Using Real-World Store Data for Foot Traffic Forecasting

Soheila Abrishami, Piyush Kumar
Department of Computer Science
Florida State University
Tallahassee, FL
Email: abrisham, piyush@cs.fsu.edu

***Abstract*—Time series forecasting is a fundamental task in machine learning and data mining. It is an active area of research, especially in applications that have direct impact on the real-world. Foot traffic forecasting is one such application, which has a direct impact on businesses and non-profits alike. In this paper, we propose and compare different prediction models for foot traffic forecasting. Our foot traffic data has been collected from wireless access points deployed at over 65 businesses across the United States, for more than one year. We validate our work by comparing to state-of-the-art time series forecasting approaches. Results show the competitiveness of our proposed method in comparison to our previous work and state-of-the-art procedures for time series forecasting.**

***Keywords*-Time series forecasting; Foot traffic Forecasting; Regression**

I. Introduction

In the research community of machine learning and data science, time series forecasting has a prominent role [1]–[4]. It is important because prediction problems that involve time are ubiquitous. Time series forecasting methods are used in sales prediction [5], stock forecasting [6], weather forecasting [7], foot traffic prediction [8] and earthquake predictions [9]. Despite the fact, that time series forecasting is used in many areas, improving quality of forecasts still remains a challenge. Forecasting accuracy can be impacted by different factors such as time, weather conditions, economic factors, and outliers. Several methods such as Winters exponential smoothing [10], the Autoregressive Integrated Moving Average (ARIMA) model [11], multiple regression, and artificial neural networks (ANNs) have been the most widely used approaches to time series forecasting. These models can capture trend and seasonal fluctuations present in time series data [12]. However, all these methods have shown difficulties and limitations. Therefore, it is necessary to investigate further how to improve the quality of forecasts.

In this paper we focus on foot traffic forecasting which is a particular application of time series forecasting. Based on a business dictionary, foot traffic is "a term used to describe pedestrian visitors to a business or commercial site". Foot traffic forecasting is helpful with budgeting, planning, staffing and adjusting product stock levels [13]–[15]. These ultimately are directly correlated with business profits, which is the main driver for most companies today.

Compared to stock forecasting, food sales forecasting, and similar problems, the subject of foot traffic prediction has received limited research focus. The main reason for insufficient research in foot traffic forecasting is lack of real world data. To our knowledge there does not exist a public dataset that has foot traffic per minute, hour, or day, for a large number of stores. Due to lack of data, restaurants and physical stores have not had a reliable way of predicting foot traffic. To collect large scale and reliable foot traffic data, a comprehensive system with sensors is required. One way of inferring foot traffic data is using smart devices with WiFi [16]–[18]. Wireless access points can detect the presence and relative position of smart devices with WiFi, which can then be used to collect foot traffic data.

In [8], we described such a scalable data collection system that we designed and used for our experiments. This paper uses a more reliable dataset, presents our improvements in algorithms and improves accuracies compared to our previous work [8].

In this research, we aim to predict foot traffic of different stores, a day or week in advance. Our work should scale to larger windows of time (for example monthly or yearly forecasting), given more data. We adopt different learning models to capture the different behavior of foot traffic. We use feature engineering along with machine learning and deep learning techniques to learn the bearing information for the forecasting models. Figure 1 summarizes our entire learning and prediction architecture. Our results improve on our previous results.

Data for our work was gathered by Cisco Meraki wireless access points that are installed in different stores, such as gyms, restaurants, and bars. WiFi access points installed in businesses detect smartphones with WiFi turned on, whether or not a user connects his or her device to the wireless network. However, the device's presence can only be detected while the device is within range of the network [16]. After preprocessing the raw data, we do feature engineering to map the preprocessed data to a feature space; then, we build multiple prediction models to predict foot traffic of the next 7 days in advance.

Our prediction models are built using different machine learning approaches. We apply regression algorithms such as Random Forest Regression [19], neural networks such

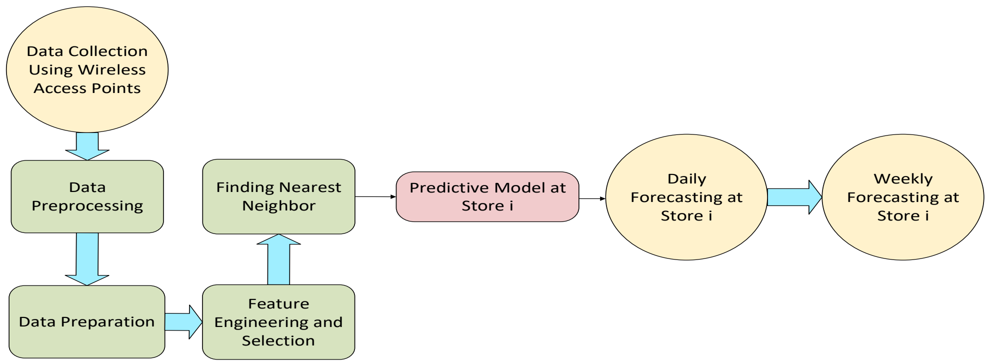

Figure 1. Structure of our proposed system

as Long Short-Term Memory network (LSTM) [20] and Sequence to Sequence learning [21]. One of our models performs Nearest Neighbor [22] search to find the NN of each store, and uses this data for building the prediction model. Besides, we use an additive model Prophet [23] published by Facebook's data science team to compare our proposed method.

II. Data Collection System

For this research, raw data is collected from Cisco Meraki Wireless Access Points (AP) by Bloom Intelligence. The AP's are installed at customers of Bloom Intelligence such as gyms, bars, and restaurants. The raw data is processed and classified to extract foot traffic and other engagement metrics for the location. This data is then made available to Bloom Intelligence customers through an analytics dashboard. The details regarding the data collection and preprocessing phase can be found in our previous paper [8].

This study uses daily foot traffic of 65 different retails locations that are collected from October 2016 to December 2017. Using Apache Spark, this substantial dataset is then preprocessed to compute visitors and passerby information [8] for each calendar day in this period. Figure 2 represents daily foot traffic time series for a fast food, bar, restaurant, and coffee shop. There are several effects and obstacles that can be seen in real-world time series: 1) weekly cycle , 2) trend change 3) holidays and special events 4) outliers, 5) different starting points, which most of the real datasets have. The mentioned effects can be observed in figure 2. For example, data collection for each store typically starts at a different time and date (5) - as can be seen in Figure 2. In this figure, the fast food data was collected beginning October 2016, whereas the other types of stores shown, start data collection in January 2017.

III. Forecasting Model

Before building the prediction model, we prepare and normalize the data. Then, feature engineering is applied to map the prepared data into a feature space. And finally, the learning algorithms such as regression algorithms and neural networks use the representation of the data in the feature space to learn a model for predicting the foot traffic of different stores. For each store, we have a separate predictive model which is built from the gathered data in the store and its nearest neighbor. In section III-C3, we discuss how we find the nearest neighbor of each store and use the data of nearest neighbor to build more accurate forecasting model.

A. Data Preparation and Normalization

In order to use machine learning for building the prediction models, we divide the data into overlapping windows of time, and then use it to predict the next days (or weeks). Our validation results support this fact, table III. Foot traffic also depends on the time of year, day of week, and time of day. For instance, in a restaurant, traffic may be higher on weekends, or at specific times such as during periods of lunch or dinner. Therefore, we can argue that foot traffic on hour h and day d, where $0 \leq h \leq 23$ and $0 \leq d \leq 6$, is close to foot traffic on hour h and day d of last weeks. For example, if we want to predict foot traffic on Monday at 8 pm, we should look at the foot traffic on the previously observed Mondays at 8 pm.

Thus, based on this logic, we can define a function learning problem which predicts daily data. Let foot traffic on day d be equal to v_d. Our function f takes N inputs, and will output the foot traffic for a day. The N inputs to f correspond to the foot traffic from the same day in previous N weeks. Here each input contains daily data. Formally:

$$f(v_{d-w}, v_{d-2w}, v_{d-3w}, \ldots, v_{d-Nw}) = v_d$$

where $w = 7$. Now we can easily use our data to solve for f using regressions and neural network of different types.

We normalized the prepared data using feature scaling to $(-1, 1)$. After getting results from models on test data used in the section IV, we denormalized the predicted values and compare with the actual values.

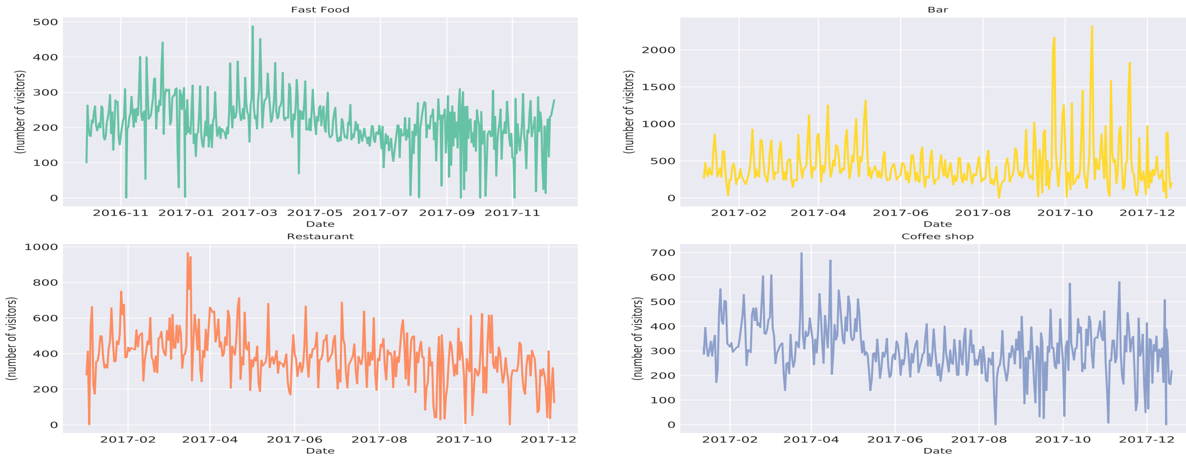

Figure 2. The plots show number of visit per day for a fast food, bar, restaurant and, coffee shop.

B. Feature Engineering and Selection

After data preparation and normalization, we will have N features corresponding to daily foot traffic from the same day in previous weeks. Our output will be a prediction for one day of next week. We optimize the number of previous weeks (N) using our recent work (See Figure 2, Section 4 of [8]). The numbers for our current dataset come out to be 8 features for 8 previous weeks using this method. In another word, for predicting daily foot traffic of a Monday our prediction function f takes the daily foot traffic on Mondays at 8 previous weeks.

Since our data is a time series, we use time series feature extraction techniques to map the data into a high dimensional feature space. Feature extraction from time series is a well-studied problem, and we use the `FRESH` approach described by [24] to map the prepared data into the feature space. When $N = 8$, there were 788 features extracted using the `FRESH`. The extracted features include basic and advanced characteristics of the time series. Basic features are such as the number of peaks, the average or maximal value and complex attributes are such as the time-reversal symmetry statistic.

We tried and used `SelectKBest` and `RFECV`, both implemented in ScikitLearn to rank all the features. For the neural network approach in section III-C2, we use the 10 top features which were produced from the `SelectKBest` to lead to lower training and testing times. Besides, we observed having more features does not improve the accuracy significantly. For the regression approach in section III-C1, we use `RFECV`, which chooses 128 features on average. Some selected features for a sample store are listed based on their rank in table I.

In [8], we discussed in details some factors such as holiday, special events and weather that may affect the foot traffic. In order to include holiday into our learning models, we use categorical variables. For instance, normal days are coded as 0, regular holidays as 1 and festival days as 2. This way is effective with large data sets, but we have only around a year of data, and there are just a few holidays in one year. We conjecture that our current approach will do better, given more data.

C. Learning Models

After data preparation and feature engineering phase we want to predict daily foot traffic of different stores. As the foot traffic (number of people who are visited) is a continuous variable, we formulate the problem as a regression problem. We use several regressors and neural network for building the prediction model.

1) Random forest Regression: The first approach we applied for building our learning model is regression methods. In our previous research [8], we tried and tested a few regressors. The results showed that there is no significant difference between the errors derived from regression algorithms. So in this project, we only report the results of Random Forest Regression. We use the data from the feature engineering and selection phase to build the prediction model. We use Random Forest Regression implemented in Python package ScikitLearn [1]

2) LSTM: Another approach that we use for building our learning model is a neural network. We use Long Short-Term Memory (LSTM) [20], a specific recurrent neural network (RNN) architecture that is well-suited for time series prediction with time lags of unknown size. Compared to simpler vanilla RNN implementations, LSTMs are generally considered more robust for long time series [20]. We used Keras framework [2] for our experiments, which is a high-level neural networks API, written in Python.

In this research, we use the network presented in figure 3 as our reference architecture. Different stores may have

[1] http://scikit-learn.org/stable/
[2] https://keras.io

Table I
ENGINEERED AND EXTRACTED FEATURES FROM THE DAILY FOOT TRAFFIC OF 8 OBSERVED WEEKS

Feature Name	Description
abs_energy	Absolute energy of the time series which is the sum over the squared values
mean	The mean of the time series values
median	The median of the time series values
sum_values	Calculates the sum over the time series values
quantile	The q quantile of time series x. This is the value of x such that q% of the order values from x are lower than.
fft_coefficient	The fourier coefficients of the one-dimensional discrete Fourier Transform for real input by fast fourier transformation algorithm

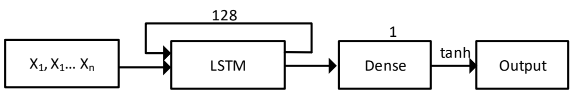

Figure 3. Network architecture used for a sample store

variant network structures. The reference architecture is for a random restaurant that uses a first layer of 128 LSTM cells, followed by final dense output. In this simple architecture, which is based on many to one LSTM, the role of the LSTM layer is to abstract a meaningful representation of the input time series. These higher-level "features" are then combined by the one dense layer in order to produce the output, in our case the predicted daily foot traffic. The activation function is chosen as tanh since the data normalized between (-1, 1) in section III-A. For our application and most of the data collected from shops, stacked layers of LSTM do not provide improvements over a single layer. Also, we tried the many to many LSTM, which did not work as well as we had anticipated, so we do not present the results of this method in our paper. We do hope to tune this method in the near future and see if we can use it for our application.

Neural Networks like LSTM need hyperparameter optimization to be useful in the real-world. We use Hyperas [3], a Python package, which is a wrapper around Hyperopt for fast prototyping with Keras models. For each store, the hyperparameter optimization is performed separately.

3) Nearest Neighbor + Random Forest: For each store, we have a separated prediction model that is built either by Random Forest Regression or LSTM. The foot traffic prediction models are built by using the data collected from each store. Another approach that we applied to train the foot forecasting model is using collected data from a shop and its nearest neighbor. For finding the nearest neighbor of stores, we tried different methods such as using Euclidean distance. The most effective approach is as follow; for store i and j, where $1 \leq i, j \leq Number\ of\ stores$, we build prediction models $Model_{i,j}$, using prepared data with $N = 8$ features, from store i and j and then applying Random Forest Regression. If for store i the model $Model_{i,k}$ has the least error, then store k will be the nearest neighbor of store i. Note that this relation is not symmetric. After finding the nearest neighbor of stores, we combine the data of each store and its neighbor, before we feed it into the predictor. This changes the input of the prediction algorithm from 8 features to 16 features. The input is interleaved for the two stores to create the 16 dimensional data. We tried many other methods of combining these two 8 dimensional vectors (for example concatenation), but this worked best for our tests. Finally, we build the foot traffic forecasting models by applying Random Forest Regression on combined data of stores and their nearest neighbor.

[3]http://maxpumperla.github.io/hyperas/

Table II
DATA COLLECTED FROM DIFFERENT STORES

Group #	Business type	Avg. foot traffic per day per store	# of stores
1	Gym	597	5
2	Coffee shop	252	3
3	Restaurant	275	21
4	Bar	357	9
5	Car wash	114	1
6	Fast food	363	26

IV. EXPERIMENTAL RESULTS

In this section, we present the results of our experiments. In order to validate the general forecasting performance of our proposed prediction models, multiple experiments were performed to explore foot traffic prediction in different types of businesses and locations with a variety of foot traffic. The data is collected from 65 different stores across the United States. The stores are categorized as gyms, fast foods, restaurants, coffee shops, bars and a car wash. For each store, the data collection system has been activated on a different date. In total, our data is collected from October 2016 to December 2017, and contains the aggregated foot traffic counts for each day in this period. Details regarding the datasets are shown in table II.

We tried out several regressors, namely, (Random Forest Regression) and strategies: `SelectKBest`, `RFECV`, Nearest Neighbor, and neural networks). The objective for this set of tests was to find the predictor that yielded the lowest mean absolute percentage error (MAPE), which is the average of the absolute percentage of each error from that test.

A. Evaluation

For time series data, model evaluation and selection is not easy. We cannot use a method like cross-validation because observations in time series are not exchangeable [23]. Hence the simplest way to overcome this problem is to use the first M months of data for training and last L months for testing, where $L < M$. But this might not be the best approach. Another approach is simulated historical forecasts (SHFs) presented in [23]. SHFs produce K forecasts at various cutoff points in the historical data. First K points in historical data are chosen. For each point the data that lie before the cutoff is used as training and data after that is used for testing. This produces K forecasting models at multiple cutoff points, and the total error is the average error of K models. This approach has some advantages and disadvantages as discussed in [23].

B. Comparison of Forecasting Models

In this section, we compare and evaluate our present methods with our previous approach, called RF [8] and Prophet, a time series forecasting method presented by Facebook's data science team [23]. In [23], Prophet is compared with other automated forecasting methods such as ARIMA [11], and Exponential smoothing [10]. Results show that Prophet has much lower prediction error compared to these baseline models. This is the reason we do not report results with baseline models in this paper.

We use mean absolute percentage error, MAPE, to compare the predictive models for all the 65 stores. We use the first 11 months for training and the last one month for testing. The number of forecasts for this experiment is set to $h = 30$ (one month).

The various prediction models that we compare are as follows:

- **Prophet:** A time series forecasting method presented by Facebook's data science team [23].
- **RF:** Applying Random Forest regression on prepared data without feature engineering [8].
- **RF+FE:** Applying Random Forest Regression on extracted features from prepared data, as discussed in section III-C1.
- **LSTM:** Applying Long Short-Term Memory neural network on prepared data without feature engineering, as described in section III-C2.
- **LSTM +FE:** Applying Long Short-Term Memory neural network on extracted features from prepared data, as described in section III-C2.
- **RF +NN:** Applying Random Forest Regression on data of each store and its neighbor, as described in section III-C3.

The results of the experiments are shown in table III. As results illustrate methods 5 (LSTM+FE), and 6 (RF+NN) have very close accuracy and both beat other methods.

Table III
MAPE COMPARISON FOR THE FOOT TRAFFIC FORECASTING MODELS

Method #	Method Name	MAPE
1	Prophet [23]	23.87%
2	RF [8]	20.08%
3	RF+FE	19.53%
4	LSTM	18.72%
5	LSTM+FE	17.54%
6	RF+NN	17.06%

Table IV
ERROR COMPARISON FOR DIFFERENT TYPE OF BUSINESS

Group #	Business type	MAPE
1	Gym	19.6%
2	Coffee shop	15.3%
3	Restaurant	17.5%
4	Bar	18.7%
5	Car wash	26.0%
6	Fast food	15.0%

Compared to our previous method, RF, presented in [8], in this research we achieved 3.02% improvement in accuracy (Row 2 and Row 6 of Table III). Our approaches outperform the Prophet presented by Facebook's data science team.

In order to compare error for different types of businesses, mean absolute percentage error is examined, for stores described in table II. As table IV illustrates, the coffee shops and fast foods have around 15% errors, while restaurants, bars, gyms and car wash have 17.5%, 18.7%, 19.6%, and 26% errors respectively. This is due to the types of stores (datasets); as table II and plot 2 show, the number of visits and foot traffic pattern for different types of businesses are varying so different accuracies are anticipated.

C. Multiple Cutoff Points

In this experiment, we use the SHF approach and compare the 6 methods mentioned in section IV-B. In order to overcome the high computational time of SHFs procedure for our application, we choose 10 stores randomly out of our 65 different shops. The randomly selected stores are from the different type of businesses. For $h = 7$ day-ahead forecast, we chose four cutoff points, on every two weeks of last two months. We call these points as cutoff points at week 2, 4, 6, and 8.

The results of the experiments are shown in Table V. The proposed methods LSTM+FE and RF+NN, achieve competitive accuracy with respect to the other models. Besides, we observe that Prophet competes with our present methods as well. However, from the approaches in section IV-B, Prophet has a high error. This we conjecture is due to the long duration of prediction ($h = 30$).

V. CONCLUSIONS AND FUTURE WORK

This paper addresses a particular application of time series forecasting, called foot traffic forecasting. Raw data for more

Table V
MAPE COMPARISON FOR THE FOOT TRAFFIC FORECASTING MODELS USING MULTIPLE CUTOFF POINTS

Method #	Method Name	MAPE
1	Prophet [23]	14.32%
2	RF [8]	15.85%
3	RF+FE	13.99%
4	LSTM	15.58%
5	LSTM+FE	13.09 %
6	RF+NN	13.25%

than one year was gathered by using wireless access points installed in more than 65 stores. The experimental result indicated our approach outperforms the state-of-the-art of time series forecasting method for this application. In our future work, we plan to combine our prediction models using metalearning strategies. Also, we will investigate more on the sequence to sequence learning of LSTM. Another aspect that needs to be explored is outliers. Outliers have a significant impact on forecasting models. In the near future, we plan to focus on this aspect by using Variational autoencoders (VAE). Another interesting direction of research is to improve the efficiency of our learning algorithms using GPUs and clusters of machines.

REFERENCES

[1] M. Qi and G. P. Zhang, "Trend time series modeling and forecasting with neural networks," in *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.*, pp. 331–337, March 2003.

[2] N. Wagner, Z. Michalewicz, S. Schellenberg, C. Chiriac, and A. Mohais, "Intelligent techniques for forecasting multiple time series in real-world systems," *International Journal of Intelligent Computing and Cybernetics*, pp. 284–310, 2011.

[3] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443 – 473, 2006. Twenty five years of forecasting.

[4] L. Li, B. A. Prakash, and C. Faloutsos, "Parsimonious linear fingerprinting for time series," *Proc. VLDB Endow.*, vol. 3, pp. 385–396, Sept. 2010.

[5] I. Alon, M. Qi, and R. J. Sadowski, "Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods," *Journal of Retailing and Consumer Services*, vol. 8, no. 3, pp. 147 – 156, 2001.

[6] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497 – 505, 2005.

[7] S. Wang, J. Feng, and G. Liu, "Application of seasonal time series model in the precipitation forecast," *Mathematical and Computer Modelling*, pp. 677 – 683, 2013. Computer and Computing Technologies in Agriculture 2011 and Computer and Computing Technologies in Agriculture 2012.

[8] S. Abrishami, P. Kumar, and W. Nienaber, "Smart stores: A scalable foot traffic collection and prediction system," in *Advances in Data Mining. Applications and Theoretical Aspects: 17th Industrial Conference, ICDM 2017, New York, NY, USA, July 12-13, 2017, Proceedings*, pp. 107–121, 2017.

[9] M. Moustra, M. Avraamides, and C. Christodoulou, "Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15032 – 15039, 2011.

[10] E. S. Gardner, "Exponential smoothing: The state of the art," *Journal of Forecasting*, vol. 4, no. 1, pp. 1–28, 1985.

[11] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[12] R. Adhikari and R. K. Agrawal, "An introductory study on time series modeling and forecasting," *CoRR*, vol. abs/1302.6613, 2013.

[13] S. Lam, M. Vandenbosch, and M. Pearce, "Retail sales force scheduling based on store traffic forecasting," *Journal of Retailing*, vol. 74, no. 1, pp. 61 – 88, 1998.

[14] O. Perdikaki, S. Kesavan, and J. M. Swaminathan, "Effect of traffic on sales and conversion rates of retail stores," *Manufacturing & Service Operations Management*, vol. 14, no. 1, pp. 145–162, 2012.

[15] Özgür Kabak, F. Ülengin, E. Aktaş, Şule Önsel, and Y. I. Topcu, "Efficient shift scheduling in the retail sector through two-stage optimization," *European Journal of Operational Research*, vol. 184, no. 1, pp. 76 – 90, 2008.

[16] Cisco, "Location Analytics (CMX)," 2015.

[17] J. Anand, K. Young, R. Choudhary, and D. Howell, "Predicting shopper traffic at a retail store," Aug. 9 2011. US Patent 7,996,256.

[18] M. U. O. K. D. R. M. U. M. B. D. F. M. U. Welbourne, Neal D. (BEVERLY HILLS, "Analytic data capturing and processing system and method," February 2017.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct 2001.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, Nov. 1997.

[21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, 2014.

[22] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theor.*, vol. 13, pp. 21–27, Sept. 2006.

[23] S. J. Taylor and B. Letham, "Forecasting at scale," *PeerJ Preprints 5:e3190v1*, Aug. 2017.

[24] M. Christ, A. W. Kempa-Liehr, and M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications," *CoRR*, vol. abs/1610.07717, 2016.