DeepFit: Deep Learning based Fitness Center Equipment Use Modeling and Prediction

Adita Kulkarni, Anand Seetharam, Arti Ramesh
Computer Science Department, SUNY Binghamton
(akulka17,aseethar,artir)@binghamton.edu

ABSTRACT

In today's busy modern life, modeling and accurately predicting fitness center equipment usage and availability is essential for improving human fitness and well-being as it provides people the flexibility to plan their schedule and exercise at their convenience. In addition to its crucial role in ensuring a healthy and sustainable future, adopting a data-driven approach for modeling and predicting fitness center equipment usage is necessary for planning the optimal square footage for developing a fitness center, and determining the kinds of equipment to purchase and install. In this paper, we develop *DeepFit*, a deep learning based system that predicts future fitness center equipment usage based on historical data. To this end, we design a Long Short Term Memory (LSTM) based sequence-to-sequence model that captures the dependencies in the data. The sequence-to-sequence model comprises of an encoder and a decoder, each of which separately is a deep Recurrent Neural Network (RNN). The basic cell structure in the RNN architecture is an LSTM cell.

We evaluate *DeepFit* on equipment usage data collected from a university campus fitness center over a period of 1.5 years and demonstrate that it is able to accurately predict future fitness center equipment usage. We show that *DeepFit* outperforms the linear regression and ARIMA baselines in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) providing 17% performance improvement on average. We then present a discussion on hyper-parameter tuning and selection in our model. Finally, we investigate the benefits of augmenting the deep learning model in *DeepFit* with features such as whether the school is in session and month of the year and observe that the enhanced *DeepFit* system obtains performance improvements of 35% and 32% over linear regression and ARIMA, respectively. Our experiments show that the trained *DeepFit* model requires limited computational resources at test time, thus making it an attractive system for practical deployment.

CCS CONCEPTS

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

MobiQuitous, November 12–14, 2019, Houston, TX, USA

ACM ISBN 978-1-4503-7283-1/19/11…$15.00
https://doi.org/10.1145/3360774.3360803

KEYWORDS

Deep learning, RNN, Time series modeling

ACM Reference Format:
Adita Kulkarni, Anand Seetharam, Arti Ramesh. 2019. DeepFit: Deep Learning based Fitness Center Equipment Use Modeling and Prediction. In *16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous), November 12–14, 2019, Houston, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3360774.3360803

1 INTRODUCTION

According to the US department of Health and Human Services, "*Physical activity, along with proper nutrition, is beneficial to people of all ages, backgrounds, and abilities. And it is important that everyone gets active: over the last 20 years, there's been a significant increase in obesity in the United States*" [2]. Given the importance of physical activity and exercise and the long sedentary working hours of modern life, fitness centers play a crucial role in promoting good health and wellness [7]. Having access to fitness center equipment that works with an individual's schedule is essential for the person to engage in physical activity and maintain a healthy lifestyle.

One of the main challenges in this regard is accurately predicting future fitness center equipment usage based on prior data so that a person can plan her visit to the fitness and wellness center. By aiding planning and time management, such predictions can reduce wait times at the fitness center and thus, encourage more people to engage in physical activity and take better care of their health. Additionally, these predictions are also key to resource management—they can help determine the real estate needs for setting up a new fitness center in a community and for determining the types of new equipment to purchase. Despite its importance in the context of smart cities and for maintaining a healthy and sustainable future, this problem has received little attention till date from the computing research community.

In this paper, our goal is to develop a system for accurately predicting fitness center equipment usage based on historical data. We model the fitness center equipment usage prediction as a time series forecasting problem, which means that both model-based approaches such as Auto-Regressive Integrated Moving Average (ARIMA) as well as machine learning models, in particular sequence-to-sequence models, can be used for addressing this problem. Recent research has demonstrated the usefulness and appropriateness of adopting a sequence-to-sequence modeling approach for problems that require mapping input sequences to output sequences (including time-series data) as they elegantly capture the underlying dependencies in the data [4, 6].

Therefore, we develop *DeepFit*, a Long Short Term Memory (LSTM) based sequence-to-sequence neural network model that predicts usage of fitness center equipment using data collected from a university campus recreational center. *DeepFit* comprises of two components, an encoder and a decoder. The encoder takes the input sequence (i.e., past equipment usage data) into consideration and generates an encoded vector. The decoder then uses this encoded vector to generate the future predictions (i.e., future equipment usage). Both the encoder and the decoder are designed as a standard Recurrent Neural Network (RNN) with the inner cell structure in each RNN being an LSTM cell.

To evaluate the efficacy of *DeepFit*, we collect fitness center equipment usage data at multiple different times of a day for a period of approximately 1.5 years from the recreational center at our university. We collect data at three different times of a day (i.e., morning, afternoon and evening) for the four most popular classes of gym equipment—Cardio, Free Weights, Strength Machines and Synergy360. We compare the performance of *DeepFit* with the linear regression and ARIMA baselines with respect to multiple performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), and demonstrate that *DeepFit* significantly outperforms these baselines (17% on average).

We also present qualitative results that visually demonstrate that by taking entire input sequences into account for generating the output, *DeepFit* is able to predict future equipment usage better than the linear regression and ARIMA baselines. We then present a discussion on the choice of hyper-parameters such as sequence length, number of stacked layers, number of hidden units in each layer and learning rate. Finally, we investigate the benefits of augmenting the deep learning model in *DeepFit* with features such as whether the school is in session and month of the year and observe that the enhanced *DeepFit* system obtains performance improvements of 35% and 32% over linear regression and ARIMA, respectively. Our experimental results demonstrate that the trained *DeepFit* system requires limited computational resources at test time, thus making it practically useful.

The rest of paper is organized as follows. We provide an overview of related work in Section 2 and then describe the data and the problem studied in this paper in Section 3. We discuss *DeepFit* and the LSTM based deep sequence-to-sequence prediction model in Section 4. We present our evaluation results and conclude the paper in Sections 5 and 6 respectively.

2 RELATED WORK

In this section, we first describe previous work related to fitness and healthcare, and then discuss machine learning applications in smart cities and ubiquitous computing domain.

Human Activity Recognition (HAR) using wearable devices is directly related to fitness monitoring. Deep learning applications in HAR using wearable sensors have been presented in [10, 11, 13, 22, 27, 29]. For example, the authors explore deep, convolutional and recurrent models for recognizing human activities such as walking, running and exercise [11]. Similarly, a system called AROMA is presented in [22], where the authors use CNNs for simple activity recognition and LSTMs for complex activity recognition. In comparison, the authors in [10] combine multiple LSTM learners into ensemble classifiers for activity recognition.

In recent years, deep learning applications in the healthcare domain have also been proposed [3, 8, 14, 24, 25]. In [8], authors propose multi-output deep architectures for multi-step forecasting of blood glucose trajectories. Sideris *et al.* design a deep RNN based model for measuring blood pressure continuously through pulse oximeters [24], while Liu *et al.* design an end-to-end neural model for automatic dietary assessment [17]. Similarly, Subspace Network [25], an efficient deep modeling approach for non-linear multi-task censored regression has been developed to model neurodegenerative diseases. Disease diagnosis using context-aware hierarchical reinforcement learning and cardiovascular risk prediction using semi-supervised and multi-task LSTM has also been developed in [14] and [3], respectively. Designing models and systems for detecting social anxiety and stress has also received attention in the computing domain. For example, the authors in [23] present a weakly supervised learning framework for detecting social anxiety and depression from long audio clips, while Lin *et al.* design a deep neural network model in [16] to detect users' psychological stress from social media.

Additionally, machine learning approaches have also been applied to solve a variety of problems in the urban and ubiquitous computing domain [5, 12, 15, 20, 21, 30]. Authors in [30] present a deep multi-view spatial-temporal network to model taxi demand. Hasan *et al.* propose an optimization approach to maximize trip sharing that exploits clustering, shareability graphs and mixed integer programming. In [5], authors propose ADAIN, a model for urban air quality inference, which combines feedforward and recurrent neural networks. Authors in [20] introduce SpotGarbage, an Android app which employs CNN to detect garbage. [15] adapts three neural network architectures for energy disaggregation, while [21] presents two approaches for forecasting gas consumption—Generalized Additive Models and LSTM.

In contrast to prior work, in this paper, we design *DeepFit*, a system to predict future fitness center equipment usage, a problem that falls in the category of smart and ubiquitous computing, but has not been investigated in previous research.

3 PROBLEM STATEMENT AND DATA

In this section, we provide an overview of the fitness center equipment usage prediction problem and then describe the dataset that we collect at our university to validate the performance of our solution.

3.1 Problem Statement

In this paper, our goal is to leverage recent advancements in machine learning and computing research to improve the health and wellness of our communities. To this end, we tackle the fitness center equipment usage prediction problem, where the objective is to accurately predict future equipment usage based on past data. We cast this problem as a time series prediction problem, where given an input sequence (i.e., x_1, x_2, ..., x_n), which corresponds to the machine usage for the last N time steps, the goal is to generate predictions (i.e., $\hat{y}_1$, $\hat{y}_2$, ..., $\hat{y}_k$) for the machine usage for the next K time steps.

Traditionally, time series prediction/forecasting problems have been tackled using statistical and model-based approaches (e.g., regression, ARIMA models) that primarily attempt to capture the dependencies in the data using a combination of linear models. While these model-based approaches provide valuable insights and are useful when one has limited data and computational power at her disposal, they often provide below par prediction performance due to underlying complexities in real-world data. Therefore, in this paper, we develop LSTM based sequence-to-sequence deep learning models for the fitness center equipment usage prediction problem and demonstrate empirically that they perform better than linear regression and ARIMA models.

3.2 Data

We collect machine usage data from a university campus fitness center for around 515 days (28 November 2017 to 27 April 2019). We collect data for four machines—Cardio, Free Weights, Strength Machines and Synergy360 for three time periods during the day (i.e., morning, afternoon and evening). Data is collected between 6 am to 10.59 am for the morning session, 11.00 am to 3:59 pm for the afternoon session and 4.00 pm to 8.59 pm for the evening session. We record multiple values during each session and take an average over them to get one reading in each session. The gym is closed during holidays such as Christmas, Thanksgiving and Memorial day. Hence, we do not record usage data during the holidays. Apart from this, the data contains 9% missing values. We fill all these missing values by taking an average of the five previous readings for that particular day and session. We note that due to the data collection procedure and sensor/meter installation by the physical facilities department at the university, the time interval at which data is collected is periodic and not uniform.

We next discuss the trends in data. Table 1 shows the average percentage usage for the different types of equipment. We observe that usage of recreational facilities is lower on weekends as compared to weekdays, with Free Weights are Strength Machines being used higher than Cardio and Synrgy360. Figure 1 shows the percentage of equipment usage for 80 time steps for all four machines. We can see that usage varies over time, thus making the prediction problem an interesting one. We also observe from the data that the recreational facility at our university is generally overprovisioned, which emphasizes the importance of making accurate predictions such that they can be leveraged for optimal resource allocation.

Table 1: Average machine usage

Machine	All days	Weekdays	Weekends
Cardio	19%	19%	18%
Free Weights	26%	27%	24%
Strength Machines	22%	23%	20%
Synrgy360	19%	20%	17%

Figure 2 shows the best times to visit the recreational center based on the availability of machines with respect to time of day, day of week and month of year. From Figure 2a, we observe that usage for all machines is lowest in the morning, followed by the afternoon and highest in the evenings. This is expected as college students generally wake up late and have more classes scheduled during the afternoons. From Figure 2b, we see that the day of the week does not significantly impact the number of people using the recreational facilities, though Sunday has the least number of visitors. As expected, Figure 2c shows that months of June and July have lowest usage as most students are not on campus during the summer session. Similarly, May, August, December and January also have lower usage because the school is in session only for some days during these months.

4 MODEL

In this section, we describe *DeepFit*, a system that predicts future fitness center equipment usage based on historical data. Figure 3 shows the different components of *DeepFit*. As mentioned earlier, we collect usage data for four different equipment from a university fitness center. After data is collected, *DeepFit* pre-processes it to compute the usage statistics based on the different times of the day (i.e., morning, afternoon and evening). This data pre-processing component removes noisy data and imputes missing data by taking the average of the five previous readings for that particular day and time. This processed data stream is then fed to an LSTM-based sequence-to-sequence deep learning prediction component that uses it to generate the desired predictions. Before describing our model, we discuss the appropriateness of using sequence-to-sequence and deep learning models for the prediction task at hand.

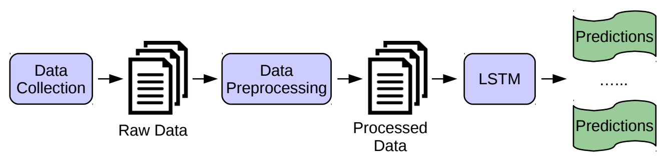
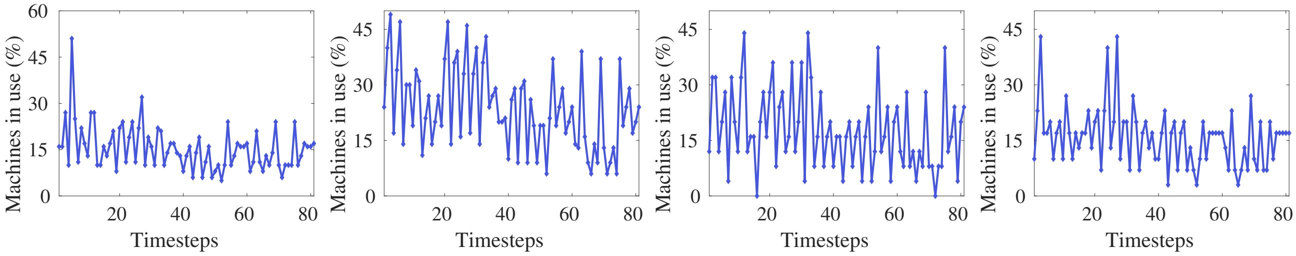


Figure 3: System Architecture

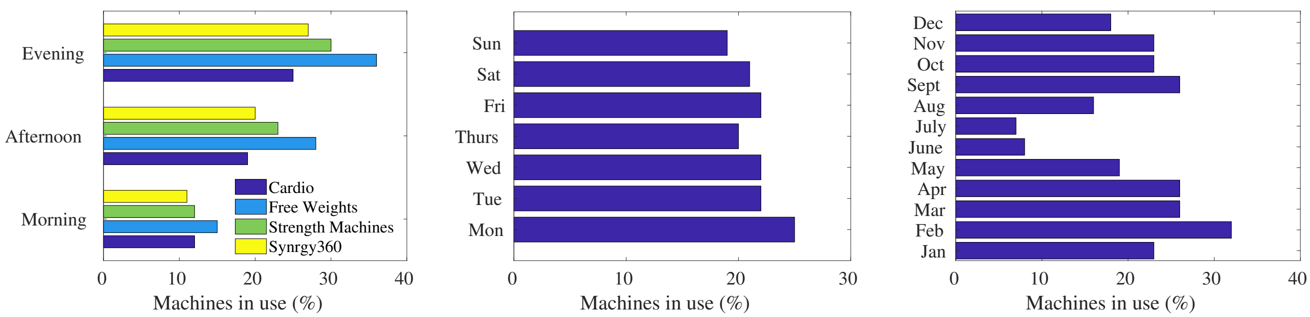
4.1 Why Sequence-to-Sequence Deep Models?

Model-based and statistical approaches usually design simple models that explain the available data and predict the future based on these models. While these hand-crafted models are easy to interpret, they often fail to capture the underlying dependencies in real-world data leading to poor prediction performance. Model-based approaches also lack the ability to learn from vast amounts of prior data. In comparison, the primary objective of deep sequence-to-sequence models is to map entire input sequences to output sequences and are ideally suited for time-series forecasting problems, similar to the one studied here. By leveraging the entire training data, during the training phase, deep sequence-to-sequence models learn the underlying correlations in the data and the complex connections between the input and output variables as the encoded signal passes the different layers of the neural network. These trained models then use their learned knowledge to generate predictions at test time by taking the current input sequence into consideration, thus providing superior performance. Deep sequence-to-sequence models have been applied to a wide variety of problems ranging from computer vision to natural language processing [26, 28]) as well as for time series predictions [18, 19].



(a) Cardio Machines (b) Free Weights (c) Strength Machines (d) Synrgy360

Figure 1: Trends in datasets



(a) Time of day (b) Day of week (c) Month of year

Figure 2: Best Gym Times

4.2 RNN Encoder-Decoder Model

The prediction component of *DeepFit* comprises of an encoder-decoder based sequence-to-sequence deep model as shown in Figure 4. The model consists of two components—an encoder and a decoder, each of which is an RNN. An RNN consists of a network of neural nodes that are organized in layers, with there being directed connections from one layer to the next. At the highest level, the encoder accepts an input sequence $x_1, x_2, ..., x_n$, which corresponds to the equipment usage in the last n time steps and generates a hidden encoded vector C which encapsulates information for the input sequence. This encoded vector is given as an input to the decoder which generates $\hat{y}_1$, $\hat{y}_2$, ..., $\hat{y}_k$, the predicted equipment usage for the next k time steps.

Internally, at each time step t, an RNN consists of a hidden state h_t that gets updated based on the input x_t and the previous hidden state (i.e., h_{t-1}) using some non-linear function f. h_t serves as memory and after the entire input sequence is read, the hidden state is the summary C capturing the information of the entire input sequence. This summary C is then used by the decoder to generate the output sequence by predicting the next value y_t given the hidden state. We use ReLU activation function after each decoder output to prevent prediction of negative equipment usage values.

In the standard RNN architecture, the neural nodes are usually composed of basic activation functions such as tanh and sigmoid. During the training phase, the weights are learned by the back-propagation algorithm that propagates errors through the network. However, the use of these basic activation functions can cause RNNs to suffer from the vanishing/exploding gradient problem that causes the gradient to have either infinitesimally low or high values, respectively. This prevents RNN from being able to learn long-term dependencies based on the data. To overcome this problem, we use LSTM cells as the basic cell in both encoder and decoder to capture and store relevant long-term temporal dependencies in the data. LSTM cells circumvent the well-known vanishing/exploding gradient by incorporating the ability to 'forget'.

Figure 5 shows the architecture of an LSTM cell. An LSTM cell consists of three gates— input gate, output gate, and forget gate. The input gate decides which information is relevant in the current step, while the output gate determines what the next hidden state is. The forget gate decides which information is relevant from the previous steps and helps to handle long-term dependencies. The gates are represented by the following equations.

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ c_t &= f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{1}$$

where, x_t is the input vector to the LSTM unit, i_t, o_t and f_t are the input gate's activation vector, output gate's activation vector

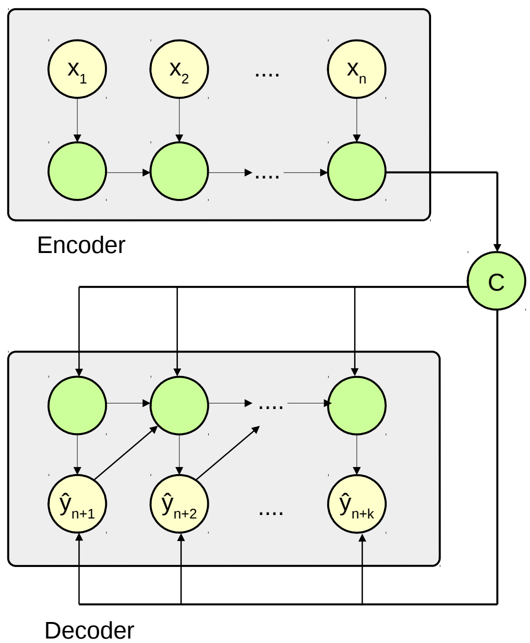


Figure 4: Sequence-to-Sequence RNN Model

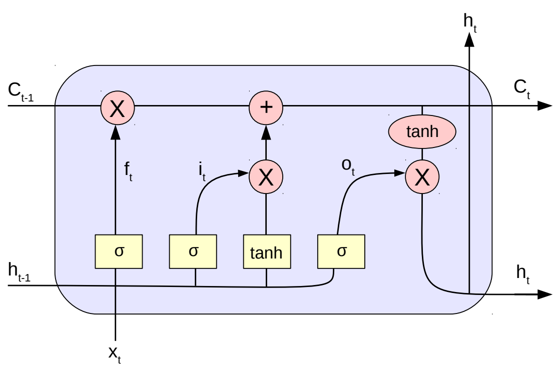


Figure 5: LSTM Cell

and forget gate's activation vector respectively. h_t is the hidden state vector, and c_t is the cell state vector. W are the weight matrices and b are bias vector parameters that are learned during training. σ and denotes application of the logistic sigmoid function and hyperbolic tangent function, respectively. $\odot$ denotes element-wise multiplication. We note that the nodes shown in Figure 4 do not correspond to LSTM cells in the architecture. They denote the abstract hidden state as the encoded signal passes through the network. We refer the reader to Goodfellow *et al.* for more details [9].

4.3 Training and Implementation Details

We split the data into two parts— 75% for training and 25% for testing and use TensorFlow[1] for implementing the deep learning model. We use equipment usage data for the past 2 weeks to predict 1 week into the future. We train our models on Spiedie, a high computing cluster available at our university. The configuration used on the cluster for all experiments is 4 cores and 8 GB RAM.

At training time, the encoder and decoder are trained jointly using the backpropagation algorithm. We adopt unguided training as the training methodology. In unguided training, the decoder uses the previous predicted output value as an input to the next step of the decoder. One of main benefits of unguided training is that it enables a better exploration of the state space, which results in superior prediction performance at test time. At both training and test times, for a given equipment usage value, we use a sliding window of one step to obtain the input sequences. This ensures that we achieve the maximum overlap of sequences used. We incorporate L2 regularization in our model to minimize overfitting.

We experiment extensively with the hyper-parameters such as number of stacked layers, numbers of hidden units in each layer, the length of the input and output sequences and the learning rate. Based on our experiments, we observe that overall 1 layer with 10 hidden units generalizes best across different types of fitness equipment and provides overall best performance. We use a learning rate of 0.01 and train the model for 500 epochs. We note that the optimal set of parameters is dependent strongly on the problem being investigated and the dataset in consideration. We present a discussion on hyper-parameter initialization and their effect on performance and our rationale for choosing this specific parameter configuration for our experiments in Section 5.4.

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

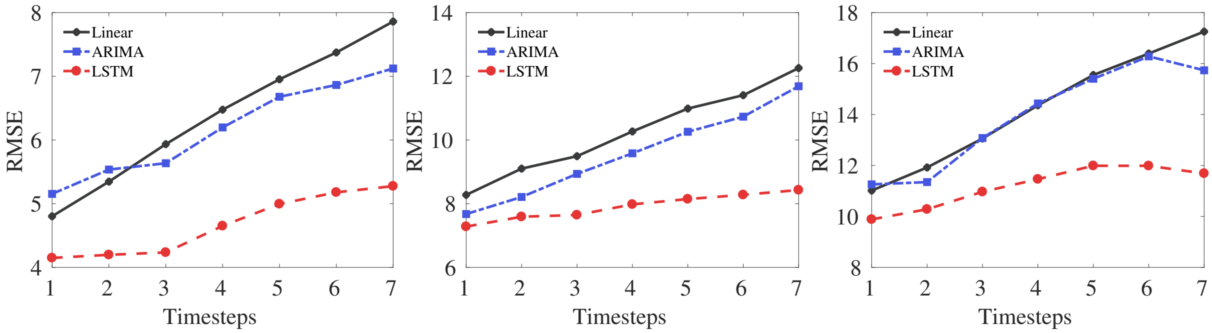
5 EXPERIMENTAL RESULTS

In this section, we demonstrate the superior prediction performance of *DeepFit* by comparing it with two baseline approaches, linear regression and ARIMA. The code for our model and the pre-processed data is available in [1]

(1) **Linear Regression** - It is a statistical model that produces the best fit straight line based on the data.
(2) **ARIMA (p, d, q)** - Auto-Regressive Integrated Moving Average, popularly known as ARIMA is a statistical model that comprises of three terms. The first term is the Autoregressive term (AR), the second is the differencing term (I) and the third is the moving average term (MA). The parameters for AR, I and MA terms are specified by p, d and q, respectively. p denotes the number of the past samples used for predicting future values, d denotes the number of times the differencing operation is performed to make a series stationary, while 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,

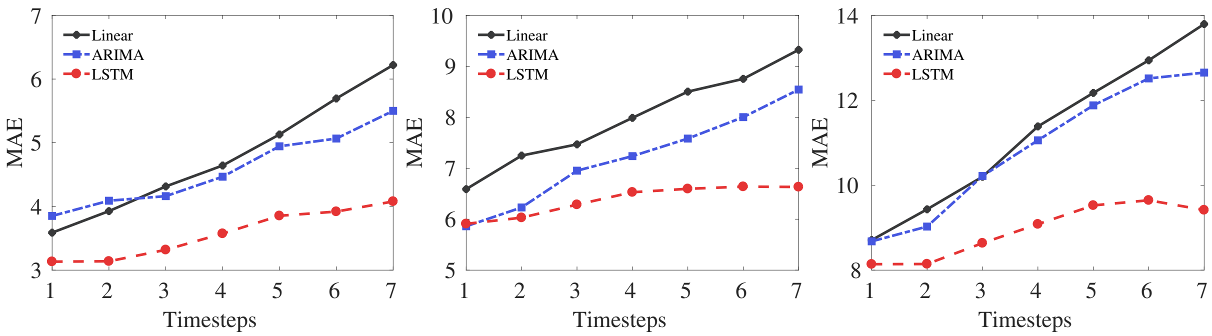
$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d x_t = (1 - \sum_{i=1}^{q} \theta_i L^i) e_t \tag{2}$$

where x_t corresponds to the equipment usage values, ϕ_i is the auto-regressive parameter, θ_i is the moving average parameter, e_t are the error terms and L is the lag term.



(a) Morning (b) Afternoon (c) Evening

Figure 6: Free Weights: RMSE



(a) Morning (b) Afternoon (c) Evening

Figure 7: Free Weights: MAE

In our experiments, we use the Auto-ARIMA toolkit[2] in python that picks the optimal values for p, d, and q after searching through different combinations of the parameters. Linear regression and ARIMA do not require training as they are statistical models. Equipment usage for the past 2 weeks is used to predict 1 week into the future.

The main metrics used in our evaluation are root mean squared error (RMSE) and mean absolute error (MAE) which are given by Equations 3 and 4, respectively.

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^{h}\left(\hat{y}_{ij} - y_{ij}\right)^2}{h}} \tag{3}$$

$$MAE_j = \frac{\sum_{i=1}^{h}|\hat{y}_{ij} - y_{ij}|}{h} \tag{4}$$

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

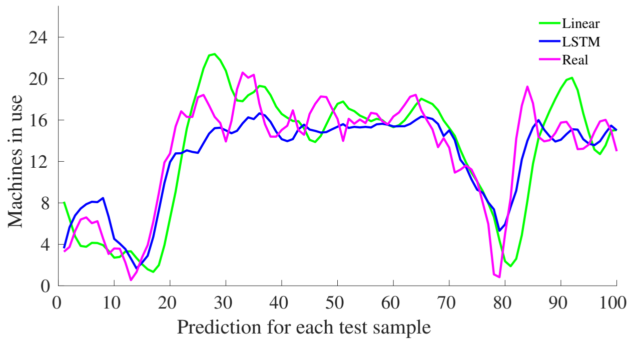
5.1 RMSE

In this subsection, we discuss the RMSE performance of *DeepFit* in comparison to the baselines. Figure 6 shows the results for Free Weights for all three sessions (i.e., morning, afternoon and evening). We observe from the figure that *DeepFit* significantly outperforms the baselines for each session. We also observe that *DeepFit* is able to make better predictions into the future as its RMSE values only increase gradually. In contrast, the prediction performance of linear regression and ARIMA become considerably worse as they predict further into the future. We attribute the superior performance of *DeepFit* to its sequence-to-sequence modeling aspect that takes the entire input sequence into account to predict the output. Table 2 shows the RMSE values calculated as an average over predictions for 7 time steps for the four machines. We see that *DeepFit* outperforms the baselines for all machines and all three sessions. *DeepFit* provides an average performance improvement of 16% over ARIMA and 18% over linear regression.

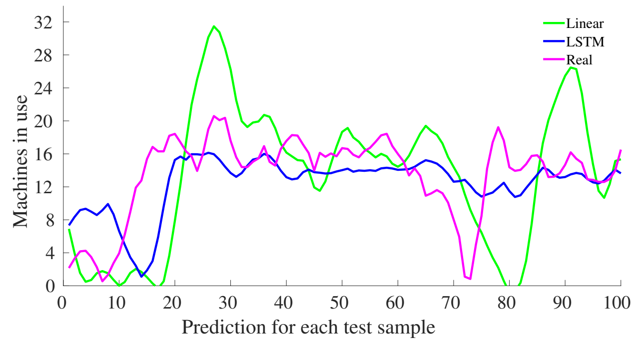
5.2 MAE

In this subsection, we discuss the MAE results. Figure 7 shows the results for Free Weights for the three sessions. We observe that *DeepFit* significantly outperforms the baselines with respect to MAE as well. We also observe that the trends for MAE are similar to RMSE. Table 3 shows the average MAE over predictions for 7 time steps for all machines. Once again, we observe that *DeepFit* achieves better performance than the baselines. *DeepFit* shows a performance improvement of 14% over ARIMA and 20% over linear regression on average.

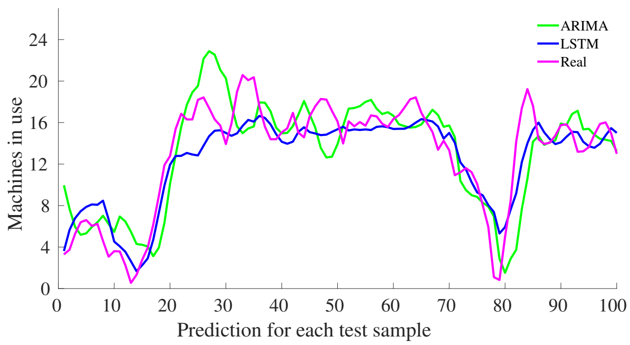
[2]https://pypi.org/project/pyramid-arima/



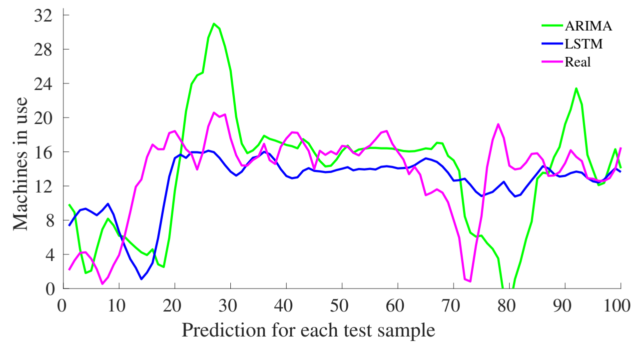
(a) LSTM vs Linear: Time Step 1



(b) LSTM vs Linear: Time Step 7



(c) LSTM vs ARIMA: Time Step 1



(d) LSTM vs ARIMA: Time Step 7

Figure 8: Qualitative Results: Free Weights(Morning Session)

Table 2: RMSE

Machines	Session	LSTM	ARIMA	LR
Cardio	Morning	3.8	4.3	4.4
	Afternoon	5.55	7.13	6.85
	Evening	7.86	9.12	9.55
Free Weights	Morning	4.67	6.17	6.4
	Afternoon	7.91	9.59	10.26
	Evening	11.18	13.94	14.23
Strength Machines	Morning	1.82	2	2.1
	Afternoon	3.81	5.22	4.48
	Evening	3.71	4.14	4.54
Synrgy360	Morning	2.44	2.59	2.85
	Afternoon	3.62	4.17	4.52
	Evening	4.07	4.67	4.8

Table 3: MAE

Machines	Session	LSTM	ARIMA	LR
Cardio	Morning	3.01	3.42	3.48
	Afternoon	4.17	5.1	5.29
	Evening	5.25	6.05	6.65
Free Weights	Morning	3.58	4.59	4.79
	Afternoon	6.38	7.21	7.99
	Evening	8.95	10.87	11.24
Strength Machines	Morning	1.28	1.41	1.52
	Afternoon	2.87	3.29	3.42
	Evening	2.76	3.18	3.55
Synrgy360	Morning	1.72	1.89	2.13
	Afternoon	2.79	3.19	3.57
	Evening	2.96	3.54	3.66

5.3 Qualitative Results

In this subsection, we present qualitative results to visually demonstrate the superior prediction performance of *DeepFit*. Figures 8a and 8b show the predictions for LSTM and linear regression for Free Weights for the morning session for time step 1 and time step 7, respectively. We observe that linear regression shows poor prediction performance than LSTM. In comparison to linear regression that predicts the future based on the trend obtained from the previous values, LSTM learns the mapping between input and output sequences, which enables it to obtain better performance. We also see from Figure 8b that the performance of linear regression becomes worse by time step 7, but LSTM still manages to capture the underlying structure in data. Figures 8c and 8d show predictions for LSTM and ARIMA for Free Weights for the morning session for time step 1 and time step 7, respectively. Similar to linear regression, we observe that ARIMA's prediction is worse than LSTM and its performance deteriorates significantly by time step 7.

5.4 Discussion on Hyper-parameters

In this subsection, we investigate the impact of the different hyper-parameters on the performance of *DeepFit*. We vary four hyper-parameters — number of stacked layers, number of hidden units, input sequence length and learning rate, and discuss the rationale behind choosing the default parameters. Figures 9 and 10 show RMSE results for *DeepFit* for Free Weights for the afternoon and evening sessions respectively. We observe from Figures 9a and 9b that the configuration with 1 stacked layer and a learning rate of 0.01 gives the best performance. Similarly, we see from Figures 10a and 10b that training the model with 10 hidden units and input sequence length of 14 gives the highest performance scores.

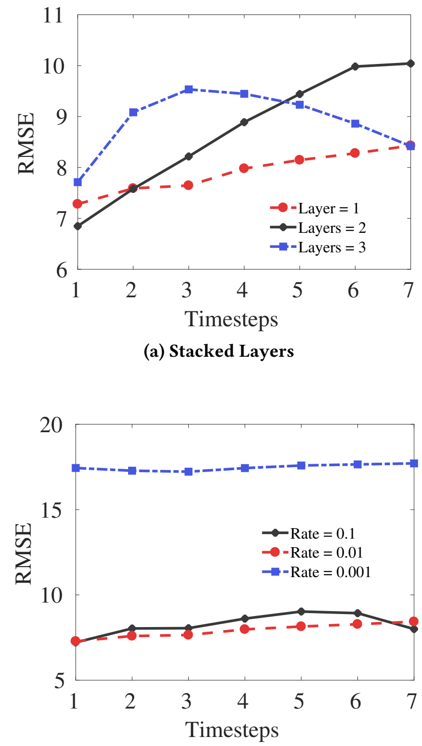
Table 4 shows average improvement of *DeepFit* over the baselines with respect to RMSE calculated over all machines and all sessions. We observe that as the number of stacked layers increase, the performance of *DeepFit* decreases. Similarly, performance of *DeepFit* increases as we increase the number of hidden units to 10. However, increasing the number of hidden units beyond 10 deteriorates its performance. We hypothesize the limited variation in the data as the primary reason as to why a simple model with 1 layer and 10 units gives the best performance. We also train the model by varying the sequence length from 2 to 4 weeks (i.e., sequence length 14, 21, and 28). We observe from the table that giving more than two weeks as input does not help the model learn additional information. Similarly, we find that a learning rate of 0.01 gives the best performance and increasing or decreasing the learning rate deteriorates the overall performance. In fact a learning rate of 0.0001 provides worse performance than the baselines. Our analysis throws light on the importance of choosing hyper-parameters appropriately and provides the setup for performing hyper-parameter tuning for other similar datasets or problems.

Table 4: Hyper-parameters

Hyper-parameters	Values	Average improvement over	
		ARIMA	LR
Stacked Layers	**1**	**15.62%**	**18.14%**
	2	12.62%	15.25%
	3	9.84%	12.35%
Hidden Units	5	15.52%	17.98%
	10	**15.62%**	**18.14%**
	15	10.93%	13.56%
Sequence Length	**14**	**15.62%**	**18.14%**
	21	6.55%	7.56%
	28	9.86%	16.26%
Learning Rate	0.1	13.33%	15.83%
	0.01	**15.62%**	**18.14%**
	0.001	-6.25%	-3.03%

5.5 Discussion on adding features

In our experiments so far, the LSTM-based sequence-to-sequence deep learning model used in *DeepFit* only considers previous equipment usage data to predict the future. In this subsection, we describe how we augment *DeepFit* by including additional features in the base LSTM model and investigate the performance benefits of this



(a) Stacked Layers

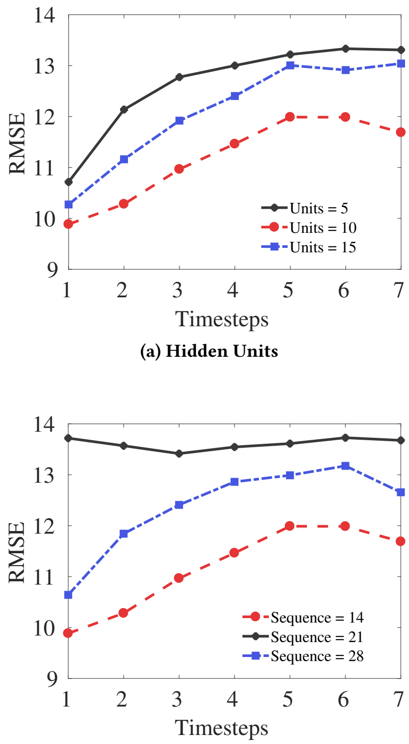
(b) Learning Rate

Figure 9: Varying hyper-parameters: Free Weights (Afternoon Session)

enhanced model. To this end, we identify two features that can be incorporated in the base LSTM model, namely, *i)* whether school is in session (*Academic Term*) and *ii)* month of the year (*Month*). Recall from Figure 2c that equipment usage changes significantly based on the month, whether the school is in session, and for how long the school is in session during a particular month.

We study performance gains of including each of these features individually as well as jointly in the enhanced model. While at a cursory glance, it might appear that including both features will be more useful, this may not be the case depending on the underlying correlation between the features and the complexity in the data. We obtain the information regarding whether the school is in session from the university academic calendar. The feature *Academic Term* takes values 1 or 0, where 1 denotes that the school is in session. The feature *Month* takes values from 1 to 12, where 1 denotes the month of January. To evaluate the performance of the enhanced model, we require train and test data spanning all the months. Therefore, instead of using the first 75% data at training time and the remaining 25% at test time, we generate a new training dataset by randomly sampling 75% sequences from the original dataset. The remaining 25% sequences from the original dataset form the new test dataset. This process ensures that we have representative train and test samples spanning all the months.

Table 5 shows the percentage improvement, obtained as an average over 7 time steps for the enhanced LSTM model over the base LSTM model (i.e., without additional features) when we add



(a) Hidden Units

(b) Sequence Length

Figure 10: Varying hyper-parameters: Free Weights (Evening Session)

Table 5: Features: Average improvement of Enhanced *DeepFit* over base *DeepFit*

Machines	Session	Feature *Academic Term*	Feature *Month*	Feature *Academic Term & Month*
Cardio	Morning	3.04%	7.75%	**10.74%**
	Afternoon	**19.16%**	9.39%	3.63%
	Evening	3.01%	4.51%	**17.24%**
Free Weights	Morning	10.34%	**11.73%**	9.3%
	Afternoon	4.19%	**7.47%**	3%
	Evening	2.53%	7.92%	**13.02%**
Strength Machines	Morning	**11.04%**	5.75%	9.45%
	Afternoon	6.86%	**11.4%**	3.07%
	Evening	6.02%	5.82%	**15.19%**
Synrgy360	Morning	2.21%	5.5%	**10.75%**
	Afternoon	**2.23%**	1.36%	0.05%
	Evening	**15.39%**	12.41%	15.35%

individual features *Academic Term* and *Month*, and when we add the features together. We observe from the table that depending on the equipment and the session, one version of the enhanced model can outperform the other and it is difficult to pick a clear winner. Overall, from the table, we observe that adding *Academic Term* improves performance by up to 19%, adding *Month* improves performance up to 12%, and adding them together shows performance improvement of up to 17%.

Table 6: Features: Average improvement of Enhanced *DeepFit* over ARIMA

Machines	Session	Feature *Academic Term*	Feature *Month*	Feature *Academic Term & Month*
Cardio	Morning	32.76%	36.03%	38.1%
	Afternoon	39.75%	32.47%	28.18%
	Evening	35.49%	36.49%	44.95%
Free Weights	Morning	34%	35.03%	33.23%
	Afternoon	28.14%	30.6%	27.25%
	Evening	28.18%	32.15%	35.91%
Strength Machines	Morning	36.28%	32.49%	35.14%
	Afternoon	36.46%	39.55%	33.87%
	Evening	19%	18.83%	26.91%
Synrgy360	Morning	24.79%	27.31%	31.35%
	Afternoon	35.79%	35.22%	34.36%
	Evening	31.06%	28.64%	31.04%

Table 6 shows the average percentage improvement of the enhanced LSTM model over ARIMA. We observe that adding *Academic Term* and *Month* individually improves performance between 19% and 40%, whereas adding both features together improves performance between 27% to 45%. Additionally, comparing with linear regression, we observe that adding the features individually improves performance by 35% on average, while adding them together shows 36% improvement on average. Our results thus demonstrate that augmenting the LSTM model with these additional features (if available) can significantly improve the prediction performance.

6 CONCLUSION

In this paper, we investigated fitness center equipment usage prediction problem using data collected from a university campus fitness center. We designed *DeepFit*, an LSTM based encoder-decoder recurrent neural network model that accurately models and predicts equipment usage based on historical data. Our experimental evaluation shows that by successfully encoding and capturing dependencies in the underlying equipment usage data, *DeepFit* achieves superior prediction performance (17% improvement on average in RMSE and MAE) for all kinds of fitness equipment in comparison to the linear regression and ARIMA baselines. We also demonstrated that augmenting the base LSTM model with features such as whether the school is in session and the month of the year improves the performance of *DeepFit* over linear regression and ARIMA by 35% and 32%, respectively. Additionally, *DeepFit* requires limited data and computational resources at test time making it a practically attractive and viable system.

REFERENCES

[1] [n.d.]. Code and data. https://bitbucket.org/aditakulkarni/equipment_usage_prediction.
[2] [n.d.]. U.S. Department of Health & Human Services. https://www.hhs.gov.
[3] Brandon Ballinger, Johnson Hsieh, Avesh Singh, Nimit Sohoni, Jack Wang, Geoffrey Tison, Gregory Marcus, Jose Sanchez, Carol Maguire, Jeffrey Olgin, and Mark Pletcher. 2018. DeepHeart: Semi-Supervised Sequence Learning for Cardiovascular Risk Prediction. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/

view/16967/15916
[4] Gissella Bejarano, David DeFazio, and Arti Ramesh. 2019. Deep Latent Generative Models For Energy Disaggregation. In *Thirty-Third AAAI Conference on Artificial Intelligence*.
[5] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. A Neural Attention Model for Urban Air Quality Inference: Learning the Weights of Monitoring Stations. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16607
[6] 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.
[7] Scott Forrester. [n.d.]. The benefits of campus recreation. ([n. d.]).
[8] Ian Fox, Lynn Ang, Mamta Jaiswal, Rodica Pop-Busui, and Jenna Wiens. 2018. Deep Multi-Output Forecasting: Learning to Accurately Predict Blood Glucose Trajectories. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, New York, NY, USA, 1387–1395. https://doi.org/10.1145/3219819.3220102
[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*.
[10] Yu Guan and Thomas Plötz. 2017. Ensembles of Deep LSTM Learners for Activity Recognition Using Wearables. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 2, Article 11 (June 2017), 28 pages. https://doi.org/10.1145/3090076
[11] Nils Y. Hammerla, Shane Halloran, and Thomas Plötz. 2016. Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 1533–1540. http://dl.acm.org/citation.cfm?id=3060832.3060835
[12] Mohd. Hafiz Hasan, Pascal Van Hentenryck, Ceren Budak, Jiayu Chen, and Chhavi Chaudhry. 2018. Community-Based Trip Sharing for Urban Commuting. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16915
[13] Wenchao Jiang and Zhaozheng Yin. 2015. Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. ACM, New York, NY, USA, 1307–1310. https://doi.org/10.1145/2733373.2806333
[14] Hao-Cheng Kao, Kai-Fu Tang, and Edward Chang. 2018. Context-Aware Symptom Checking for Disease Diagnosis Using Hierarchical Reinforcement Learning. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17130
[15] Jack Kelly and William Knottenbelt. 2015. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys '15)*. ACM, New York, NY, USA, 55–64. https://doi.org/10.1145/2821650.2821672
[16] Huijie Lin, Jia Jia, Quan Guo, Yuanyuan Xue, Qi Li, Jie Huang, Lianhong Cai, and Ling Feng. 2014. User-level Psychological Stress Detection from Social Media Using Deep Neural Network. In *Proceedings of the 22Nd ACM International Conference on Multimedia (MM '14)*. ACM, New York, NY, USA, 507–516. https://doi.org/10.1145/2647868.2654945
[17] Qi Liu, Yue Zhang, Zhenguang Liu, Ye Yuan, Li Cheng, and Roger Zimmermann. 2018. Multi-Modal Multi-Task Learning for Automatic Dietary Assessment. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16235
[18] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang. 2015. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (April 2015), 865–873. https://doi.org/10.1109/TITS.2014.2345663
[19] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2017. TimeNet: Pre-trained deep recurrent neural network for time series classification. *CoRR* abs/1706.08838 (2017). arXiv:1706.08838 http://arxiv.org/abs/1706.08838
[20] 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 (UbiComp '16)*. ACM, New York, NY, USA, 940–945. https://doi.org/10.1145/2971648.2971731
[21] N. Pathak, A. Ba, J. Ploennigs, and N. Roy. 2018. Forecasting Gas Usage for Big Buildings Using Generalized Additive Models and Deep Learning. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. 203–210. https://doi.org/10.1109/SMARTCOMP.2018.00092
[22] Liangying Peng, Ling Chen, Zhenan Ye, and Yi Zhang. 2018. AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2 (07 2018), 1–16. https://doi.org/10.1145/3214277
[23] Asif Salekin, Jeremy W. Eberle, Jeffrey J. Glenn, Bethany A. Teachman, and John A. Stankovic. 2018. A Weakly Supervised Learning Framework for Detecting Social Anxiety and Depression. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 81 (July 2018), 26 pages. https://doi.org/10.1145/3214284
[24] C. Sideris, H. Kalantarian, E. Nemati, and M. Sarrafzadeh. 2016. Building Continuous Arterial Blood Pressure Prediction Models Using Recurrent Networks. In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*. 1–5. https://doi.org/10.1109/SMARTCOMP.2016.7501681
[25] Mengying Sun, Inci M. Baytas, Liang Zhan, Zhangyang Wang, and Jiayu Zhou. 2018. Subspace Network: Deep Multi-Task Censored Regression for Modeling Neurodegenerative Diseases. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, New York, NY, USA, 2259–2268. https://doi.org/10.1145/3219819.3219966
[26] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
[27] Alireza Abedin Varamin, Ehsan Abbasnejad, Qinfeng Shi, Damith C. Ranasinghe, and Hamid Rezatofighi. 2018. Deep Auto-Set: A Deep Auto-Encoder-Set Network for Activity Recognition Using Wearables. In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*. ACM, New York, NY, USA, 246–253. https://doi.org/10.1145/3286978.3287024
[28] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to Sequence – Video to Text. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15)*. IEEE Computer Society, Washington, DC, USA, 4534–4542. https://doi.org/10.1109/ICCV.2015.515
[29] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. AAAI Press, 3995–4001. http://dl.acm.org/citation.cfm?id=2832747.2832806
[30] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16069/15978