SWaP: Probabilistic Graphical and Deep Learning Models for Water Consumption Prediction

Gissella Bejarano*, Adita Kulkarni*, Raushan Raushan*, Anand Seetharam, Arti Ramesh
Computer Science Department, SUNY Binghamton
(gbejara1,akulka17,rrausha1,aseethar,artir)@binghamton.edu

ABSTRACT

Accurately predicting water consumption in residential and commercial buildings is essential for identifying possible leaks, minimizing water wastage, and for paving the way for a sustainable future. In this paper, we present **SWaP**, a **S**mart **Wa**ter **P**rediction system that predicts future hourly water consumption based on historical data. To perform this prediction task, in **SWaP**, we design discriminative probabilistic graphical and deep learning models, in particular, sparse Gaussian Conditional Random Fields (GCRFs) and Long Short Term Memory (LSTM) based deep Recurrent Neural Network (RNN) models, to successfully encode dependencies in the water consumption data. We evaluate our system on water consumption data collected from multiple buildings in a university campus and demonstrate that both the GCRF and LSTM based deep models are able to accurately predict future hourly water consumption in advance using just the last 24 hours of data at test time. **SWaP** achieves superior prediction performance for all buildings in comparison to the linear regression and ARIMA baselines in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), with the GCRF and LSTM models providing 50% and 44% improvements on average, respectively. We also demonstrate that augmenting our models with temporal features such as time of the day and day of the week can improve the overall average prediction performance. Additionally, based on our evaluation, we observe that the GCRF model outperforms the LSTM based deep learning model, while simultaneously being faster to train and execute at test time. The computationally efficient and interpretable nature of GCRF models in **SWaP** make them an ideal choice for practical deployment.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Machine learning approaches**.

*The first three authors have made equal contribution

BuildSys'19, November 13-14 2019, New York, New York USA

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

KEYWORDS

RNN, Sparse GCRF, Time series modeling

ACM Reference Format:
Gissella Bejarano*, Adita Kulkarni*, Raushan Raushan*, Anand Seetharam, Arti Ramesh. 2019. SWaP: Probabilistic Graphical and Deep Learning Models for Water Consumption Prediction. In *Proceedings of ACM BuildSys (BuildSys'19).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With climate change exacerbating extreme weather conditions including droughts and famines [1], understanding and predicting human water consumption is critical for ensuring a sustainable future. For example, the state of California, USA experienced one of its longest droughts from December 2011 to March 2019 [2]. Similarly, in recent times, the city of Capetown, South Africa was faced with a severe water crisis, where it was about to run out of drinking water for its citizens [4]. Therefore, predicting future water consumption in residential and commercial buildings has become an extremely important problem, particularly to efficiently monitor water consumption, identify possible leaks, minimize wastage, and match demand and supply. However, despite this need to design intelligent solutions to facilitate smart water usage, there is limited prior research from the computing community in this research area [7, 26].

Therefore, in this paper, we design **SWaP**, a **S**mart **Wa**ter **P**rediction system, which predicts future hourly water consumption based on historical data. The water consumption prediction problem can be viewed as a classic time series prediction problem, thus making it amenable to statistical methods such as ARIMA as well as recently developed machine learning methods. To enable **SWaP** make effective predictions, we explore two classes of discriminative machine learning models—probabilistic graphical models and deep learning models that have been shown to be effective for multiple time-series prediction problems [8, 16]. We design a structured regression graphical model, Gaussian conditional random fields (GCRFs), to successfully encode dependencies between historical and future water consumption [27]. Specifically, we leverage and adapt a recently developed sparse and computationally efficient variant of GCRFs [34]. We also design a Long Short-Term Memory (LSTM) based recurrent neural network (RNN) model that captures the underlying patterns in water consumption data.

The proposed GCRF model is parsimonious in nature and captures the underlying dependencies between the input (i.e., the past water consumption data) and output variables (i.e., the future water consumption predictions) as well as those between the output variables. As we construct a sparse GCRF model, the model only learns the necessary dependencies among the input and output variables

that are helpful in the prediction. In comparison, the proposed deep learning model consists of an encoder and a decoder, each of which separately is an RNN. The encoder takes past water consumption data and computes a state vector that encodes the underlying dependancies in the data. The decoder then utilizes this state vector to generate water consumption predictions.

To evaluate the performance of **SWaP**, we collect hourly water consumption data for 14 buildings from a university campus for the Fall 2018 semester (approximately 4.5 months). We classify these buildings into 4 categories—academic building, dining hall, gym and residence hall. The buildings in the dataset comprise of 6 academic buildings, 1 dining hall, 1 gym and 6 residence halls. We compare the performance of **SWaP** with linear regression and ARIMA baselines with respect to the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) and demonstrate that **SWaP** significantly outperforms the baselines. The GCRF and LSTM based deep learning models in **SWaP** provide an average improvement of 50% and 44%, respectively. Additionally, we demonstrate that augmenting our models with temporal features such as time of the day and day of the week can improve the overall average prediction performance.

We note that both the GCRF and deep models only require the past 24 hours of water consumption data to predict future water consumption at test time, thus making **SWaP** an attractive system that can be readily deployed in practice. Additionally, our experiments also show that the GCRF model provides overall better performance than the LSTM-based deep model. Therefore, based on our experiments, we recommend using a GCRF-based **SWaP** for the hourly water consumption prediction problem. The superior performance of GCRF models along with its low computational requirement during the training and execution phases makes **SWaP** a highly desirable and practically feasible prediction framework. Moreover, the sparse GCRF model only captures the necessary dependencies between the input and output variables, thus making the GCRF-based **SWaP** inherently interpretable.

2 RELATED WORK

In this section, we first outline research related to addressing water management problems, and then review literature related to forecasting applications in the ubiquitous computing domain.

To mitigate the negative impacts of climate change, a number of recent research initiatives have focused their attention on water management related problems [7, 23, 25, 26, 32]. Short-term forecasting of water consumption based on water meter readings is conducted in [13], while neural network based models for daily water demand forecasting on a touristic island is proposed in [25]. Assem et al. use DeepCNN to predict urban water flow and water level based on input features such as maximum temperature, minimum temperature and run-off [7]. Bejarano et al. design a random forest and SVM based framework to investigate the availability of water pumps in developing and under-developed regions [9]. Similarly, logistic regression and Bayesian analysis have been applied to understand the factors associated with the non-functionality of hand pumps [18, 19]. Prior work has also investigated the interaction and use of water with other resources such as energy and food, popularly known as the water-energy-food nexus [5, 17, 22, 31]. In comparison to existing research, we propose GCRF and LSTM based deep learning models for water consumption prediction and validate the efficacy of the models using real-world data collected from multiple buildings in a university campus.

Recently, a variety of different models including statistical models such as ARIMA [12, 33], evolutionary algorithms [28] and data-driven approaches [6, 8, 36] have been applied to variety of forecasting and smart computing tasks. Arjunan et al. design a framework called OpenBAN for electricity demand forecasting leveraging algorithms such as decision tree, neural networks, SVM, naive bayes and k-NN [6]. Deep learning models for crime prediction from multimodal data and spotting garbage from images has been proposed in [24] and [30], respectively. Mobility and traffic flow modeling at the city level has been explored in [10, 11, 35]. Similarly, model-based and machine learning techniques have also been proposed for solar power and irradiance forecasting [14, 29].

3 PROBLEM STATEMENT AND DATA

In this section, we discuss the water consumption prediction problem and provide an overview of the data collected to validate the performance of our model.

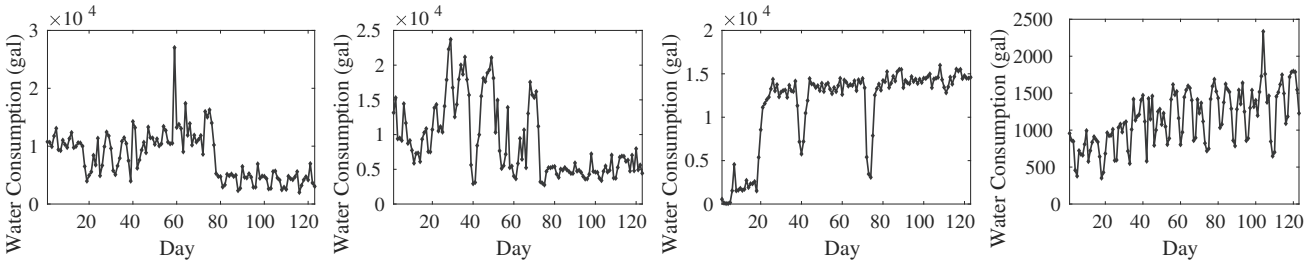
3.1 Problem Statement

In this paper, our goal is to design a system to predict hourly water consumption based on real-world data collected from multiple buildings in a university campus. Water consumption forecasting can be modeled as a classic univariate time series forecasting problem, where at any time T, the goal is to predict water consumption k steps into the future (i.e., $\hat{y}_{T+1}, \hat{y}_{T+2}$ ..... $\hat{y}_{T+k}$) based on data available for the past n time steps (i.e., x_T, x_{T-1}..... x_{T-n}). Note that $\hat{y}_{T+i}$ denotes the predicted value of the actual water consumption y_{T+i} at time $T+i$. As the problem studied here can be cast as a time series forecasting problem, both statistical techniques such as ARIMA and recently developed data-driven and machine learning approaches can be leveraged and adapted to address this problem. In this paper, we develop sequence-to-sequence probabilistic graphical and deep learning models for the water consumption prediction problem and demonstrate empirically that they perform better than ARIMA models. We discuss our rationale for choosing the above-mentioned models and the details of our system in Section 4.

3.2 Data

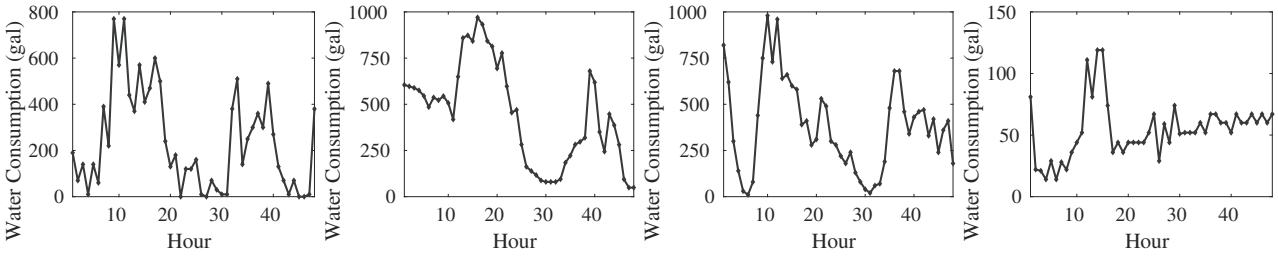
We collect hourly water consumption data for 14 buildings in a university campus. These buildings fall into 4 categories— academic building, dining hall, gym, and residence hall. The buildings in the dataset comprise of 6 academic buildings, 1 dining hall, 1 gym and 6 residence halls. We collect data for approximately 4.5 months when the university is in session, beginning from August 1, 2018 to December 8, 2018 (i.e., Fall 2018 semester). Therefore, we have approximately 3000 data points for each building. Table 1 shows the median hourly and daily water consumption for all buildings.

We discuss the general trends in water usage for buildings in each category. Figure 1 shows the daily water usage for one representative building in each category for the entire time period. We observe that during the last two months, the total water consumption decreases for the dining hall and gym (Figures 1a and 1b). While



(a) Gym GE (b) Dining Hall C4 (c) Residence Hall RA (d) Academic Building S2

Figure 1: Trends in datasets (daily consumption)



(a) Gym GE (b) Dining Hall C4 (c) Residence Hall RA (d) Academic Building S2

Figure 2: Trends in datasets (hourly consumption)

Table 1: Median Water Consumption

Building	Category	Median (Hourly)	Median (Daily)
EB	Academic Building	33	1765
FA	Academic Building	49	2478
LH	Academic Building	72	4029
S2	Academic Building	43	1148
S3	Academic Building	357	12413
SN	Academic Building	243	6455
C4	Dining Hall	304	7029
GE	Gym	280	8420
BN	Residence Hall	380	10180
BR	Residence Hall	220	5920
DE	Residence Hall	500	12910
DG	Residence Hall	410	11230
JS	Residence Hall	520	13510
RA	Residence Hall	490	13490

the exact reason is unknown, based on the timing, we hypothesize that this could be related to air-conditioning, cooling/heating. We also observe that residence halls have lower water consumption for the first 20 days (Figures 1c). This is because residence halls open from 20th August after the student orientations. In comparison, as academic buildings are in use throughout the year, we observe that the water consumption remains in the same range throughout the year (Figures 1d).

Figure 2 shows the hourly water usage for 48 hours (September 6 and 7) where hour 1 and hour 25 correspond to the time between 12 am and 1 am for two consecutive days. We observe that gym and dining hall have highest water usage from 9 am to 9 pm (which approximately corresponds to the time duration for which these facilities are open). Water consumption for residence halls drops at night for around 5 hours when most students are asleep. In comparison, academic buildings have water consumption in the same range throughout the day. We hypothesize long/late working hours of graduate students and cooling needs for equipment to be the main reason for this behavior. We note that most utilities including water and electricity are shut down during Thanksgiving week for all campus buildings. As water consumption values mostly correspond to zeroes during this week, we remove the Thanksgiving week values to prevent possible misrepresentation in the model due to this data. Additionally, the dataset has around 0.3% missing values. We use linear regression to fill in these missing values.

4 SWaP: SMART WATER PREDICTION

In this section, we provide an overview of **SWaP**, a **S**mart **Wa**ter **P**rediction system that takes as input historical water consumption data and outputs future water consumption predictions. Figure 3 shows the different components of our system. **SWaP** comprises of a data pre-processing component, which pre-processes the water consumption data and a prediction component consisting of the proposed models that takes the pre-processed data to generate the desired predictions. We design two models, a discriminative probabilistic graphical model and a deep learning model for the

prediction component in **SWaP**. Specifically, we design i) sparse Gaussian Conditional Random Fields (GCRFs) and ii) Long Short Term Memory (LSTM) based deep Recurrent Neural Network (RNN) models to successfully encode dependencies in the water consumption data. At time T, both models accept an input sequence $X = [x_T, x_{T-1}, ..., x_{T-n}]$, which corresponds to amount of water consumed in the last n time steps and generate predictions $Y = [\hat{y}_{T+1}, \hat{y}_{T+2}, ..., \hat{y}_{T+k}]$ for the next k time steps. We note that the input and output sequences can be of different lengths.

4.1 Why Sequence-to-Sequence Models?

Traditional model-based and statistical approaches (e.g., ARIMA models, filtering techniques) provide valuable insights into data, and are highly desirable when limited computational resources and data are available to make decisions. The increase in computational power, the availability of large amounts of data, and growth in the field of machine learning presents the opportunity to design data-driven techniques capable of providing superior prediction performance in real-world settings. This provides us the opportunity to explore sequence-to-sequence models that are well suited for time-series data problems requiring mapping input sequences to output sequences. Sequence-to-sequence models possess the ability to predict an entire sequence of data points based on past data, thus being able to predict further into the future. To this end, we identify sequence-to-sequence probabilistic graphical (i.e., sparse GCRFs) and deep learning models that have been extensively used for a number of forecasting and prediction tasks [7, 8]. Both GCRF and deep models elegantly learn and capture non-linear dependencies as the encoded signal passes through the network, thus having a positive impact on prediction.

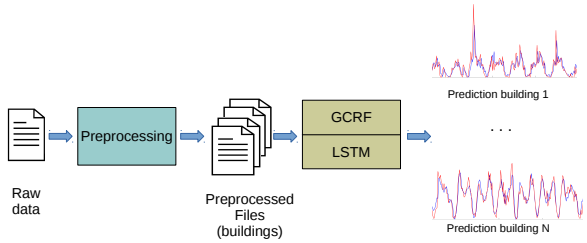


Figure 3: System Architecture

4.2 Sparse GCRF Model

In any machine learning model, a careful tradeoff between model complexity and prediction performance needs to be made to avoid overfitting and achieve good prediction performance. Hence, it is important to capture the dependencies that are important to the prediction. Conditional random fields (CRFs) are a discriminative model that only model dependencies between input features and output variables and among output variables and do not require the specification of dependencies among input features. This helps in avoiding any incorrect assumptions in dependencies among input features and focus on the dependencies that directly affect the target variables, and hence, prediction performance. In our problem, we leverage a recent version of CRF extended to structured regression, sparse GCRFs [34], for predicting future consumption.

The GCRF distribution is given by

$$P(Y|X;\Lambda,\Theta) = (1/Z(X)) * exp(-Y'\Lambda Y - 2X'\Theta Y) \quad (1)$$

where, $X = [x_1, x_2, ..., x_n]$ represents historical hourly consumption, n is the number of hours in the past, $Y = [\hat{y}_{n+1}, \hat{y}_{n+2}, ..., \hat{y}_{n+k}]$ represents predicted hourly consumption, and k indicates the number of hours 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 and Λ is the mxm inverse covariance matrix, containing the edges amongst the y's. The CRF is a Gaussian distribution with mean $-\Lambda^{-1}\Theta'X$ and variance Λ^{-1}, $\mathcal{N}(-\Lambda^{-1}\Theta'X, \Lambda^{-1})$. $Z(X)$ in Equation 1 is the partition function, which ensures that the posterior is integrated to 1.

At training time, we estimate the parameters Θ and Λ by maximizing the probability of the data given the parameters using maximum likelihood,

$$\max_{(\Lambda,\Theta)} P(Y|X;\Lambda,\Theta)$$

This is equivalent to minimizing the log-likelihood,

$$\min_{(\Lambda,\Theta)} -log(P(Y|X;\Lambda,\Theta))$$

Regularization is a way to avoid overfitting by penalizing high-valued regression coefficients and helps in making models generalize better at test time. 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 k future time steps for Y is $nk + \frac{k(k+1)}{2}$, where nk edges are given by Θ, and $\frac{k(k+1)}{2}$ by Λ. Even for $k = 12$ (as is the case in our setting), it is possible that the model can overfit due to the large number of parameters.

To retain only meaningful dependencies, this sparse variant of GCRFs incorporates L_1 regularization. L_1 regularization reduces the parameter values of dependencies that do not contribute to the prediction to zero, thus creating sparsity in the graphical model structure. As part of L_1 regularization, a penalty term equal to the absolute value of the magnitude of the coefficients is added to the GCRF objective to penalize high-valued regression coefficients and avoid overfitting due to large number of parameters. L_1 is more preferred than L_2 here as it drives less contributing parameter values to zero, thus completely removing their effect on the prediction. Thus, L_1 learns a model that is appropriately complex for the prediction problem.

We use the optimization method developed by Wytock et al. [34] to solve the GCRF with the L_1 regularization term. They develop 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 [34]. Figure 4 gives the structure of the GCRF model. We can see that there are edges showing the dependencies between the inputs X and outputs Y. Also

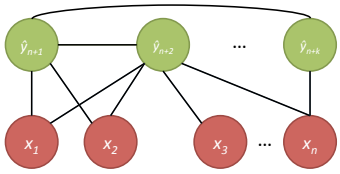


Figure 4: GCRF water consumption prediction model showing connections between historical consumption, x_1,...,x_n and y_{n+1}, ...,y_{n+k}, and among y_{n+1}, ...,y_{n+k}. Note that our model is sparse, learning only edges between variables that matter. In the graphical model, we illustrate this by leaving out some edges.

note that some edges in the graphical model have been left out to illustrate sparsity in the learned model.

4.2.1 Implementation Details. The GCRF training and test setup is given in Figure 5. We implement our models using *SGCRFPy*, a Python toolkit for sparse GCRFs[1]. We split the datasets into two parts—the first part consisting of 75% of the data is used for training and the remaining 25% is used for testing. At training time, our GCRF models use past n hours as input and next k hours as output. As water consumption patterns typically are likely to follow a 24-hour cycle, we use $n = 24$ and $k = 12$ in our experiments. The parameter Λ is initialized to the identity matrix and Θ is initialized to all zeros. We use regularization constant $\lambda = 0.1$ and train the model for $10,000$ iterations to converge on a set of dependencies learned from the training data. For each building, we train a separate GCRF model.

The parameter values Λ and Θ learned at training time are plugged into each test sequence of length n to generate a prediction for the next k hours. To compute the prediction performance scores, the predicted values $\hat{Y}$ are compared with the ground truth water consumption values Y. For a particular configuration of parameter values, the training time for GCRF is less than 5 minutes on a standalone lab machine. The RAM requirement for training is also low. The testing phase only takes a couple of minutes.

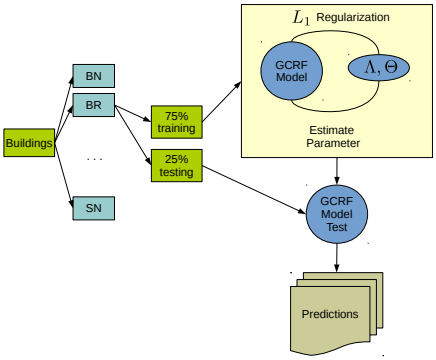


Figure 5: GCRF Training and Test Setup

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

4.3 RNN Encoder-Decoder Model

The temporal dependence between the data instances in a sequence prediction problem make recurrent neural networks (RNNs) an appropriate fit for the problem. An RNN consists of a hidden state h and an output Y that operates on input X. At each time step t, the hidden state of the RNN is given by,

$$h_t = f(h_{t-1}, x_t) \tag{2}$$

where, f is any non-linear activation function and $1 \leq t \leq n$. In our problem, f follows a neural network architecture comprising of a network of nodes organized into sequential hidden layers with each node in a given layer being full connected to every other node in the next successive layer. Each hidden state serves as memory and its output is calculated using the output of the previous hidden state and the input x_t as shown in Equation 2. Since RNNs are known to suffer from the vanishing or exploding gradient problem [21] when sigmoid functions are used, our architecture uses LSTM cells that use memory cells to store relevant information needed to learn long range temporal dependencies in the data. We refer the reader to Goodfellow et al. for more details on RNN [20].

We develop an LSTM-based RNN encoder-decoder sequence-to-sequence model as shown in Figure 6 [15]. The architecture of both the encoder and decoder is an RNN. The basic cell in both the encoder and the decoder is an LSTM. The encoder accepts an input sequence and generates a hidden encoded vector c encapsulating the information for the input sequence. This encoded vector is given as an input to the decoder, which then generates the predictions. The input X is transformed into the output $\hat{Y}$ using the hidden layers and the weight matrices. The weight matrices essentially capture the information needed to generate the output predictions based on the input data. The LSTM cell used in our model consists of a number of interconnected gated units. The three gates in an LSTM cell are namely, the input gate, the output gate, and the forget gate that lets it handle long-term dependencies. To prevent prediction of negative water consumption values, a ReLU activation function is used after each decoder output.

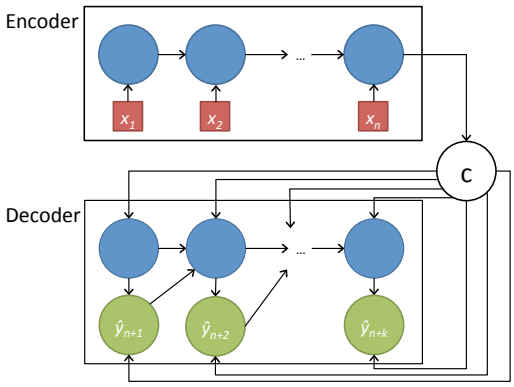


Figure 6: Encoder-decoder based RNN

4.3.1 Implementation Details. We use TensorFlow[2] for implementing the deep learning models. As mentioned in Section 4.2.1, we

[2]https://www.tensorflow.org/

similarly split the datasets into two parts—the first part consisting of 75% of the data is used for training and the remaining 25% is used for testing. Similar to the GCRF models, we use water consumption of past 24 hours (i.e., n = 24) and predict 12 hours into the future (i.e., k = 12). These settings ensure that the results from both these models are directly comparable. As deep models are computationally expensive, we train our models on a shared high performance computing cluster available at our university. Using this cluster, we are able to execute 10 to 15 experiments in parallel. Each experiment is allocated 4 cores and 10 GB of RAM. For the datasets considered in this work, for a particular configuration of parameters, training the deep models (i.e., a single experiment) can take in the order of 1 - 12 hours, which is typical of deep learning models.

We experiment with different number of stacked layers, different numbers of hidden units in each layer as well as the lengths of the input and output sequences. We observe that depending on the dataset, different parameter configurations provide the best performance. However, we empirically observe that overall 1 stacked layer with 200 hidden units generalizes better across all the buildings. We use a learning rate of 0.01 and train the model for 1000 epochs. At training time, the encoder and decoder are trained jointly using the backpropagation algorithm. We use unguided training as the training scheme, where the decoder uses previous predicted output value as an input to the next step of the decoder. Unguided training enables the model to better explore the state space, which usually results in superior prediction performance at test time. Additionally, to minimize overfitting, we incorporate L2 regularization in the models.

In comparison to training, the testing phase of a model takes only a couple of minutes for each experiment. The learned weight values for the different connections in the neural network are used to generate a prediction for the test data instances.

5 PERFORMANCE EVALUATION

We compare the performance of GCRF and deep learning models with two baselines—linear regression and Auto-Regressive Integrated Moving Average (ARIMA) models. The code for our models, the pre-processed data, and the experiments is available in [3].

Linear Regression: It is a simple statistical model that fits the best straight line based on the input data.

ARIMA(p, d, q): It is a statistical model that has three components — AR (autoregressive term), I (differencing term) and MA (moving average term), which are specified by p, d and q respectively. p represents the past values used for predicting the future values, d represents the degree of differencing (i.e., the number of times the differencing operation is performed to make a series stationary), and q represents the number of error terms used to predict the future values. At any time T, the equation of ARIMA used for prediction is given by,

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d x_T = (1 - \sum_{i=1}^{q} \theta_i L^i) e_T \tag{3}$$

where x_T corresponds to the water consumption values, ϕ_i and θ_i are the auto-regressive and moving average parameters, e_T are the error terms and L is the lag term. The error terms e_T are assumed to be independently and identically distributed according to normal distribution. In our experiments, we use the Auto-ARIMA toolkit[3] in python that searches through a combination of the parameters p, d, and q, and picks the optimal combination for the data in consideration. As both linear regression and ARIMA are statistical baselines, they do not require any explicit training. They use water consumption for the past 24 hours to predict 12 hours into the future.

We use root mean squared error (RMSE) and mean absolute error (MAE) as the main evaluation metrics, which are given by Equations 4 and 5 respectively.

Table 2: RMSE

(a) Hour 1

Building	GCRF	LSTM	ARIMA	LR
BN	**130.69**	140.08	175.71	314.12
BR	68.62	**65.44**	92.6	166.06
C4	101.64	**99.56**	123.83	233.66
DE	**151.17**	169.45	192.74	330.34
DG	**141.86**	168.11	200.16	330.06
EB	59.16	**58.64**	63	89.36
FA	**48.48**	57.91	63.27	85.49
GE	**119.88**	128.27	140.88	184.78
JS	**127.04**	173.75	183.13	361.19
LH	**80.55**	83.2	109.29	180.39
RA	**161.22**	178.15	230.01	400.57
S2	**18.78**	22.19	24.12	32.88
S3	**196.37**	259.56	269.45	400
SN	**112.79**	116.23	119.68	125.89

(b) Average

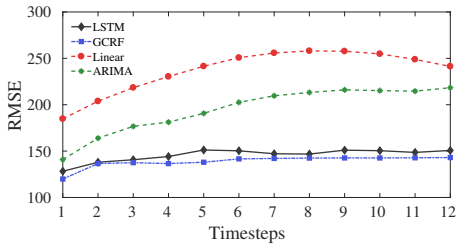
Building	GCRF	LSTM	ARIMA	LR
BN	**151.2**	154.95	345.31	406.11
BR	88.28	**84.89**	171.29	206.36
C4	**148.26**	164.81	256.28	316.19
DE	**184.95**	201.96	364.89	418.18
DG	**168.83**	190.53	329.99	398.63
EB	**70.5**	75.46	103.09	114.35
FA	**58.17**	81.93	107.35	111.26
GE	**138.81**	145.61	195.25	237.16
JS	**162.15**	190.96	380.72	461.5
LH	**116.77**	122.53	249.26	242.29
RA	**188.31**	203.88	412.93	488.15
S2	**23.26**	26.01	40.23	44.11
S3	**238.64**	313.26	516.58	536.86
SN	**124.43**	134.13	144.42	155.8

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^{h} \left(\hat{y}_{ij} - y_{ij}\right)^2}{h}} \tag{4}$$

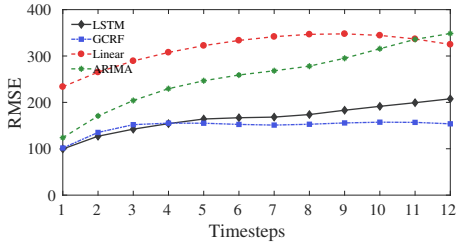
$$MAE_j = \frac{\sum_{i=1}^{h} |\hat{y}_{ij} - y_{ij}|}{h} \tag{5}$$

[3] https://pypi.org/project/pyramid-arima/

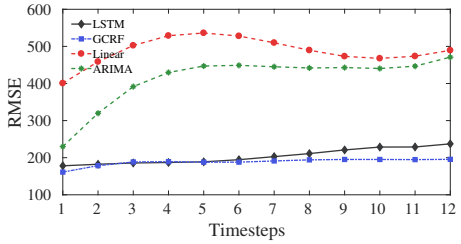
where y_{ij} is the i^{th} test sample for j^{th} hour, $\hat{y}_{ij}$ is the predicted value of y_{ij}, and h is the total number of test samples.



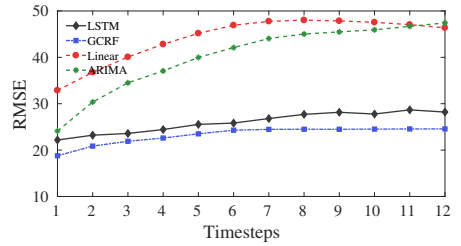
(a) Gym GE



(b) Dining Hall C4



(c) Residence Hall RA

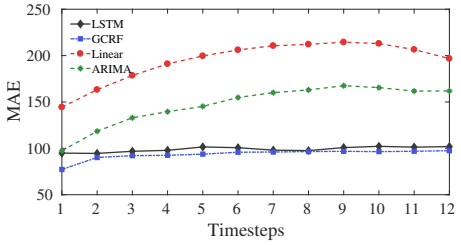


(d) Academic Building S2

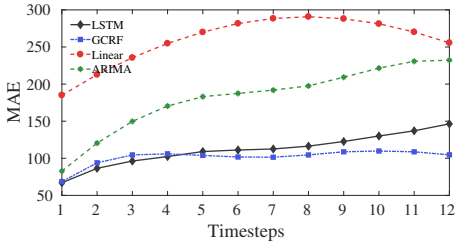
Figure 7: RMSE

5.1 RMSE

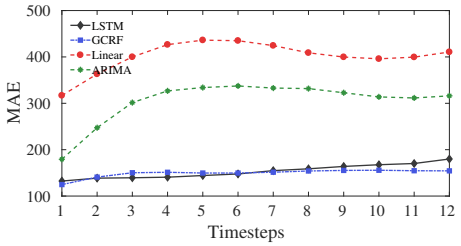
In this subsection, we discuss RMSE results for all models. Figure 7 shows the performance of the models for one building in each category. From Figure 7, we observe that GCRF and LSTM outperform the baselines significantly. We also observe that RMSE values for linear regression and ARIMA increase considerably with each predicted hour into the future. In comparison, the RMSE values increase gradually for the GCRF and LSTM models, which demonstrates that our models are able to predict considerably better into the future. We attribute this to the sequence-to-sequence modeling aspect of these models.



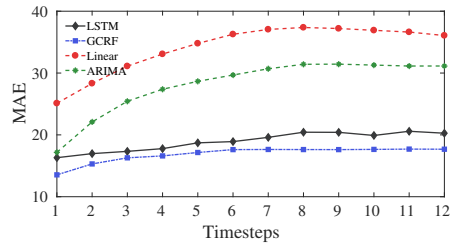
(a) Gym GE



(b) Dining Hall C4



(c) Residence Hall RA



(d) Academic Building S2

Figure 8: MAE

Table 2 shows RMSE results for hour 1 and the average over the 12 predicted hours for all buildings. We observe from the table that for all the buildings, GCRF and LSTM outperform the baselines. The overall performance improvement of GCRF over ARIMA and linear regression is in the range of 14% to 65%, while the gains of LSTM over ARIMA and linear regression is in the range of 7% to 62%. We also see that for most buildings GCRF performs better than LSTM. We believe that the sparse nature of the L_1-regularized GCRF model helps in learning the dependencies that positively affect the prediction performance, while excluding those that do

Table 3: MAE

(a) Hour 1

Building	GCRF	LSTM	ARIMA	LR
BN	**98.51**	105.04	138.36	254.13
BR	52.5	**49.44**	73.14	132.17
C4	68.52	**67.05**	82.81	185.04
DE	**105.55**	121.29	147.45	266.17
DG	**108.11**	122.9	158.01	262.01
EB	**26.48**	30.02	33.07	58.62
FA	**28.08**	36.51	38.51	64.97
GE	**77.25**	94.86	97.35	144.4
JS	**98.73**	134.3	146.78	291.42
LH	**51.54**	52.84	65.61	129.14
RA	**124.78**	132.38	179.4	316.81
S2	**13.54**	16.32	17.18	25.11
S3	**134.77**	165.3	179.33	295.57
SN	**66.02**	70.38	77.5	87.94

(b) Average

Building	GCRF	LSTM	ARIMA	LR
BN	**112.94**	115.82	257.6	341.42
BR	65.84	**63.82**	132.02	169.25
C4	**101.38**	111.46	181.42	259.52
DE	**133.9**	147.14	266.29	345.39
DG	**129.51**	145.06	259.23	327.47
EB	**37.09**	41.76	63.51	81.45
FA	**36.21**	51.58	71.64	89.36
GE	**93.48**	99.01	147.34	194.71
JS	**125.53**	149.52	291.68	383.13
LH	**74.62**	77.7	150.07	178.77
RA	**149.31**	153.18	304.59	401.4
S2	**16.87**	18.94	28.13	34.16
S3	**161.37**	203.31	325.96	414.75
SN	**80.42**	89.05	100.92	115.11

not matter. This helps in yielding a model that is better suited to the data.

5.2 MAE

In this subsection, we discuss the MAE results for all models. Figure 8 shows the performance of the models for one building in each category. In comparison, Table 3 shows the 1 hour and the average (taken over predictions for the next 12 hours) MAE results. We observe that the GCRF and LSTM models outperform the baselines for all buildings with respect to MAE. The performance improvement of GCRF over ARIMA and linear regression is in the range of 20% to 67%, while improvement of LSTM over ARIMA and linear regression is in the range of 12% to 66% with respect to the average MAE. Once again, we see that for most buildings GCRF achieves a better performance than LSTM. The performance improvement of GCRF over LSTM is approximately 10%.

5.3 Qualitative Results

In this subsection, we compare the qualitative prediction performance of the GCRF and LSTM models with the baselines to help the reader appreciate the superior performance of our models. Figures 9a and 9b show the 1 hour and 12 hour predictions for GCRF and linear regression, while Figures 10a and 10b show the 1 hour and 12 hour predictions for LSTM and ARIMA for residence hall RA. For the 1 hour prediction, we observe that as linear regression tends to closely follow the actual values in the previous time step, it provides poor prediction performance as the recent past may not mirror the future. In comparison, GCRF generates smoothened predictions as it is trained on entire input sequences and thus provides superior performance. Additionally, we observe from Figure 9b that the 12 hour prediction for linear regression is notably worse than its 1 hour prediction. In comparison, as GCRF takes entire sequences into account and captures the underlying variations in the data, its 12 hour prediction performance does not deteriorate significantly. Similar to GCRF, as LSTM is also a sequence-to-sequence model and elegantly capture the dependencies in the data, its prediction performance does not decrease with larger time steps (Figures 10a and 10b).

5.4 Adding Temporal Features

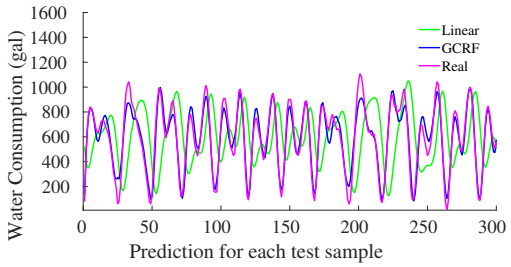
In the experimental results reported so far, we have only used the previous water consumption data to predict future water consumption. In this subsection, we investigate the performance improvement of augmenting our GCRF and LSTM models with temporal features. To this end, we add two features— i) day of the week and ii) hour of the day in our model. Day of the week takes values from 1 to 7, where 1 denotes Sunday. Hour of the day take values from 1 to 24, where 1 denotes the time period from 12 am to 1 am. Table 4 shows the average performance improvement over 12 predictions obtained by our augmented models over their respective baseline GCRF and LSTM models. We observe from the table that including the temporal features improves performance for most buildings for both GCRF and LSTM. The average performance improvement for GCRF and LSTM are 8.42% and 10.31%, respectively. The highest improvement is observed in academic buildings where the performance is enhanced by around 15% for building S3 in GCRF and 24% for building LH in LSTM.

Table 4: Percentage improvement after adding features

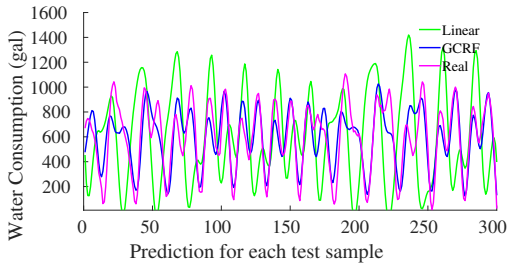
Building	GCRF	LSTM
BN	6.68%	10.49%
BR	8.36%	10.54%
C4	5.01%	-
DE	7.66%	6.67%
DG	7.6%	15.6%
JS	8.34%	7.51%
LH	-	24.34%
RA	6.43%	5.92%
S2	11.13%	6.57%
S3	14.58%	5.21%

5.5 Varying Sequence Length

In this subsection, we discuss the impact of varying sequence length and the rationale behind choosing 24 time steps as the input sequence length. Table 5 shows the average RMSE results for input

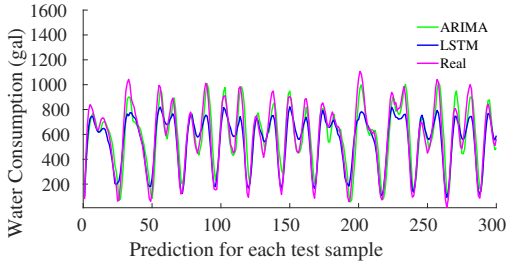


(a) Residence Hall RA: Hour 1

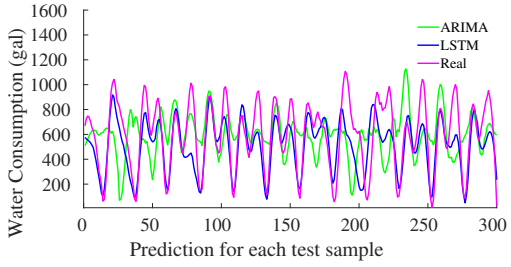


(b) Residence Hall RA: Hour 12

Figure 9: Qualitative Results: GCRF vs Linear Regression



(a) Residence Hall RA: Day 1



(b) Residence Hall RA: Day 12

Figure 10: Qualitative Results: LSTM vs ARIMA

Table 5: RMSE Varying Sequence Length

Building	GCRF			LSTM		
	12	18	24	12	18	24
BN	252.71	175.74	151.2	176.83	165.27	154.95
BR	134.87	101.33	88.28	95.81	80.69	84.89
C4	184.67	151.78	148.26	173.33	158.43	164.81
DE	272.67	208.73	184.95	220.61	217.21	201.96
DG	264.93	198.56	168.83	221.84	189.84	190.53
EB	79.56	72.18	70.5	80.91	80.37	75.46
FA	71.76	60.57	58.17	90.91	87.39	81.93
GE	163.67	142.03	138.81	226.45	148.29	145.61
JS	283.89	194	162.15	244.38	187.51	190.96
LH	135.44	118.01	116.77	126.84	138.97	122.53
RA	312.18	225.17	188.31	259.07	204.04	203.88
S2	28.77	24.11	23.26	31.65	28	26.01
S3	316.76	247.7	238.64	330.36	326.43	313.26
SN	130.51	125.12	124.43	135.75	133.03	134.13

sequence lengths 12, 18 and 24. We observe that for both models RMSE values are the worst for all buildings when the sequence length is 12. We also see that for most buildings having sequence length of 24 provides better performance than sequence length of 18. This is because a sequence length of 24 captures water consumption behavior for all hours of the day. Having sequence lengths greater than 24 does not significantly improve performance as longer sequences only reinforce previously learnt structure in the data.

5.6 Discussion on SWaP's practicality

The above experiments demonstrate that the GCRF-based **SWaP** overall outperforms the LSTM-based **SWaP**. Therefore, we recommend using the GCRF-based **SWaP** due to its superior prediction performance. Employing the GCRF-based **SWaP** also provides the system with greater interpretability as GCRF is a probabilistic graphical model and it is easy to understand and appreciate which inputs/past outputs are instrumental in arriving at the predictions. These insights can help in understanding the inherent patterns in the data and explain the predictions, when necessary.

Additionally, in comparison to deep learning models, GCRF models require significantly less time (around 5 minutes when compared to few hours for deep learning models) and limited computational resources to train. This further means that in a deployed system, as new data becomes available, it is relative easy to re-train the model. Also, we observe that both the models perform well during test time on > 30 days of consecutive data without the need for re-training. Thus, it is only required to re-train both the models at comparatively infrequent intervals, aiding in practical deployment.

Another attractive aspect of **SWaP** is its low data and computational power requirement at test time. A well-trained **SWaP** system only requires 24 prior data points at test time to make strong predictions. Moreover, both GCRF and deep models are highly computationally efficient at test time, which means that it can generate the predictions quickly, a desired attribute in a practical system. These characteristics of **SWaP**, in particular the GCRF-based one, make it a useful system for managing water consumption. These

qualities also make the system potentially extensible to other water management scenarios.

6 CONCLUSION

In this paper, we investigated the hourly water consumption prediction problem using data collected from multiple buildings in a university campus. We designed **SWaP**, a **S**mart **Wa**ter **P**rediction system to accurately predict future hourly water consumption based on historical data. To enable **SWaP** make good predictions, we designed discriminative probabilistic graphical and deep learning models, in particular sparse GCRF and LSTM based deep models that successfully capture dependencies in the water consumption data. Our experimental evaluation shows that **SWaP** achieves superior prediction performance for all buildings, when compared to linear regression and ARIMA baselines in terms of RMSE and MAE. Additionally, we observed that a GCRF-based model provides better performance than an LSTM based deep learning model. Therefore, we recommend adopting the computationally efficient and interpretable GCRF-based **SWaP**, which makes our model practically attractive.

REFERENCES

[1] 2014. Executive Summary of the National Climate Assessment. https://nca2014.globalchange.gov/highlights/report-findings/extreme-weather

[2] 2019. Drought.gov,US Drought Portal. https://www.drought.gov/drought/states/california

[3] 2019. Paper: Code and Data. https://bitbucket.org/gissemari/water-consumption-prediction

[4] 2019. Wikipedia: Cape Town Water Crisis. https://en.wikipedia.org/wiki/Cape_Town_water_crisis

[5] Babkir Ali. 2018. Forecasting model for water-energy nexus in Alberta, Canada. *Water-Energy Nexus* 1, 2 (2018), 104 – 115. https://doi.org/10.1016/j.wen.2018.08.002

[6] Pandarasamy Arjunan, Mani Srivastava, Amarjeet Singh, and Pushpendra Singh. 2015. OpenBAN: An Open Building ANalytics Middleware for Smart Buildings. In *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. 70–79.

[7] Haytham Assem, Salem Ghariba, Gabor Makrai, Paul Johnston, Laurence Gill, and Francesco Pilla. 2017. Urban water flow and water level prediction based on deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 317–329.

[8] Gissella Bejarano, David DeFazio, and Arti Ramesh. 2019. Deep Latent Generative Models For Energy Disaggregation. In *Thirty-Third AAAI Conference on Artificial Intelligence*.

[9] Gissella Bejarano, Mayank Jain, Arti Ramesh, Anand Seetharam, and Aditya Mishra. 2018. Predictive analytics for smart water management in developing regions. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 464–469.

[10] Ravi Bhandari, Akshay Nambi, Venkat Padmanabhan, and Bhaskaran Raman. 2018. DeepLane: camera-assisted GPS for driving lane detection. In *Proceedings of the 5th Conference on Systems for Built Environments*. ACM.

[11] Romil Bhardwaj, Gopi Krishna Tummala, Ganesan Ramalingam, Ramachandran Ramjee, and Prasun Sinha. 2017. Autocalib: automatic traffic camera calibration at scale. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM, 14.

[12] Grzegorz Borowik, Zbigniew M Wawrzyniak, and Paweł Cichosz. 2018. Time series analysis for crime forecasting. In *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE, 1–10.

[13] Antonio Candelieri, Davide Soldi, and Francesco Archetti. 2015. Short-term forecasting of hourly water consumption by using automatic metering readers data. *Procedia Engineering* 119 (2015), 844 – 853. https://doi.org/10.1016/j.proeng.2015.08.948 Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management.

[14] Dong Chen, Joseph Breda, and David Irwin. 2018. Staring at the sun: a physical black-box solar performance model. In *Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 53–62.

[15] Kyunghyun Cho, Bart van Merriënboer Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares Holger Schwenk, and Yoshua Bengio. [n.d.]. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. ([n. d.]).

[16] David DeFazio, Arti Ramesh, and Anand Seetharam. 2018. NYCER: A Non-Emergency Response Predictor for NYC using Sparse Gaussian Conditional Random Fields. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 187–196.

[17] Aiko Endo, Izumi Tsurita, Kimberly Burnett, and Pedcris M Orencio. 2017. A review of the current state of research on the water, energy, and food nexus. *Journal of Hydrology: Regional Studies* 11 (2017), 20–30.

[18] Michael B Fisher, Katherine F Shields, Terence U Chan, Elizabeth Christenson, Ryan D Cronk, Hannah Leker, Destina Samani, Patrick Apoya, Alexandra Lutz, and Jamie Bartram. 2015. Understanding handpump sustainability: Determinants of rural water source functionality in the Greater Afram Plains region of Ghana. *Water resources research* 51, 10 (2015), 8431–8449.

[19] Tim Foster. 2013. Predictors of sustainability for community-managed handpumps in sub-Saharan Africa: evidence from Liberia, Sierra Leone, and Uganda. *Environmental science & technology* 47, 21 (2013), 12037–12046.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*.

[21] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6, 02 (1998), 107–116.

[22] H Hoff. 2011. Understanding the Nexus; Background paper for the Bonn2011 Conference: The Water, Energy and Food Security Nexus; Stockholm Environment Institute: Stockholm, Sweden, 2011.

[23] Aida Jabbari and Deg-Hyo Bae. 2018. Application of Artificial Neural Networks for Accuracy Enhancements of Real-Time Flood Forecasting in the Imjin Basin. *Water* 10, 11 (2018), 1626.

[24] 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.

[25] Dimitris Kofinas, Elpiniki Papageorgiou, C Laspidou, Nikolaos Mellios, and Konstantinos Kokkinos. 2016. Daily multivariate forecasting of water demand in a touristic island with the use of artificial neural network and adaptive neuro-fuzzy inference system. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. IEEE, 37–42.

[26] Hyosun Kwon, Joel E Fischer, Martin Flintham, and James Colley. 2018. The Connected Shower: Studying Intimate Data in Everyday Life. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 176.

[27] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[28] Karl Mason, Jim Duggan, and Enda Howley. 2018. Forecasting energy demand, wind generation and carbon dioxide emissions in Ireland using evolutionary neural networks. *Energy* 155 (2018), 705–720.

[29] Fateh Nassim Melzi, Taieb Touati, Allou Same, and Latifa Oukhellou. 2016. Hourly solar irradiance forecasting based on machine learning models. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 441–446.

[30] 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 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 940–945.

[31] Khulood A. Rambo, David M. Warsinger, Santosh J. Shanbhogue, John H. Lienhard V, and Ahmed F. Ghoniem. 2017. Water-Energy Nexus in Saudi Arabia. *Energy Procedia* 105 (2017), 3837 – 3843. https://doi.org/10.1016/j.egypro.2017.03.782 8th International Conference on Applied Energy, ICAE2016, 8-11 October 2016, Beijing, China.

[32] Christopher Tull, Eric Schmitt, and Patrick Atwater. [n.d.]. How Much Water Does Turf Removal Save? Applying Bayesian Structural Time-Series to California Residential Water Demand. ([n. d.]).

[33] Qiang Wang, Shuyu Li, and Rongrong Li. 2018. Forecasting energy demand in China and India: Using single-linear, hybrid-linear, and non-linear time series forecast techniques. *Energy* 161 (2018), 821 – 831. https://doi.org/10.1016/j.energy.2018.07.168

[34] Matt Wytock and Zico Kolter. 2013. Sparse Gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In *International conference on machine learning*. 1265–1273.

[35] 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 ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 117.

[36] Zhiang Zhang and Khee Poh Lam. 2018. Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system. In *Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 148–157.