NYCER: A Non-Emergency Response Predictor for NYC using Sparse Gaussian Conditional Random Fields

David DeFazio, Arti Ramesh, and Anand Seetharam
Department of Computer Science, SUNY Binghamton
{ddefazi1, artir, aseethar}@binghamton.edu

ABSTRACT

Cities have limited resources that must be used efficiently to maintain their smooth operation. To facilitate efficient resource allocation and management in cities, in this paper, we study one such important problem: how long does it take to resolve non-emergency 311 service requests? We present *NYCER*, a Non-emergency Response prediction system based on a recently developed structured regression model, sparse Gaussian conditional random fields (GCRFs), that successfully captures the dependencies between historical and future response times. Through extensive experimentation on 311 service requests in New York City (NYC) over a three and a half year period between Jan 2015 to June 2018, we demonstrate that our trained system is able to accurately predict future response times one week in advance using just the previous two weeks data at test time. NYCER achieves superior prediction performance across all agencies, complaint types, and locations, when compared to a linear regression baseline (up to a factor of 2X). The trained NYCER system requires low computational resources and data at test time, thus making it an attractive system that can be readily deployed in practice.

ACM Reference Format:
David DeFazio, Arti Ramesh, and Anand Seetharam Department of Computer Science, SUNY Binghamton {ddefazi1, artir, aseethar}@binghamton.edu. 2018. NYCER: A Non-Emergency Response Predictor for NYC using Sparse Gaussian Conditional Random Fields. In *Proceedings of EAI (MobiQuitous'18).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Non-emergency helpline numbers have gained prominence in the last two decades and have been widely adopted by major cities in various countries around the world [8]. In both USA and Canada, this special non-emergency telephone number is 311. New York City (NYC) implements one of the largest 311 operations in USA, which began in 2003. To enable the government to run these services efficiently, all data related to 311 service requests from 2010 have been made publicly available and is updated daily [1, 6].

In this paper, we design a non-emergency response time prediction system, *NYCER*, that adopts a data-driven approach to predict the service request response times on a weekly basis (i.e., predicting the response times at the beginning of a week for the entire week). Predicting the response times for requests ahead of time is beneficial for performing efficient resource allocation (e.g., personnel) to address different types of requested services. Higher than usual predicted future response times can also be utilized by agencies to determine the number of additional temporary contractors that need to be hired for the coming week so as to keep the actual response times within a desired threshold.

Response times depend on a number of factors such as demand, agency the complaint is assigned to, type of complaint, and the location of the complaint. NYCER takes these factors in account and leverages past 311 data to make decisions related to future response times. NYCER consists of two stages: i) an initial data pre-processing stage that removes noisy data and extracts relevant data based on start and close date of the complaint, the agency, the complaint type, and the location, and ii) the core second stage that consists of the proposed prediction models that take this filtered input and produce the desired output (i.e., predicted response times). We create separate data sequences and develop models for each agency, complaint type, and location, as then these models can be executed independently by the responsible agencies, county/city officials to estimate and plan for their resource needs.

Specifically, our main contributions are as follows:

(1) We design NYCER, a system that leverages and adapts sparse Gaussian Conditional Random Fields (GCRFs) to predict future response times based on historical data. We develop two GCRF models: *i)* GCRF Response, which uses past response times to predict future response, and *ii)* GCRF Demand, which uses past response and demand to predict future response times. The proposed GCRF models are computationally efficient and consider only minimal past information (past 2 weeks) at test time. We demonstrate that the parsimonious GCRF Response model that only relies on past response times to predict future response times provides performance similar to and in some cases outperforms a more complex GCRF Demand model that considers both past demands and response times. Furthermore, the sparse nature of our models aids in learning only those dependencies among the observed features and target variables that are helpful in the prediction, thus ensuring that the model has the ideal and required amount of complexity.
(2) We conduct extensive experiments on NYC 311 service request data from January 2015 to June 2018 to demonstrate the efficacy of our models. We train separate models for each of the different factors that affect response times: i) agency, ii) complaint type, and iii) location. This helps us reduce the uncertainties in prediction imposed by high variance in the

MobiQuitous'18, November 2018, New York, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

data. This separation also ensures that the models can be executed independently, thus aiding responsible agencies and city/county officials to make important resource allocation decisions based on their respective historical response times. Our experiments demonstrate that NYCER significantly outperforms a baseline linear regression model both in terms of Root Mean Squared Error (RMSE) and Relative Error (RE) across all agencies, complaint types, and locations, yielding up to a factor of 2X improvement in prediction performance.

(3) We also conduct experiments by training on data sequences of varying lengths of 1, 2, 4, 6, and 8 weeks and find that training on 2-week length sequences achieves a good prediction performance at test time, comparable to or better than training on longer data sequences. Consequently, our model is faster to train and at test time only requires 2 weeks of past data to predict future response times for 1 week.

Our results and analysis indicate that the trained NYCER requires limited data and computational resources at test time, thus making it a practically useful system.

2 RELATED WORK

In this section, we provide an overview of related work. We first describe prior work related to 311 service request data and then discuss prior work related to the application of machine learning and statistical models to solve problems in the pervasive and ubiquitous computing domain.

Most prior work related to non-emergency services focus primarily on noise-related complaints [3, 13, 26]. Zheng et al. [26] develop a 3-dimentional tensor to detect noise using a combination of 311 service requests and social media data. In [13], the authors design a mobile collaborative social application for urban noise monitoring, while a system for monitoring, analyzing and mitigating urban noise pollution is proposed in [3]. Wang et al. [11] and Zha et al. [25] analyze 311 requests and use simple gradient boosting regression and random forest models to predict the number of 311 requests, respectively. The authors in [20] use 311 service requests to understand local urban context. In contrast to prior work that focus primarily on developing systems for data analysis and clustering, our goal in this paper is to design a prediction system using GCRFs [23] to predict future response times for 311 service requests.

In recent years, a myriad of machine learning techniques have been applied in various urban and ubiquitous computing contexts toward creating smart citites and societies [2, 10, 14, 17, 19, 21]. In [4, 21] the authors design spatial analysis and auto-regressive models to detect high-risk crime regions and to reliably forecast crime trends, and to predict pedestrian foot traffic in urban areas, respectively. Ghosh et al. [5] develop a mixed integer linear programming formulation to enable dynamic repositioning of bike systems in cities. Kang et al. [9] and Mittal et al. [14] design deep learning models for prediction of crime occurrence from multi-modal data and spotting garbage from images, respectively. Papangelis et al. [16] and Yabe et al. [24] model mobility and traffic flow aiding in better utilization of city resources and an easy commute.

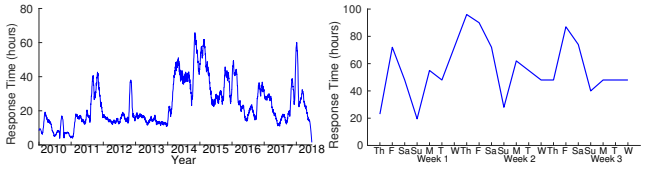
Our proposed system falls under this broad umbrella of designing machine learning systems for ubiquitous and smart computing. Our work primarily differs from prior work in that it leverages sparse GCRFs, a recently proposed and state-of-the-art structured prediction model particularly suited for regression analysis. Further, we design models to predict future response time for 311 service requests, a problem that has not been addressed in previous research. We also evaluate the efficacy of our models and relationship among different spatio-temporal factors affecting response times through extensive experimentation.

3 DATA & PROBLEM STATEMENT

3.1 Data

We use the data collected and distributed by NYC open data [1], a site that provides 311 complaints since 2010, updated daily. This data contains 17, 945, 594 rows, and 41 different attributes providing extensive information associated with each complaint. For the purposes of our study, we select the following attributes to work with:

(1) *Created Date* - Specifies the date and time the complaint was created.
(2) *Closed Date* - Specifies the date and time the complaint was closed.
(3) *Agency* - Specifies the responding city government agency.
(4) *Complaint Type* - Identifies the topic of the complaint.
(5) *Location* - Specifies the borough of the complaint.



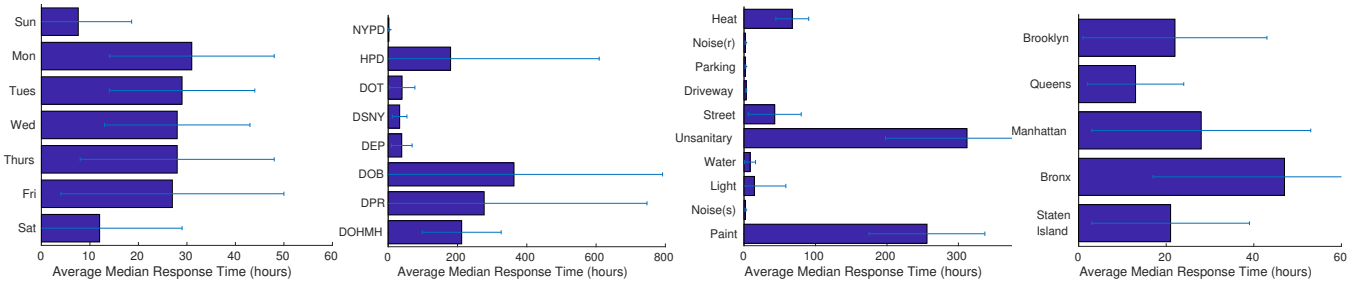
(a) Median response time per day. This line was smoothed, in order to better visualize the yearly trends.

(b) Response time from January 1st 2015 to January 21st 2015. Reponse time dips on weekends, and peaks on week days.

Figure 1: Response time across years and weeks

Response times are calculated by subtracting *Created Date* from *Closed Date*. Figure 1a shows the median response time per day, across all years of data. The data has been smoothed, in order to better visualize the long term trend. Figure 1b is a zoomed in view of the actual data, across a 3 week period. Note that there is a weekly trend, where median response time dips during weekends and peaks during week days. While long term seasonality in the data exists, in this work, we are particularly interested in learning the short-term weekly structure. Figure 2a identifies a recurring pattern across weeks, confirming the pattern we see in Figure 1b. This temporal structure of peaks and dips present in median response time is what we want our model to capture and predict.

Figures 2b, 2c, and 2d present the average median response times for each agency, complaint type, and location, respectively. These agencies, complaint types, and locations are sorted on the total number of complaints received by them. These figures illustrate that the median response time per day varies significantly across different agencies, complaint types, and locations. Tables 1a, 1b,



(a) Day of week **(b) Agency** **(c) Complaint** **(d) Location**

Figure 2: Average median response time based on day of week, agency, complaint type, and location

Agency	Abbreviation	Complaints Addressed	Percentage of total complaints
New York Police Department	NYPD	Noise, illegal parking, drug activity, ...	29.7%
Housing Preservation and Development	HPD	Heat/hot water, plumbing, electric, ...	26.2%
Department of Transportation	DOT	Broken parking meter, highway condition, ...	12.6%
Department of Sanitation	DSNY	Overflowing trash, collection truck noise, ...	9.4%
Department of Environmental Protection	DEP	Air quality, industrial waste, ...	8.3%
Department of Buildings	DOB	Plumbing, boilers, damaged tree, ...	3.8%
Department of Parks and Recreation	DPR	Sidewalk damage, dead tree, park rule violations, ...	3%
Department of Health and Mental Hygiene	DOHMH	Smoking, rodents, mosquitoes, ...	2%

(a) Agency Information

Complaint type	Abbreviation	Description	Percentage of total complaints
Heat/Hot Water	Heat	Heat and/or hot water does not work	10.1%
Noise - Residential	Noise(r)	Loud neighbors, parties	9.7%
Illegal Parking	Park	Blocking vehicle, traffic, sidewalk, ...	5.5%
Blocked Driveway	Driveway	Blocking someones driveway	5.3%
Street Condition	Street	Various types of street damage	4.6%
Unsanitary Condition	Unsanitary	Sewage, mold, bugs, ...	3.5%
Water System	Water	No water, fire hydrant issues, ...	3.1%
Street Light Condition	Light	Street light dead or damaged	3%
Noise - Street/Sidewalk	Noise(s)	Loud talking or music	2.7%
Paint/Plaster	Paint	Peeling paint from walls	2.7%

(b) Complaint Type information

Location	Percentage of total complaints
Brooklyn	31.4%
Queens	23.8%
Manhattan	21%
Bronx	18.8%
Staten Island	5%

(c) Location Information

Table 1: Description of different agencies, complaint types, and locations in 311 service requests.

and 1c give a description of the top 8 agencies, 10 complaint types, and all locations in the data and give the percentage of complaints in each of them. The percentage of total complaints is calculated as the number of complaints for that agency, complaint, or location, divided by the number of complaints, with a valid closed date and positive response time.

3.2 Problem Statement

Implementing an effective non-emergency response system in a large city such as New York City (NYC) is a challenging endeavor, due to the sheer volume of complaints and uncertainty associated with how, when, and where people can lodge a complaint. Allocating sufficient resources ahead of time is crucial in improving response times of different types of complaints. A system that predicts future response times can thus be beneficial in understanding where resources can be better allocated. Recognizing the short-term trends in median response time, along with variation in average median response time based on factors such as agency, complaint type, and location, we observe that response time has an underlying structure that can be captured in a predictive model. *In this work, we develop a predictive modeling system using a recently developed structured regression graphical model, sparse GCRFs to predict future median response time based on past median response time.*

4 NYCER: A NON-EMERGENCY RESPONSE PREDICTION SYSTEM

In this section, we provide an overview of our non-emergency response prediction system, NYCER. Figure 3 shows the different components of our system. NYCER takes 311 service request data

as input and produces predicted response times for the future as output. NYCER comprises of two main components: *i)* the data pre-processing component, which pre-processes the 311 service request data with respect to a number of factors such as agency and complaint type to produce processed data, and *ii)* the prediction component consisting of the GCRF models that takes the pre-processed data to generate the desired predictions. Figures 4 and 5 illustrates the data pre-processing and the model components in more detail, respectively. We explain these components in the following sections.

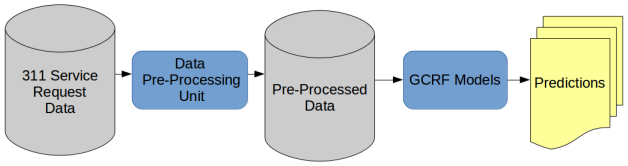


Figure 3: NYCER Architecture

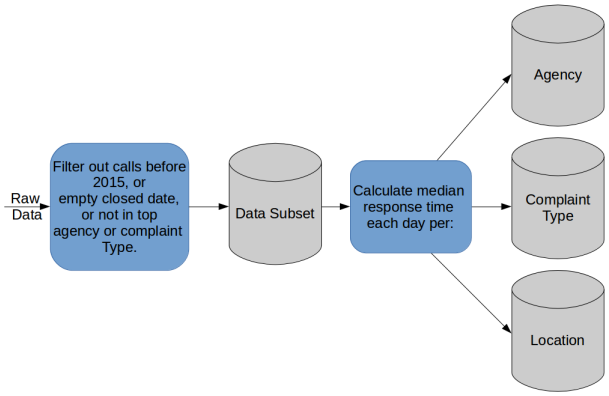


Figure 4: NYCER Architecture of Data Pre-Processing Component. Raw data is filtered, and cloned to 3 different datasets, one each for agency, complaint type, and location.

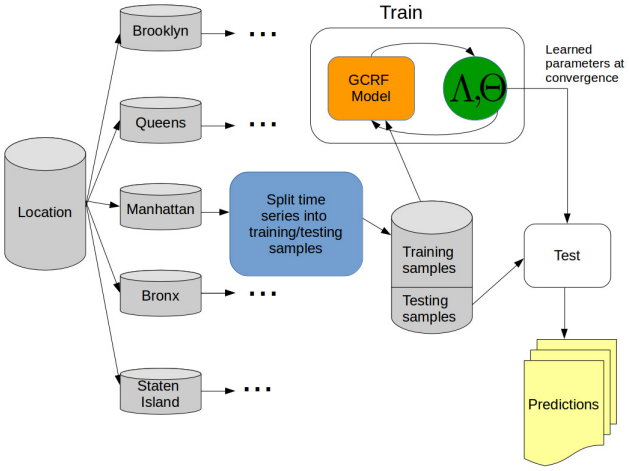


Figure 5: NYCER Architecture of GCRF Models Unit. For each time series within each dataset, we produce training and testing samples, which a GCRF model will use to learn its parameters, and generate predictions respectively.

4.1 Data Pre-processing

In our models and analysis, we consider response data recorded between January 2015 to June 2018. We consider the day of the service request to be the day listed in *Created Date*. Complaints with no closed date make up 3.2% of the requests after 2015 and are filtered out. The dataset includes 261 different possible complaint types and 29 different possible agencies after 2015. Upon performing a closer analysis, we find that a majority of complaints fall under a smaller number of complaint types, and get a response from a smaller number of agencies. The top 8 agencies account for 93.8% of the data after 2015, while the top 32 complaint types account for 82.4%. In this study, we only consider 31 out of 32 of the most common complaint types after 2015, and the 8 most common responding agencies. We do not consider the 17th most popular complaint type after 2015—*Request Large Bulky Item Collection*, because this is a relatively new complaint, with its first request on August 29th, 2017. Since we do not have data for this complaint type for the years 2015 and 2016, we exclude it from our analysis. After filtering out requests with empty *Closed Date*, unspecified boroughs, and service requests before 2015, we find that the 8 most popular agencies and 31 complaint types, make up 80.8% of the filtered data.

We filter out 311 service requests based on the following criteria:

(1) Each request must have a valid *Created Date* and *Closed Date*.
(2) Each request must get a response from one of the top 8 *agencies*.
(3) Each request must have one of the top 32 *complaint types*, excluding *Request Large Bulky Item Collection*.
(4) Each request must have one of the 5 *locations* of NYC specified.
(5) Each request must be from 2015 or later.
(6) Subtracting the *Created Date* from the *Closed Date* must produce a positive value.

After this filtering, we are left with 6, 405, 589 calls. Next, we produce 3 different time series datasets, all consisting of the same set of data points. Each of these datasets aggregate the number of service requests per day, and take the median of those requests' response times per day. The different datasets are created based on the responding agency, the complaint type, and the location of the service request. We create one time series with median response times per day and another with median demand per day for each agency, complaint type, and location. Thus, our agency dataset has 16 time series (8 agencies, 2 for each agency), complaint type dataset has 62 times series (31 complaint types, 2 for each complaint type), and location dataset has 10 time series (5 locations, 2 for each location). Creating these seperate time series not only decreases the variance, when compared to the median response times of the entire data, but also allows different NYC responding agencies to run the model only on data applicable to them.

4.2 GCRF Prediction Model

When modeling short-term response times, one must consider how to capture dependencies between the historical and predicted response times, and how these dependencies can be used to improve prediction performance. Figures 1b and 2a show a repeated pattern in the data, where response time dips during the weekend,

and peaks during the week. This clearly shows that response time itself is not independent and identically distributed, and that its structure must be incorporated in our model. This inspired us to consider a probabilistic graphical modeling approach, as graphs can be constructed to represent this structure in the data.

The generative approach of explicitly encoding all dependencies among the inputs in the graphical model often times leads to overfitting and poor prediction performance as the performance hinges on effectively capturing these dependencies. Conditional Random Fields (CRFs) [12] belong to the discriminative class of models, which model the output variables given the input features and do not explicitly encode the dependencies among the inputs. Instead, the dependencies that CRFs model are edges among the output variables in the graphical model, and edges between the input and output variables. This is helpful, as it avoids making an often incorrect independence assumption that models such as Naïve Bayes [15] encode, or incorrect dependence assumptions among features such as Gaussian Markov Random Fields [18].

Since capturing the precise dependence among input features is a difficult and time-consuming task, in our problem, instead of modeling the joint distribution of both observed features (historical response times) and target variables (future response times), we only model the posterior distribution of the future response times given the historical response times. As mentioned above, CRFs are a great choice for this problem as they accomplish what we need with minimal assumptions. Other well studied time series approaches also exist, such as LSTM (long short-term memory) [7]. We chose GCRF instead of LSTM, as GCRFs learn dependencies that can be interpreted easily as opposed to LSTMs, where the dependencies are harder to interpret. We choose a recent version of CRF, extended to structured regression, sparse GCRFs [23], for predicting future response times. In the following section, we give an overview of GCRFs and show how to adapt them for response prediction.

4.2.1 Gaussian Conditional Random Fields (GCRFs). The posterior distribution modeled by GCRFs is as follows [23]:

$$P(y \mid x; \Lambda, \Theta) = \frac{1}{Z(x)} exp(-y^T \Lambda y - 2x^T \Theta y) \tag{1}$$

where $x = [x_1, x_2, ..., x_n]$ represent median historical response times per day, where n is the number of days in the past and $y = [y_1, y_2, ..., y_m]$ represent median predicted response times per day, where m indicates the number of days in the future. Θ and Λ are parameters/regression coefficients of the GCRF model. Θ is an nxm matrix, containing the edges between x and y, while Λ is the mxm inverse covariance matrix, containing the edges amongst the y's. The CRF is a Gaussian distribution with mean $-\Lambda^{-1}\Theta^T x$ and variance Λ^{-1}, $\mathcal{N}(-\Lambda^{-1}\Theta^T x, \Lambda^{-1})$. $Z(x)$ in Equation 1 is the partition function, which ensures that the posterior is integrated to 1. This is necessary, because edges in the graph can be represented by any real number.

4.2.2 GCRF Training. At training time, we find the most likely estimates of the parameters Λ and Θ given the training data. We perform maximum likelihood estimation of the parameters by finding the parameter values that maximize the probability of the data given the parameters (also known as the likelihood function), as shown below.

$$\max_{\Lambda,\Theta} P(y \mid x; \Lambda, \Theta) \tag{2}$$

This is equivalent to minimizing the negative log likelihood, given by:

$$\min_{\Lambda,\Theta} -log(P(y \mid x; \Lambda, \Theta)) \tag{3}$$

Regularization is a way to avoid overfitting by penalizing high-valued regression coefficients. In order to make sure that our models generalize better, we add a regularization term to the maximum likelihood estimate equations above. L_1 and L_2 are two popularly used regularization norms that add a penalty term corresponding to the absolute value of the magnitude of the coefficients and square of the magnitude of the coefficients, respectively. The total number of parameters in this problem for n historical time steps given by x and predicting m future time steps for y is $nm + \frac{m(m+1)}{2}$, where nm edges are given by Θ, and $\frac{m(m+1)}{2}$ by Λ. Since we are only predicting a week ahead, the value of m is 7. For this small value of m, it is possible that the model can overfit due to the large number of parameters. To overcome this, we use L_1 regularization as it leads to a sparser set of regression coefficients by driving the less contributing coefficients to zero.

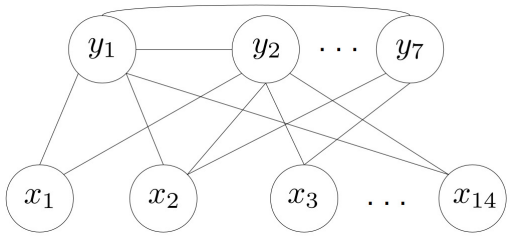


Figure 6: GCRF Response: graphical model showing connections between historical response times, $x_1, ..., x_{14}$ and $y_1, ..., y_7$, and among $y_1, ..., y_7$. Note that our model is sparse, learning only edges between variables that matter; we illustrate this phenomenon by leaving out some edges in the graphical model.

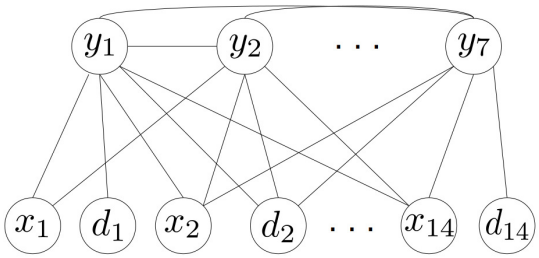


Figure 7: GCRF Demand: graphical model showing connections between historical response times, $x_1, ..., x_{14}$, historical demands, $d_1, ..., d_{14}$, and $y_1, ..., y_7$, and among $y_1, ..., y_7$. Note that our model is sparse, learning only edges between variables that matter; we illustrate this phenomenon by leaving out some edges in the graphical model.

After expanding $P(y \mid x; \Lambda, \Theta)$, and adding L_1 regularization, the optimization problem is as follows:

$$\min_{\Lambda,\Theta} -log|\Lambda| + \frac{1}{k} tr(Y^T Y\Lambda + 2Y^T X\Theta + \Lambda^{-1}\Theta^T X^T X\Theta\Lambda^{-1}) + \lambda(||\Lambda|| + ||\Theta||) \quad (4)$$

where k is the number of samples in X and Y. The last term $\lambda(||\Lambda|| + ||\Theta||)$ gives the regularization term, where λ is the regularization constant. We use the optimization method proposed by Wytock et al. [23] for solving the GCRF with L_1 regularization. They propose a second-order active set method, that iteratively produces a second-order approximation to the objective function without the L_1 regularization term, and then solve the L_1 regularized objective function using alternating Newton coordinate descent. For additional details, we refer the reader to [23].

4.2.3 GCRFs for Non-emergency Response Prediction. We design two prediction models: i) GCRF Response, which takes only past response time data, and ii) GCRF Demand, which takes past response time and demand data, to predict future response times.

Figure 6 shows our GCRF Response model for one 3-week data sequence. Here, past median response times $x_1, ..., x_{14}$ have edges to predicted median response times $y_1, ..., y_7$, where each edge represents how much the variables influence each other. Note that we start with a fully-connected graph between x and y and between y's. The incorporation of the L_1 regularization norm can drive some of these edge values to zero, thus yielding a sparser graph after training. Note that in Figure 6, some edges between x and y have been left out to illustrate the sparser nature of the graphical model learned after training.

Figure 7 shows the GCRF Demand model for one 3-week data sequence, in which we include historical demand (d_1, d_2, ... d_{14}) and historical response times to predict future response times. The GCRF Demand model is a more complex model having an additional nm parameters. We implement our models using *SGCRFPy*, a Python toolkit for sparse GCRFs[1]. We will make our system available as open source software.

5 IMPLEMENTATION DETAILS

The primary objective of NYCER is to accurately predict future response times for 7 days based on previous 14 days data. To enable the proposed GCRF models (i.e., GCRF Response and GCRF Demand) used in NYCER to make accurate predictions, we generate sequences containing t weeks as input, and 1 week as output. We adopt a sliding window approach that moves the window one day at a time to cover the entire time series data, generating a sequence of length $t + 1$ weeks each time the window slides. The default value of t used in our experiments is 2.

To enable NYCER to provide accurate fine-grained predictions with respect to agency, complaint type, and location, we train separate GCRF models on data filtered according to these different attributes. We compare the performance of NYCER to a baseline system that uses linear regression to make predictions. The main metric used for evaluation is the root mean squared error (RMSE). We report RMSE results for predicting future median response time given the past median response time for each agency, complaint type, and location. We report two different values of RMSE: i) for each predicted day separately (i.e., day 1 to day 7), as given by Equation 5 and ii) across all days, as given by Equation 6. In addition to the RMSE, we also investigate the relative error (RE) in prediction, which captures the fraction of error in the predicted response with respect to the actual response (Equation 7).

$$RMSE_{day_j} = \sqrt{\frac{\sum_{i=1}^{h}(\hat{y}_{ij} - y_{ij})^2}{h}} \quad (5)$$

$$RMSE = \frac{\sum_{j=1}^{m} RMSE_{day_j}}{m} \quad (6)$$

$$RE_{day_j} = \frac{\sum_{i=1}^{h} \frac{|\hat{y}_{ij} - y_{ij}|}{y_{ij}}}{h} \quad (7)$$

where y_{ij} is the i^{th} test sample for the j^{th} day, $\hat{y}_{ij}$ is the predicted value of y_{ij}, and h is the number of test samples.

5.1 Training

At training time, our GCRF models use the first 75% of the sequences generated from the sliding window, which have t weeks input and 1 week output. The parameter Λ is initialized to the identity matrix and Θ is initialized to all zeros. In total, GCRF Response has $49t$+28 parameters while GCRF Demand has $98t$+28 parameters. Even with our default value of t=2, both GCRF models have significant number of parameters making L_1 regularization necessary. Using regularization constant $\lambda = 0.1$, and 10,000 iterations, we converge on a set of dependencies learned from the training data. For each agency, complaint type, and location, we train three GCRF Response models, and three GCRF Demand models, due to the randomness involved with the coordinate descent optimization.

5.2 Testing

At test time, our GCRF models use the remaining 25% of sequences from the sliding window. To generate predictions using our GCRF models, we sample from the normal distribution with mean $-\Lambda^{-1}\Theta^T x$. Since we already learned the parameters Λ and Θ, we simply need to use them in the first t weeks of each test sequence to generate a prediction for 1 week. The predictions of the three learned GCRF models are then averaged to produce a final prediction. This prediction is tested against the ground truth response times and the metrics RMSE and RE are calculated.

Our linear regression baseline model does not need the first 75% of training sequences, as it simply produces a best fit straight line on the first t weeks of each test sample, and predicts the next week based on the line learned for that sample.

6 PERFORMANCE EVALUATION

In this section, we present experimental results that demonstrate the superior predictive performance of NYCER. We first compare RMSE results averaged over the entire predicted week for GCRF Response and GCRF Demand with the baseline linear regression models and demonstrate that GCRF models achieve superior prediction performance. We then delve deeper and present fine-grained performance results with respect to each agency, complaint type,

[1]Sparse GCRF implementation: https://github.com/dswah/sgcrfpy.

Agency	GCRF Response	GCRF Demand	Linear Regression
NYPD	0.38	0.42	0.42
HPD	37.25	36.32	44.78
DOT	16.05	15.45	22.76
DSNY	10.25	10.34	15.85
DEP	17.78	17.59	30.07
DOB	94.24	142.28	115.38
DPR	105.41	187.73	113.48
DOHMH	49.89	49.57	63.89

Table 2: RMSE comparison across agencies

Complaint Type	GCRF Response	GCRF Demand	Linear Regression
Heat	15.09	14.02	23.65
Noise(r)	0.28	0.33	0.31
Park	0.61	0.64	0.72
Driveway	0.57	0.63	0.71
Street	21.74	21.42	33.02
Unsanitary	77.53	75.31	79.65
Water	6.38	6.37	8.19
Light	19.38	17.79	32.26
Noise(s)	0.55	0.56	0.66
Paint	71.35	69.85	78.82

Table 3: RMSE comparison across complaint types

Location	GCRF Response	GCRF Demand	Linear Regression
Brooklyn	16.33	16.01	21.93
Queens	9.99	9.77	13.62
Manhattan	16.21	15.75	22.11
Bronx	22.53	21.79	31.84
Staten Island	19.1	19.54	26.46

Table 4: RMSE comparison across locations

and location. Then, using relative error of predictions, we demonstrate that our model is able to learn the nuances in the data resulting in the predicted response times to be close to the actual response times. We then study the impact of training on different data sequence lengths on the predictive performance of the models and observe that training on data sequences of length 2 weeks of historical response time is sufficient to accurately predict future response times. We conclude this section with qualitative results comparing the predicted and actual response times.

Our experiments show the efficacy of the proposed models, with both GCRF Response and GCRF Demand consistently outperforming the baseline linear regression model. Additionally, our experiments show that the GCRF Response model achieves performance comparable to, and in some cases better than GCRF Demand, thus demonstrating that a simple parsimonious model can elegantly capture the subtle interdependencies in the data. Our results and analysis ascertain that the NYCER system can be trained quickly, and once trained requires low computational and data resources at test time, thus making it an attractive system that can be easily deployed in practice.

6.1 Average RMSE Results

In this subsection, we report the average RMSE results for the entire predicted week (Equation 6). From Figures 2a–2d, we observe that response times across agencies, complaint types, and locations have a high variance. Due to this reason, we train separate models for each agency, complaint type, and location. This also makes it possible for training and testing our models individually to efficiently manage resources. Tables 2, 3, and 4 show the average RMSE results with respect to each agency, complaint type, and location, respectively. We observe that the proposed GCRF Response and the GCRF Demand models consistently outperform the linear regression model by a significant margin. We observe that GCRF Response outperforms linear regression for all agencies, complaint types, and locations, while GCRF Demand outperforms linear regression for all agencies, complaint types, and locations, expect for the DOB and DPR agencies and the Noise-Residential complaint type. Comparing GCRF Response and GCRF Demand, we observe that both models provide comparable performance overall, with GCRF Response significantly outperforming GCRF Demand for DOB and DPR agencies. We hypothesize that the correlation between demand and response is the primary reason behind the more complex GCRF Demand model providing performance similar to the simpler GCRF Response model.

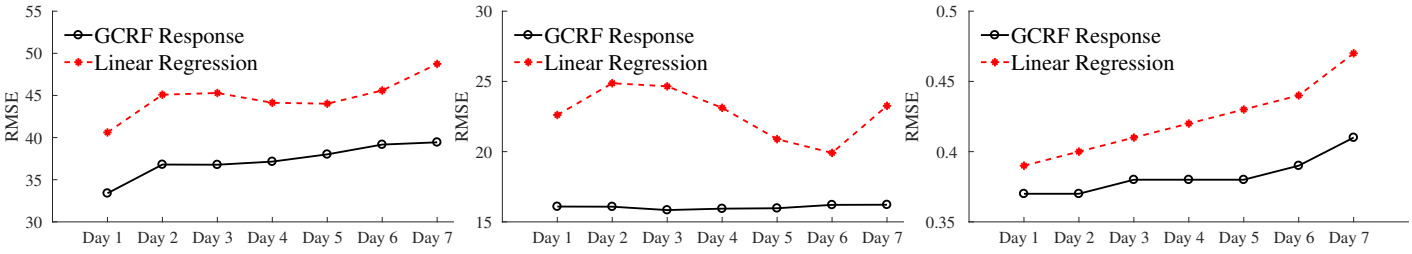
The results in Tables 2, 3, and 4 indicate that the parsimonious GCRF Response model is able to successfully capture the subtle interdependencies in the data. Based on these results, we recommend adopting the GCRF Response model in NYCER for performing all required predictions. Therefore, in the following sections, we only present results for the GCRF Response model and compare it to the linear regression model.

6.2 Agency/Complaint Type/Location-wise Performance

In this subsection, we investigate the performance of the GCRF Response model with respect to agency, complaint type, and location, per day of prediction by calculating RMSE per day (Equation 5). Figures 8, 9, and 10 show the RMSE values for each of the seven days for each agency, complaint type, and location, respectively.

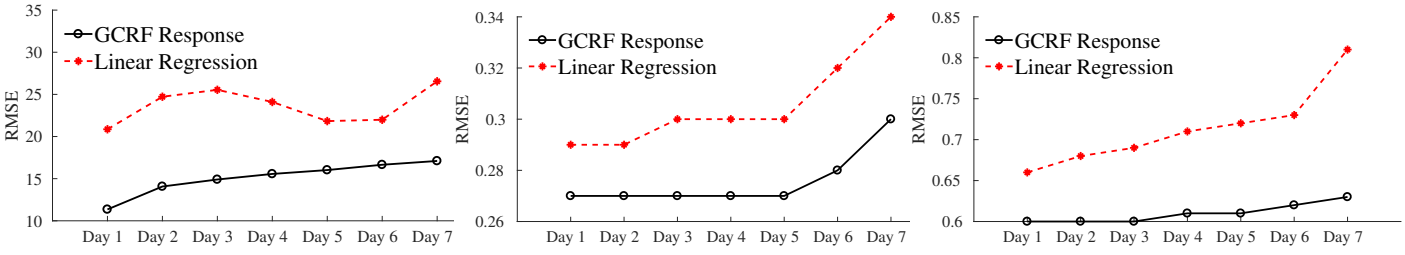
We observe from these figures that the GCRF Response model outperforms linear regression across agencies, complaint types, and locations in per day prediction as well, particularly achieving a significantly better performance further in the prediction sequence. The main reason behind this behavior is that the GCRF Response model learns the trend from historical responses, and is capable of predicting the weekly peaks and dips in response time, whereas linear regression can only produce a best fit straight line. The GCRF Response model learns these trends by finding the strength of dependencies between historical and estimated response times, along with dependencies amongst the predictions, that maximize the likelihood of the data.

We observe that the RMSE generally increases for both models as they predict further into the future. This is understandable because predicting further ahead into the future is usually more challenging. Interestingly, we observe that in comparison to linear regression, the increase in RMSE for the GCRF Response model is more gradual



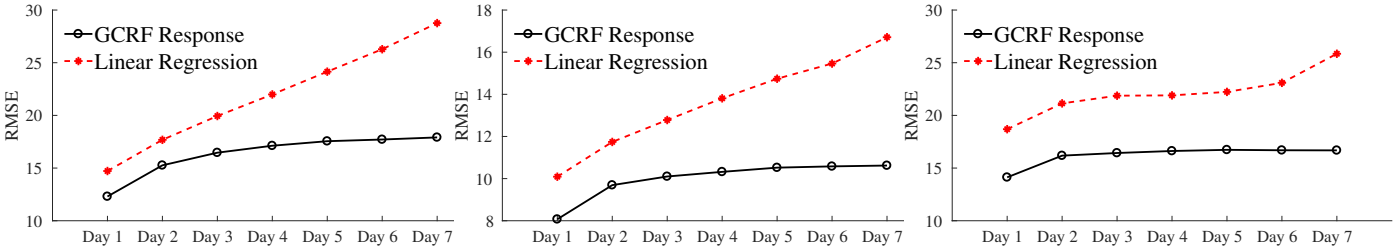
(a) HPD (b) DOT (c) NYPD

Figure 8: Agency: Average RMSE comparison for each day in the prediction sequence



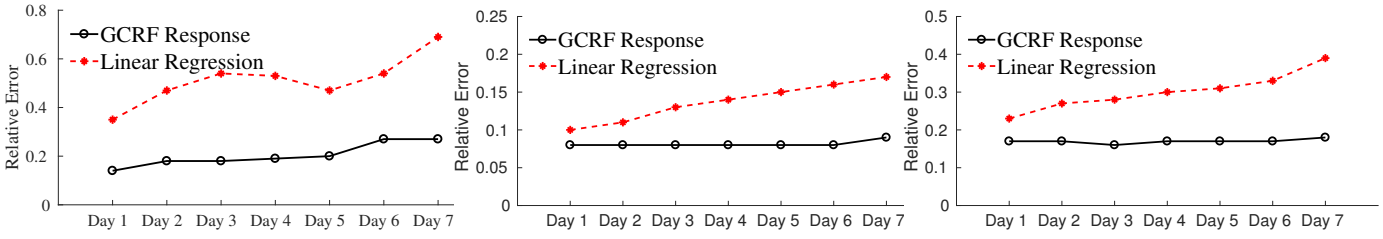
(a) Heat (b) Noise(r) (c) Park

Figure 9: Complaint Type: Average RMSE comparison for each day in the prediction sequence



(a) Brooklyn (b) Queens (c) Manhattan

Figure 10: Location: Average RMSE comparison for each day in the prediction sequence



(a) Heat (b) Noise(r) (c) Park

Figure 11: Complaint Type: Relative error for each day in the prediction sequence

indicating the superiority of the proposed model. This effect is most prominent in Figure 10.

Note that each agency, complaint type, and location have varying scales of response times. For example, NYPD has a much shorter response time than HPD. Therefore, it is expected that the absolute value of RMSE will be lower for NYPD than for HPD. For this reason, each graph has differing scales.

6.3 Relative Error

While RMSE comparisons between GCRF and linear regression provide valuable insight, studying relative error is important to ensure that GCRF predictions are within an acceptable and useful range of the actual values. Figure 11 shows the relative error for both models for the top 3 complaint types. We observe that the GCRF Response model outperforms linear regression with each of the top 3 complaint types lying within 27% of the truth, even at

Agency	1 week	2 weeks	4 weeks	6 weeks	8 weeks
NYPD	0.39	0.38	0.38	0.34	0.23
HPD	37.74	37.25	37.52	37.83	38.96
DOT	17.29	16.05	15.35	15.74	15.56
DSNY	11.43	10.25	9.76	9.67	9.67
DEP	19.4	17.78	17.39	16.86	16.53
DOB	98.69	94.24	76.76	74.33	73.98
DPR	115.48	105.41	116.38	104.84	112.05
DOHMH	50.57	49.89	49.51	48.87	49.02

Table 5: Agency: GCRF Response RMSE for different sequence lengths

Complaint	1 week	2 weeks	4 weeks	6 weeks	8 weeks
Heat	15.67	15.09	14.75	15.14	15.27
Noise(r)	0.28	0.28	0.27	0.26	0.23
Park	0.62	0.61	0.61	0.57	0.46
Driveway	0.59	0.57	0.58	0.56	0.5
Street	22.46	21.74	21.17	21.28	21.25
Unsanitary	74.53	77.53	79.2	81.13	84.59
Water	6.54	6.38	6.53	6.75	6.71
Light	22.46	19.38	17.72	17.21	16.81
Noise(s)	0.56	0.55	0.54	0.53	0.53
Paint	68.89	71.35	72.77	71.96	73.32

Table 6: Complaint Type: GCRF Response RMSE for different sequence lengths

Location	1 week	2 weeks	4 weeks	6 weeks	8 weeks
Brooklyn	16.15	16.33	15.95	16.07	16.46
Queens	9.96	9.99	10.42	10.6	10.98
Manhattan	16.07	16.21	15.89	16.31	16.38
Bronx	23.27	22.53	22.19	22.27	22.09
Staten Island	19.85	19.1	20.71	21.68	22.16

Table 7: Location: GCRF Response RMSE for different sequence lengths

7^{th} prediction step. In comparison, the worst case performance for linear regression is 69%. From the figure, it is evident that GCRF response only slightly increases in relative error by day 7, while linear regression has significantly more difficulty predicting 7 days ahead. This once again highlights how capturing the data's underlying structure in a model can yield improved performance over a sustained period of time.

We note that the GCRF Response model significantly outperforms linear regression for all other complaint types as well. Interestingly, in our experiments, we observe that certain complaint types, such as Light and Water, have higher errors for both models. For example, analyzing Light complaint type, we observe that the response time is 0 when the detailed complaint type information is "Street Light Out". The presence of zeroes adds to the variance of this complaint type and makes it difficult for both models to predict response times accurately.

6.4 Discussion on Sequence Length

Tables 5, 6, and 7 show the impact of training the GCRF Response model on sequences of varying lengths. Increasing the sequence length beyond two weeks has limited impact on the performance of the GCRF Response model. In fact, for the GCRF Response model, on average, the RMSE decreases by only 8.4% across agencies, 5.6% across complaint types, and increases 5.2% across locations, when given 8 weeks as input per sequence rather than 2 weeks. This shows that GCRF response can extract the short term trends of the data given just a few weeks, and longer sequence lengths only reinforce the structure that was already learned. We observe that though linear regression has substantially more improvement when given longer sequences, the results are still inferior to GCRF response, with only DPR and DOB beating GCRF across all agencies, complaint types, and locations, when given 8 week sequence lengths.

6.5 Qualitative Results

In this subsection, we present qualitative results to illustrate the predictive performance of the GCRF Response model. Figures 12 and 13 show the one step and seven step predictions of our model for the top agency, complaint type and location for the entire test data, respectively. Our qualitative results highlight two key points.

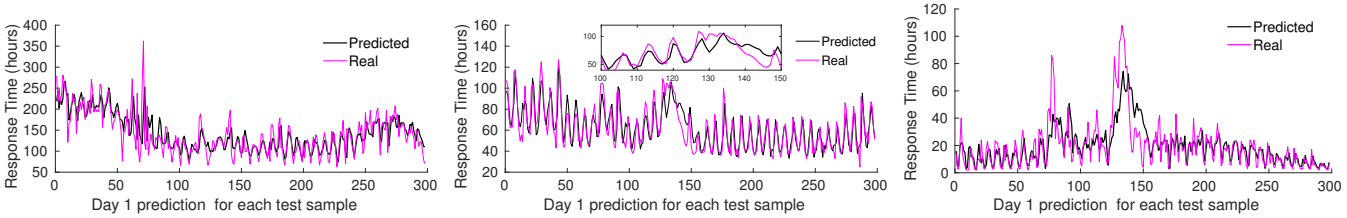
(1) GCRF Response, while parsimonious, also has enough complexity for robust time series predictions. The figures clearly show that GCRF Response predictions do not put too much weight on the previous time step's real value, a common indication of a weak prediction model.
(2) Day 7 GCRF Response predictions continue to capture the underlying structure in the data, and provide meaningful predictions, even 7 days in advance.

It is also worth noting that our predictions for Heat are extremely close to the actual median response times. We hypothesize that this is because complaint types in general have lower variances in response times than agencies and locations, as each agency responds to multiple complaint types, and each location has multiple agencies and complaint types.

7 CONCLUSION AND FUTURE WORK

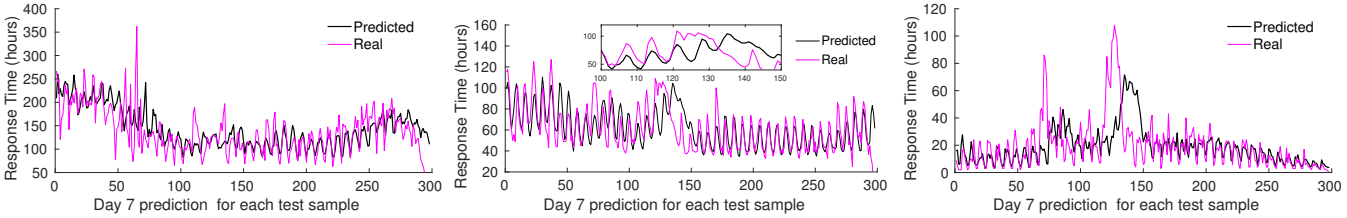
In this paper, we designed NYCER, a system for predicting future response times of non-emergency services based on historical data. NYCER uses GCRF based models at its core to elegantly capture dependencies between historical and future response times so as to accurately predict response times. We compared NYCER with a baseline system that uses Linear Regression and observed that our system significantly outperformed the baseline system both in terms of RMSE and relative error. Based on our experiments, we observed that training the GCRF models on sequences with two weeks of historical median response times is sufficient for accurately predicting the next week's median response times.

Our investigation opens up several avenues for future research. An immediate extension is to elicit the scalability of NYCER by conducting experiments on non-emergency services data from other cities and demonstrating its superior performance. We also plan to extend the proposed GRCF models so as to jointly predict demand and response. Additionally, we plan to compare and contrast different time-series prediction models, including ARMA(autoregressive moving average) [22], various deep learning approaches such as LSTM, and different graphical models. These models can be evaluated based on prediction performance, as well as other attributes such as interpretability, ease of use, and ability to encode different



(a) HPD (b) Heat (c) Brooklyn

Figure 12: Real vs. predicted: Day 1 prediction of test data



(a) HPD (b) Heat (c) Brooklyn

Figure 13: Real vs. predicted: Day 7 prediction of test data

features. NYCER's current design primarily focuses on predicting short-term response times so as to aid city planners perform efficient resource allocation on a weekly basis. An interesting direction of future research will be to predict long-term response times by capturing the monthly and yearly seasonality of the data, taking weather conditions into account. Such long-term predictions can be particularly beneficial in making annual budget allocations and for hiring new personnel for the next year. A longer term goal will be to integrate the short-term and long-term prediction models into NYCER in a manner such that it can be readily deployed in practice.

REFERENCES

[1] [n. d.]. NYC Open Data. https://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9.

[2] Gissella Bejarano, Mayank Jain, Arti Ramesh, Anand Seetharam, and Aditya Mishra. 2018. Predictive Analytics for Smart Water Management in Developing Regions. In *Proceedings of the Conference on Smart Computing (SmartComp)*.

[3] Juan Pablo Bello, Claudio Silva, Oded Nov, R Luke DuBois, Anish Arora, Justin Salamon, Charles Mydlarz, and Harish Doraiswamy. 2018. SONYC: A System for the Monitoring, Analysis and Mitigation of Urban Noise Pollution. *Communications of the ACM, in press* (2018).

[4] Charlie Catlett, Eugenio Cesario, Domenico Talia, and Andrea Vinci. 2018. A Data-driven Approach for Spatio-Temporal Crime Predictions in Smart Cities. In *Proceedings of the Conference on Smart Computing (SmartComp)*.

[5] Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak, and Patrick Jaillet. 2017. Dynamic Repositioning to Reduce Lost Demand in Bike Sharing Systems. *Journal of Artificial Intelligence Research* 58 (2017), 387–430.

[6] Sam Goodgame, David Harding, and Romulo Manzano. [n. d.]. NYC Open Data Crunched & Visualized. http://people.ischool.berkeley.edu/šamuel.goodgame/311/#daily.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Charis Elizabeth Idicheria, Alexander Schellong, Jobst Fiedler, Matthias Kammer, Marie-Therese Huppertz, and Horst Westerfeld. 2012. A Review of 311 in New York City. (2012).

[9] Hyeon-Woo Kang and Hang-Bong Kang. 2017. Prediction of Crime Occurrence from Multi-modal Data using Deep Learning. *PloS one* 12, 4 (2017), e0176244.

[10] Jack Kelly and William Knottenbelt. 2015. Neural nilm: Deep Neural Networks Applied to Energy disaggregation. In *Proceedings of the Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys)*.

[11] Constantine Kontokosta, Boyeong Hong, and Kristi Korsberg. 2017. Equity in 311 Reporting: Understanding Socio-Spatial Differentials in the Propensity to Complain. In *Bloomberg Data for Good Exchange Conference (D4GX)*.

[12] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[13] Bruno Lefevre and Valerie Issarny. 2018. Matching Technological & Societal Innovations: The Social Design of a Mobile Collaborative App for Urban Noise Monitoring. In *Proceedings of the Conference on Smart Computing (SmartComp)*.

[14] Gaurav Mittal, Kaushal B Yagnik, Mohit Garg, and Narayanan C Krishnan. 2016. Spotgarbage: Smartphone App to Detect Garbage using Deep Learning. In *Proceedings of the Conference on Pervasive and Ubiquitous Computing (UbiComp)*.

[15] Andrew Y. Ng and Michael I. Jordan. 2001. On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*.

[16] Konstantinos Papangelis, Melvin Metzger, Yiyeng Sheng, Hai-Ning Liang, Alan Chamberlain, and Ting Cao. 2017. Conquering the City: Understanding Perceptions of Mobility and Human Territoriality in Location-based Mobile Games. *Proceedings of the Conference on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 3 (2017), 90.

[17] Nilavra Pathak, Amadou Ba, Joern Ploennigs, Nirmalya Roy, and Aryya Gangopadhyay. 2018. Forecasting Gas Usage in Large Buildings using Generalized Additive Models and Deep Learning. In *Proceedings of the Conference on Smart Computing (SmartComp)*.

[18] Havard Rue and Leonhard Held. 2005. *Gaussian Markov Random Fields: Theory And Applications (Monographs on Statistics and Applied Probability)*. Chapman & Hall/CRC.

[19] Neha Singh, Nirmalya Roy, and Aryya Gangopadhyay. 2018. Analyzing the Sentiment of Crowd for Improving the Emergency Response Services. In *Proceedings of the Conference on Smart Computing (SmartComp)*.

[20] Lingjing Wang, Cheng Qian, Philipp Kats, Constantine Kontokosta, and Stanislav Sobolevsky. 2017. Structure of 311 Service Requests as a Signature of Urban Location. *PloS one* 12, 10 (2017), e0186314.

[21] Xianjing Wang, Jonathan Liono, Will Mcintosh, and Flora D. Salim. 2017. Predicting the City Foot Traffic with Pedestrian Sensor Data. In *Proceedings of the Conference on Mobile and Ubiquitous Systems (MobiQuitous)*.

[22] Peter Whitle. 1951. *Hypothesis testing in time series analysis*. Vol. 4. Almqvist & Wiksells.

[23] Matt Wytock and Zico Kolter. 2013. Sparse Gaussian Conditional Random Fields: Algorithms, Theory, and Application to Energy Forecasting. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[24] Takahiro Yabe, Kota Tsubouchi, and Yoshihide Sekimoto. 2017. CityFlowFragility: Measuring the Fragility of People Flow in Cities to Disasters using GPS Data Collected from Smartphones. *Proceedings of the Conference on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 3 (2017), 117.

[25] Yilong Frank Zha and Manuela Veloso. 2014. Profiling and Prediction of Non-Emergency Calls in NYC. In *Workshops at AAAI Conference on Artificial Intelligence*.

[26] Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. 2014. Diagnosing New York City's Noises with Ubiquitous Data. In *Proceedings of the Conference on Pervasive and Ubiquitous Computing (UbiComp)*.