominent in the most densely populated areas of the cities. Another possible explanation is that Zhou [2015] uses a continuous spatial domain, while we only have access to the anonymized grid-mapped location of events. This might make it harder to detect subtle differences in the underlying density function. Either way, it might be interesting to focus on the more densely populated areas of Oslo and Akershus to see if we can detect some patterns. It might be worthwhile using different models for different grid cells as they did in Chen et al. [2016]. An LSTM model should be tested in such a case, as it seemed like the LSTM models were slightly better than the MLP models at forecasting EMS distributions.

The online volume MLP$_b$ model made the best volume forecasts in the validation and test phase, as seen in Tables 6.5 and 6.8. The volume MLP and LSTM models had similar architecture complexity and validation errors. However, since the LSTM model is significantly more complex than the MLP models without producing better results, we think an MLP model is more suited for modeling the EMS demand volume. All of the neural network models outperformed the simple α_{1hr} model, which indicates that there are patterns in time other than the weekly seasonality that the neural networks manage to leverage.

The complete MLP models outperformed the complete LSTM models as seen in Table 6.6. The table also shows that the performance gap between the MLP and LSTM models shrunk considerably with online learning. It might be that the LSTM models can improve further and possibly supersede the performance of the MLP models if we have more training data. Indeed, the LSTM models chosen from the architecture selection phase have more complex architectures than the MLP models, which means they can capture more intricate patterns in the demand.

Input Data

In Figures 6.1 and 6.2, we saw that the volume MLP$_b$ model was able to capture the weekly seasonality. This shows that the basic inputs are enough to capture the large-scale patterns of the EMS demand. The neural network models also seem to have leveraged some of the annual seasonality of the EMS demand since they outperformed the α_{1hr}, which naively models the weekly seasonality. In Section 6.2, we saw that the performance of the distribution and complete models were not affected much by the different input sets. This indicates that the neural networks were unable to find much useful information in the extra input variables.

Our validation results in Tables 6.4, 6.5, and 6.6 show that the inclusion of the day of the month input did not improve model performance. On the contrary, it resulted in higher MSE for both volume and distribution neural network models. The date input set resulted in the best performance among the LSTM complete models but was outperformed by the complete MLP models; thus, the date set never made it past the validation phase. We hoped that the date input would let the models identify special day effects, such as New Year's Eve, and thus make better forecasts. Figure 7.1 shows the actual EMS demand and the forecasts made by the proposed and baseline models on New Year's Eve 2018. We see that the models are unable to capture this regular increase in incidents. A special-day input flag might yield better results but requires manual selection of the days to include. Such a scheme was used with success in Channouf et al. [2007] with flags for New Year's Eve and a local festival.

The models with the weather input set performed well in the validation and testing phase, as

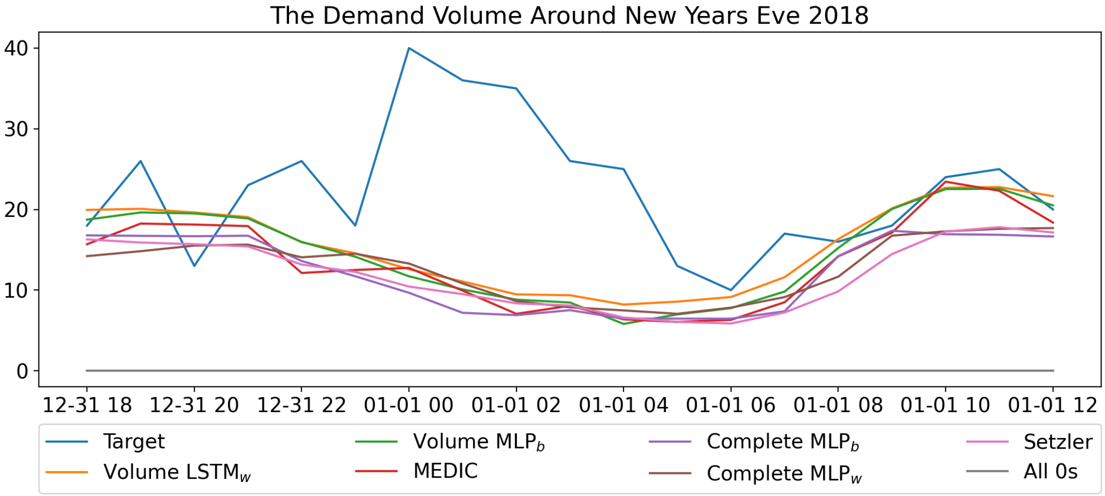


Figure 7.1: The actual demand volume on New Years Eve 2018, and the forecasts of the proposed and baseline models.

shown in Tables 6.4, 6.5, and 6.6 and 6.7, 6.9, and 6.8. The complete MLP$_w$ model was the best of the complete models and LSTM$_w$ was the second-best volume and distribution model. This indicates that the weather have the potential of improving forecasts. We discuss this further in Section 7.1.3.

The *Setzler* model uses the season of the year as an input variable in addition to our *basic* input set. We chose not to include the season variable in any of our input sets because it is implicitly covered by the month inputs and seemed redundant. The *Setzler* model makes slightly better volume forecasts and worse distribution and complete forecasts compared to our proposed complete MLP models. The fact that the Setzler model makes better volume forecasts than our complete models indicates that the inclusion of the season variable might make it easier for the neural network to capture the annual seasonality found in Section 2.3.

Metrics

We can see how much the choice of performance metric influences the performance of different methods in Table 6.7, where the *All 0s* method achieves the lowest MAE although it has no practical value. This happens because our data is very skewed towards zero, so most of the zeros “forecast” by *All 0s* are on target. However, when *All 0s* do make mistakes, they can be quite large, but these are not punished harshly by MAE. The MSE punishes deviations more than MAE, which makes it a better metric in our case as we are most interested in the deviations, with zero incidents being the base case in most regions. However, we saw that the complete and, to a lesser degree, volume models tend to underestimate the EMS demand volume in Figures 6.1 and 6.2 even though they are optimized for MSE. It might seem like a different metric might be even more appropriate than MSE for our case. We could, for example, implement a metric that punishes positive errors more than negative errors, like the one described in Equation 7.1.

$$L(\hat{y}, y) = \begin{cases} (\hat{y} - y)^2, & \text{if } \hat{y} > y \\ 2(\hat{y} - y)^2, & \text{otherwise} \end{cases} \tag{7.1}$$

Such a metric might be appropriate in the EMS field because it is vital that there are enough resources to respond to the demand. This should be investigated further.

7.1.2 Is a Split Model or a Complete Model Better for Modeling EMS Demand?

The complete models achieved lower MSE on their complete forecasts than the split models, as seen in Table 6.7. However, the specialized volume and distribution models produced better volume and distribution forecasts as seen in Tables 6.9 and 6.8. This is not surprising, as the models have been optimized for making either volume, distribution, or complete forecasts. However, it raises the question of which approach has the most practical value.

The complete models are essentially trying to solve a more difficult problem than the split models by forecasting the number and location of incidents simultaneously. Intuitively, it is a lot easier to forecast that there will occur ten incidents than to forecast that there will occur ten incidents at these exact locations. Because of the stochastic nature of the EMS demand, it is very hard to make accurate forecasts in both volume and location, causing the complete models to only forecast incidents that are very likely to happen. This makes them underestimate the number of incidents, as apparent in Figures 6.1 and 6.2. This could be remedied by using a different metric as discussed in 7.1.1, but the nature of the complete problem would probably still cause the complete models to produce worse volume forecasts than split methods.

The distribution forecasts of the specialized distribution models are also significantly better than those of the other models, as evident in Table 6.9, although it is not so easily recognizable in the distribution illustrations in Figures 6.4 and 6.6.

The MEDIC model, although a complete model, makes pretty good volume forecasts. It does not suffer from the underestimation problem of the neural networks because it simply averages previous incidents instead of trying to minimize its MSE. While this tactic works well for the demand volume, it makes poor distribution forecasts.

We conclude that a split model produces more useful information for EMS providers than complete models in our case because they make better volume and distribution forecasts.

7.1.3 Does Weather Influence the Spatial Distribution of EMS Demand in Oslo and Akershus?

As mentioned in section 7.1.1, the weather inputs looked promising for the distribution models. The LSTM$_w$ model was the best distribution neural network model but was outperformed by the basic β_{total} model.

Note that we used historical weather data in our weather input set. In a real-world application, we will have to use weather forecasts instead, which are naturally less accurate. Wong and Lai [2013] showed that also the use of weather forecasts can improve the EMS demand forecasts, but to a lesser degree than historical weather data. The previous work that found increased performance with the inclusion of weather data has been on a much larger spatial and temporal scale (daily forecasts for entire cities)[Wong and Lai, 2010, 2013; Wong and Lin, 2020; Thornes et al., 2014], which might make the pattern more pronounced. We could try to include more types of weather data such as wind speed or cloud coverage, but since the previously mentioned related work has found that temperature is the most useful parameter, we do not think this will improve the results noticeably. With this in mind, we think that there might be some patterns in our dataset distribution of the EMS demand related to the weather, but that these are too slight to improve the distribution forecasts significantly and thus have little practical value at the moment.

7.1.4 Can Online Learning Be Used to Improve EMS Demand Forecasts in Oslo and Akershus?

In the validation phase, we found that every single model improved with online learning, as shown in Tables 6.4, 6.5, and 6.6. This is not surprising, seeing how online learning leverages more of the available data and makes the models capable of adapting to possible changes in the underlying function, such as an increase in population.

Although online learning is an excellent tool for improving performance, it is susceptible to normalization and diminishing update problems.

For example, the normalization of the outputs of the volume LSTM models can become outdated. The normalization is fitted on the training data, and as such, the normalization can produce unwanted outputs for test values not represented in the training data. This can happen if the underlying function changes substantially, such as a significant increase in population. In such a case, the normalized output values can become so large that they will cause problems for the training of the model. This potential issue can be mitigated by re-fitting the normalization periodically and retraining the network from scratch. The same problem could occur with the normalization of the weather data, if the weather was to change significantly over time.

The β_{total} might struggle with adapting to changes in the underlying model as all the incidents are weighted equally, meaning that each update will affect the model less with time. This can be mitigated by using a simple moving average that takes the mean of the last k data points.

7.2 Conclusion

We proposed and tested a variety of models for forecasting the hourly EMS demand in 1x1km spatial regions in Oslo and Akershus. This problem is challenging because of the sparsity of EMS data at such a fine spatio-temporal resolution. We propose two different approaches for forecasting the EMS demand; a complete approach that directly forecasts the incidents in each of the N spatial region $\mathbf{y} \in \mathbb{R}^N$, and a split approach that forecasts the volume $\delta = \sum_{i=1}^{N} y_i$ and distribution $f = \frac{1}{\delta}\mathbf{y}$ of the demand separately. We proposed two different types of neural networks (MLP and LSTM) with four different input sets for forecasting the complete, volume and distribution of the EMS demand. In addition, we proposed some simple aggregation methods for forecasting the EMS demand volume and distribution.

We conclude that split models are better suited for modeling EMS demand than complete models.

Among the models tested, we find that a simple aggregation model is the best at modeling the spatial distribution of the EMS demand in Oslo and Akershus. The demand volume is best captured by an MLP with two hidden layers of 32 nodes each, and hour, day of the week, and month input features.

We find that all models improve with online learning, which indicates that this is an excellent tool for improving forecast accuracy.

We also find indications of the weather influencing the spatial distribution of EMS demand. However, the relationship seems too weak to have practical value.

7.3 Future Work

We believe that the forecasting of the EMS demand volume might be improved by including flags for special days such as New Year's Eve. There might be similar spikes in incidents related to concerts or other special events, which might also affect the spatial distribution of events. For example, the area in which a festival or concert is taking place might experience more incidents than usual. Gathering data on such events and feeding their location into the models is nontrivial but seems an interesting approach that, to the author's knowledge, has not been studied before.

In the EMS field, there should always be enough resources to respond to the demand of the public. Therefore the volume models cannot be followed blindly as they often underestimate the demand, as evident in Figures 6.1 and 6.2. In future research, the researchers should consider adopting a different metric that penalizes underestimates harder than overestimates, as we discussed in Section 7.1.1. Another approach that could provide even more useful information to the EMS providers is to make prediction intervals forecasts instead of point forecasts, as described in Section 3.1.3.

The spatial forecasts might be improved by looking at the most populous regions separately. One might be able to detect patterns between time or weather and the distribution of the demand in regions with many incidents. However, the practical value of leveraging such patterns is uncertain as the regions in question are spatially close to each other, and therefore probably will not affect the optimal location of the ambulances much. It might also be interesting to look for patterns in demand distribution linked to population movement data, which could be gathered from mobile providers or similar. One should also test variations of the spatial β_{total} model, such as a simple or weighted moving average model, when more data becomes available.

There is a lot of work remaining related to our research goal of positioning the ambulances in Oslo and Akershus optimally. Firstly, we have to choose an optimization model for solving the ambulance location problem. There exists a variety of such models with different assumptions and objectives, as mentioned in Section 1.3. Secondly, we need a model for the travel times between the spatial regions. There are many different approaches for this, including models based on trip distance [Budge et al., 2010; Westgate et al., 2013] or GPS data and individual road segments [Westgate et al., 2016; Li et al., 2015]. Lastly, one has to remember to add the planned regular incidents to the forecast incidents in order to get correct estimates of the total EMS demand.

Bibliography

Aboueljinane, L., Sahin, E., and Jemai, Z. (2013). A review on simulation models applied to emergency medical service operations. *Computers & Industrial Engineering*, 66(4):734 – 750.

Aringhieri, R., Bruni, M., Khodaparasti, S., and van Essen, J. (2017). Emergency medical services and beyond: Addressing new challenges through a wide literature review. *Computers & Operations Research*, 78:349 – 368.

Başar, A., Çatay, B., and Ünlüyurt, T. (2012). A taxonomy for emergency service station location problem. *Optimization Letters*, 6:1–14.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Budge, S., Ingolfsson, A., and Zerom, D. (2010). Empirical analysis of ambulance travel times: The case of calgary emergency medical services. *Management Science*, 56:716–723.

Bélanger, V., Ruiz, A., and Soriano, P. (2019). Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. *European Journal of Operational Research*, 272(1):1 – 23.

Channouf, N., L'Ecuyer, P., Ingolfsson, A., and Avramidis, A. (2007). The application of forecasting techniques to modeling emergency medical system calls in calgary, alberta. *Health Care Management Science*, 10:25–45.

Chanta, S., Mayorga, M. E., Kurz, M. E., and McLay, L. A. (2011). The minimum p-envy location problem: a new model for equitable distribution of emergency resources. *IIE Transactions on Healthcare Systems Engineering*, 1(2):101–115.

Chen, A. and Lu, T.-Y. (2014). A gis-based demand forecast using machine learning for emergency medical services. pages 1634–1641.

Chen, A. Y., Lu, T., Ma, M. H., and Sun, W. (2016). Demand forecast using data analytics for the preallocation of ambulances. *IEEE Journal of Biomedical and Health Informatics*, 20(4):1178–1187.

Church, R. and ReVelle, C. (1974). The maximal covering location problem. In *Papers of the Regional Science Association*, volume 32, pages 101–118. Springer-Verlag.

Daskin, M. S. (1983). A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70.

Daskin, M. S. and Stern, E. H. (1981). A hierarchical objective set covering model for emergency medical service vehicle deployment. *Transportation Science*, 15(2):137–152.

Erkut, E., Ingolfsson, A., and Erdoğan, G. (2008). Ambulance location for maximum survival. *Naval Research Logistics (NRL)*, 55(1):42–58.

Gendreau, M., Laporte, G., and Semet, F. (1997). Solving an ambulance location model by tabu search. *Location Science*, 5(2):75 – 88.

Gendreau, M., Laporte, G., and Semet, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel computing*, 27(12):1641–1653.

Grekousis, G. and Liu, Y. (2019). Where will the next emergency event occur? predicting ambulance demand in emergency medical services using artificial intelligence. *Computers, Environment and Urban Systems*, 76:110 – 122.

Haga, D., Bjelke, C., Ersdal, G., Hauglin, O., Hellesø, R., Hesselberg, N., Nilsen, J. E., Telje, J., Tveita, M., Tønnesen, K., Vonen, B., Wold, G., Øen, T. O., Bakkeli, M., and Thoner, J. (1998). Hvis det haster..... — faglige krav til akuttmedisinsk beredskap [if it is urgent..... - professional demands to emergency medicine]. Norwegian official report, Norwegian Ministry of Health and Care Services.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huang, H., Jiang, M., Ding, Z., and Zhou, M. (2019). Forecasting emergency calls with a poisson neural network-based assemble model. *IEEE Access*, 7:18061–18069.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts, 2 edition.

Ingolfsson, A. (2013). *EMS Planning and Management*, pages 105–128. Springer New York, New York, NY.

Ingolfsson, A., Budge, S., and Erkut, E. (2008). Optimal ambulance location with random delays and travel times. *Health care management science*, 11:262–74.

Johansen, K., Rømo, F., and Øyvind B. Hope (2002). Økonomiske konsekvenser av nye krav til responstider i ambulansetjenesten [Economic consequences of new response time demands in the ambulance service]. Technical report, SINTEF.

Jones, S. A., Joy, M. P., and Pearson, J. (2002). Forecasting demand of emergency care. *Health Care Management Science*, 5.

Kohlstrunk, P.-J. (2018). Endringsprosess i et retrospektivt perspektiv [Change Process in a retrospective perspective]. Master's thesis, University of Oslo.

Larsen, M. P., Eisenberg, M. S., Cummins, R. O., and Hallstrom, A. P. (1993). Predicting survival from out-of-hospital cardiac arrest: A graphic model. *Annals of Emergency Medicine*, 22(11):1652 – 1658.

Larson, R. C. (1974). A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers & Operations Research*, 1(1):67 – 95.

Li, Y., Zheng, Y., Ji, S., Wang, W., U, L. H., and Gong, Z. (2015). Location selection for ambulance stations: A data-driven approach. In *Proceedings of the 23rd ACM International Conference on Advances in Geographical Information Systems*. ACM SIGSPATIAL 2015.

Lin, A. X., Ho, A. F. W., Cheong, K. H., Li, Z., Cai, W., Chee, M. L., Ng, Y. Y., Xiao, X., and Ong, M. E. H. (2020). Leveraging machine learning techniques and engineering of multi-nature features for national daily regional ambulance demand prediction. *International Journal of Environmental Research and Public Health*, 17(11).

Losing, V., Hammer, B., and Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274.

Matteson, D. S., MClean, M. W., Woodard, D. B., and Henderson, S. G. (2011). Forecasting emergency medical service call arrival rates. *The Annuals of Applied Statistics*, 5.

McLay, L. A. and Mayorga, M. E. (2010). Evaluating emergency medical service performance measures. *Health care management science*, 13(2):124–136.

Nolan, J. P., Soar, J., Zideman, D. A., Biarent, D., Bossaert, L. L., Deakin, C., Koster, R. W., Jonathan Wyllie, B. B., and Group, E. G. W. (2010). European resuscitation council guidelines for resuscitation 2010 section 1. executive summary. *Resuscitation*, 81(10):1219–1276.

Norwegian Ministry of Health and Care Services (2000). St.meld. nr. 43 [White paper nr. 43]. Technical report, Norwegian Ministry of Health and Care Services.

Oftedahl, L. (2016). Forbedret ambulansetid: Til Stovner på 5 minutter [Improved ambulance time: To Stovner in 5 minutes]. *Ambulanseforum*.

O'Keeffe, C., Nicholl, J., Turner, J., and Goodacre, S. (2011). Role of ambulance response times in the survival of patients with out-of-hospital cardiac arrest. *Emergency Medicine Journal*, 28(8):703–706.

Olah, C. (2017). Understanding lstm networks.

Olsen, S., Ilkka, L., Kurola, J., Silfvast, T., Ekstrand, A., Berlac, P. A., Hansen, P. A., Riddervold, I., Christensen, E. F., Granberg, M., Åhsberg, E., Holmberg, S., Gjesteby, D., Engerström, L., Bárðarson, L., Stefánsson, B., Magnússon, V., Haaheim, H., Steen-Hansen, J. E., Borge, T., Kjøllesdal, J. K., Yang, J., Blomberg, T., and Myrmel, L. (2019). The Nordic emergency medical services. Technical report, Norwegian Directorate of Health.

Price, L. (2006). Treating the clock and not the patient: ambulance response times and risk. *BMJ Quality & Safety*, 15(2):127–130.

Rezaei, M. and Ingolfsson, A. (2021). Assessment of exponential smoothing methods for spatio-temporal forecasting of ems call volumes.

Russell, S. J. and Norvig, P. (2009). *Artificial Intelligence: a modern approach*. Pearson, 3 edition.

Schmid, V. and Doerner, K. F. (2010). Ambulance location and relocation problems with time-dependent travel times. *European Journal of Operational Research*, 207(3):1293 – 1303.

Setzler, H., Saydam, C., and Park, S. (2009). EMS call volume predictions: A comparative study. *Computers & Operations Research*, 36(6):1843 – 1851.

Stand, G.-H. and Bloch, V. V. H. (2009). Statistical grids of Norway. Technical report, Statistics Norway, Department of Economic Statistics.

Steins, K., Matinrad, N., and Granberg, T. (2019). Forecasting the demand for emergency medical services.

The Norwegian Directory of Health (2018). Tid fra AMK varsles til ambulanse er på hendelsessted [Time from EMCC is notified until an ambulance is at the scene]. Quality indicator description, The Norwegian Directory of Health.

Thornes, J. E., Fisher, P. A., Rayment-Bishop, T., and Smith, C. (2014). Ambulance call-outs and response times in birmingham and the impact of extreme weather and climate change. *Emergency Medicine Journal*, 31(3):220–228.

Toregas, C., Swain, R., ReVelle, C., and Bergman, L. (1971). The location of emergency service facilities. *Operations Research*, 19(6):1363–1373.

Westgate, B., Woodard, D., Matteson, D., and Henderson, S. (2013). Travel time estimation for ambulances using bayesian data augmentation. *The Annals of Applied Statistics*, 7.

Westgate, B., Woodard, D., Matteson, D., and Henderson, S. (2016). Large-network travel time distribution estimation for ambulances. *European Journal of Operational Research*, 252.

Wong, H. and Lai, P. (2010). Weather inference and daily demand for emergency ambulance services. *Emergency medicine journal : EMJ*, 29:60–4.

Wong, H. and Lin, J.-J. (2020). The effects of weather on daily emergency ambulance service demand in taipei: a comparison with hong kong. *Theoretical and Applied Climatology*, 141.

Wong, H.-T. and Lai, P.-C. (2013). Weather factors in the short-term forecasting of daily ambulance calls. *International journal of biometeorology*, 58.

Zhou, Z. (2015). *Predicting Ambulance Demand*. PhD thesis, Cornell University.

Zhou, Z. (2016). Predicting ambulance demand: Challenges and methods.

Zhou, Z. and Matteson, D. S. (2015). Predicting ambulance demand: A spatio-temporal kernel approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 2297–2303, New York, NY, USA. Association for Computing Machinery.

Zhou, Z. and Matteson, D. S. (2016). Predicting Melbourne ambulance demand using kernel warping. *Ann. Appl. Stat.*, 10(4):1977–1996.

Zhou, Z., Matteson, D. S., Woodard, D. B., Henderson, S. G., and Micheas, A. C. (2015). A spatio-temporal point process model for ambulance demand. *Journal of the American Statistical Association*, 110(509):6–15.

NTNU

Norwegian University of
Science and Technology