PREDICTIVE ANALYTICS FOR APPLICATION MANAGEMENT SERVICES

Research paper

Stein, Nikolai, Julius-Maximilians-University, Würzburg, Germany, nikolai.stein@uni-wuerzburg.de

Flath, Christoph M., Julius-Maximilians-University, Würzburg, Germany, christoph.flath@uni-wuerzburg.de

Boehm, Carsten, Julius-Maximilians-University, Würzburg, Germany, carsten.boehm@stud-mail.uni-wuerzburg.de

Abstract

With digitization efforts running across all industries, IT consulting firms have enjoyed ever-increasing demand for their services. To cope with this demand surge, long-term hiring decisions, as well as short-term capacity planning and staffing, are of crucial importance for business viability. Predictive analytics methods offer enormous potentials to support planning and staffing of IT service desks to ensure both high capacity utilization and service levels. Yet, the current state-of-the-art for these planning activities still relies on traditional statistical forecasting methods. We collaborated with an IT service management firm to develop and evaluate an IT service demand forecasting using machine learning techniques. This approach allows us to improve planning accuracy by more than 30% compared with standard approaches.

Keywords: Predictive Analytics, Business intelligence, IT Incident Management, Machine Learning, Forecasting

1 Introduction

Greater business data availability and IT ubiquity have created a growing need for useful theories and tools for information extraction (Fayyad, Piatetsky-Shapiro, and Smyth, 1996). With constantly increasing computing power, new possibilities to gain insights from data have arisen (Gualtieri et al., 2015). Consequently, the buzzword “Big Data" has attained fame in almost all areas of business. Companies have started to see the opportunities for turning data into a commodity of high value for strategic and operative decision making and for providing competitive advantage (Waller and Fawcett, 2013). Following (Aggarwal, 2015), the process of data mining is further gaining importance.

One important application of data mining is predictive analytics, the forecasting of future events by using past data. Modern, effective and convenient tools have accelerated the popularity and use of predictive analytics throughout various business (Gualtieri et al., 2015). Time series forecasting constitutes a part of predictive analytics in which predictions are made for temporal data. Here, all recorded data is connected to a precise date and time and the forecasts are predominantly made by using internal structures of the data such as seasonality and trend (Aggarwal, 2015). While for time series forecasting tasks in business, statistical methods have been applied for many decades, machine learning (ML) models took root as a contestant for such tasks only in the last decade. Nowadays, ML methods play a big role in the analysis of large amounts of data, as they are able to learn with low or even without supervision and improve with the amount of data they are fed (Alpaydin, 2010). However, for the purpose of forecasting temporal data, the approach of training an ML model is different than applying it to time-independent data (Bontempi,

Taieb, and Le Borgne, 2012). In principle, all ML models are able to perform time series prediction tasks. However, depending on the number of selected lags, the number of features can get very large, which is why models with faster training times are deemed beneficial.

In this work, we analyze how predictive analytics can be applied in order to forecast the future demand for the application management and support division of a SAP consulting company. Given digitization initiatives across all major industries, this firm has seen significant growth over the recent years. In turn, long-term hiring decisions, as well as short-term capacity planning and staffing need to be able to keep up with this demand surge. On the operational level, detailed forecasting is essential for employee capacity and workload planning. On the strategic level, the business unit recognizes an opportunity for better gross profit estimation as well as budget planning. These tasks are of crucial importance for business viability and signify the importance of business intelligence applications (Popovič, Turk, and Jaklič, 2010).

Along these lines we set up a data science study addressing two guiding research questions:

RQ1 What is an appropriate machine learning setup (performance metric and model) to predict IT service management support demand?

RQ2 How does the machine learning setup perform on different aggregation levels and forecasting horizons compared to traditional forecasting models?

2 Related Work and Preliminaries

In the past decades the importance of IT service management is constantly growing. Consequently, this field is gaining an increased amount of interest in the information system community (Iden and Eikebrokk, 2013; Imgrund et al., 2017). Reviewing the relevant literature, a set of often discussed research questions can be identified. A variety of empirical studies focuses on the implementation strategies for IT service management and the success factors of these strategies (Cater-Steel, 2009; Cater-Steel and McBride, 2007; Cater-Steel, Tan, and Toleman, 2006; Hochstein, Tamm, and Brenner, 2005; Iden and Langeland, 2010; Marrone and Kolbe, 2011; McBride, 2009). Possible outcomes and benefits of these implementations are analyzed by Disterer (2012), Hochstein, Tamm, and Brenner (2005), and Marrone and Kolbe (2011). In contrast, only few studies related to IT incident management can be found in the Business Intelligence literature. Here, most research focuses on the labeling of service requests. Diao, Jamjoom, and Loewenstern (2009) and Maksai, Bogojeska, and Wiesmann (2014) develop classifiers to reduce the manual labeling effort. Goby et al. (2016) use a combination of topic modelling and predictive analytics to identify relevant topics and assign them to help desk tickets automatically.

However, there is a lack of research regarding the application of business intelligence in order to forecast the number and workload of incoming service requests. Hence, we widen our search horizon to other related research streams by following a three step approach: (1) exploring literature on related business topics; (2) analyzing literature on time series forecasting with machine learning and hybrid methods; (3) finding papers utilizing gradient boosting and in particular extreme gradient boosting (XGBoost). Analyzing the literature in detail, we can assigned the relevant research papers to different application areas (Table 1).

We find that business is one of the predominant sectors with a total of eight relevant papers. Here, call center arrival forecasting by means of statistical forecasting methods is analyzed four times (Aldor-Noiman, Feigin, and Mandelbaum, 2009; Shen and J. Z. Huang, 2008; Taylor, 2008). Foster (1977) show that time series forecasting is not a new topic by forecasting the quarterly seasonal accounting data of several firms using ARIMA. Cankurt (2016) predict future tourist demand by means of a random forest model. They show that ensembling methods outperform single models.

Energy is another popular sector, in particular electricity demand forecasting. For this task gradient boosting is applied (Kim et al., 2015; Mayrink and Hippert, 2016; Nassif, 2016; Taieb and Hyndman, 2014) as well as a statistical approach (Taylor, 2010). Also in the energy sector, but for event classification

Area of Application	Count	Examples
Business	8	call center arrivals, tourism demand, accounting data
Energy	8	electricity demand, status of water pumps, wind ramp events
Financial	7	stock & commodity prices, exchange rates
Science	7	breast cancer gene expression, sunspots, several competitions
Environment	2	metropolitan air pollution, waste generation
Engineering	2	compressor failures, fuel consumption

Table 1: Number of the literature found for each area of application with examples.

and not for time-series forecasting, another two approaches which utilize gradient boosting are found (Arymurthy and Darmatasia, 2016; Gupta et al., 2016).
Looking at financial time series forecasting, Krollner, Vanstone, and Finnie (2010) conduct a study on machine learning methods applied in this area. They find that artificial neural networks are the predominant technique, particularly for the prediction of stock market movements. The oldest relevant paper on financial time series forecasting found in the literature search was published in 1992. Here, a ANN is exploited for the prediction of a multivariate time series of monthly flour prices (Chakraborty et al., 1992). W. Huang, Nakamori, and Wang (2005) apply support vector machines for stock market prediction. The performance of support vector machines and artificial neural networks in terms of stock price and index predictions is compared by Pai and Lin (2005) and Tay and Cao (2001). The only work applying a boosting method in the financial sector, is the prediction of gold price volatility (Pierdzioch, Risse, and Rohloff, 2016).

3 Research Approach and Case Study Overview

Consulting firms collect large amounts of data including customer-related as well as problem-related information. Our industry partner, an IT consulting firm specialized in the areas of retail and logistics, aims to precisely forecast future demand for its support unit. The firm seeks to generate business value from the available data and the forecasts by improving the staff assignment decisions in the short-term and the business development estimations in the long term.
We want to address this prediction task as a data science study following the guidelines for applying big data analytics (Müller et al., 2016). Correspondingly, we structure our analysis along the proposed three phases:

Data collection For the extraction of the required data, several internal databases are harnessed. The resulting raw data set contains all employee bookings from September 2003 to December 2016 resulting in 358,184 entries and 23 variables. Each entry relates to a working hour booking of an employee on a specific project. For each booking, the date, time and duration along with a name and abbreviation is given. Further, the employee and the project plus the associated customer are specified. Additionally, the data consist of ticket information, distinguishing between support cases and priorities along with annotations, customer information and time based corrections.

Data analysis We develop a predictive model to forecast future demand for support requests. Starting with four statistical time series models, we increase the performance by the means of a hybrid model. To this end, we engineer new time-based and non-time-based features from the data set at hand. To leverage these features, we combine the statistical forecasts with an advanced XGBoost model (Chen and Guestrin, 2016).

Result interpretation The system is evaluated in Section 6. On the one hand, the machine learning approach significantly improves the short-term forecast allowing the company a more efficient employee scheduling. On the other hand, the long-term forecast can be improved allowing the company a more precise estimate of the business development.

4 Data Collection

In this section, we describe in detail the data collection process as well as the resulting data set. The resulting data set contains date-derived as well as non-date-derived variables describing the booking duration of support requests over the last 13 years. For the extraction of required data, several internal databases are harnessed.

4.1 Data Extraction and Cleaning

In the problem at hand, support requests can be distinguished into either tickets or task. Tickets are initiated by a customer while tasks are triggered by internal employees. Based on the urgency of a request, each ticket or task is assigned one out of five status codes.[1] Additionally, we can query information on the billing mode of a request. Here, a distinction between three types can be made. For billable requests, the working hours are charged on the invoice of a specific customer while they are only listed and not charged for reportable bookings. Internal requests are invoiced internally and can either originate from customers with included hours or bookings for employee education and training. Additional customer information can be used to augment the data set. Each of the 33 customers in the data set is categorized into one of twelve businesses.[2] Each of the businesses is assigned to one of three sectors, namely manufacturing, services and retail. We include a dummy business and sector for internal in-house bookings. The resulting raw data set consists of 358,184 entries and 23 variables describing all employee bookings from September 2003 to December 2016.
The analysis of data revealed 114,093 missing values in the target variable *booking.duration*. These observations can not be imputed and have to be removed as they resulted from inconsistent database structures. Subsequently, we identify several non-relevant variables and remove theme based on one of the following reasons: (1) redundancy, (2) inconsistency and (3) sparsity.

4.2 Exploratory Data Analysis

In order to understand the underlying structure of the problem we conduct an exploratory data analysis of the remaining data set (Tukey, 1977). To understand the following analysis it is important to note that the data for 2003 and 2016 is incomplete, since the recordings started in September 2003 and ended in November 2016.
In Figure 1, we observe a constant increase in the number of bookings as well as in the total duration of bookings (i.e., workload). In recent years, tickets account for roughly 50% of the bookings but only about 30% of the workload. Hence, we conclude that support cases triggered by internal employees require significantly more processing time than the tickets invoked by external customers. Zooming into the business and sector gives further inside into the data. Most of the workload can be assigned to the four businesses engineering, food, IT and OEM. While the former two are in the sector manufacturing and the later two in the sector services. Especially in the last two years, we observe an increase in the manufacturing sector and a decrease in the service sector. We also observe a heterogeneity in the processing behaviours of customers. The number of ticket hours is relatively low for OEMs and nonexistent for

[1] Statuses by urgency (in descending order): Incident, Service Request, Problem, Request for Change, Change
[2] Business to Business (B2B), Business to Consumer (B2C), Automotive, Chemical, Construction, Defence, Engineering, Food, Pharmaceutical, IT, Logistics and Original Equipment Manufacturer (OEM)

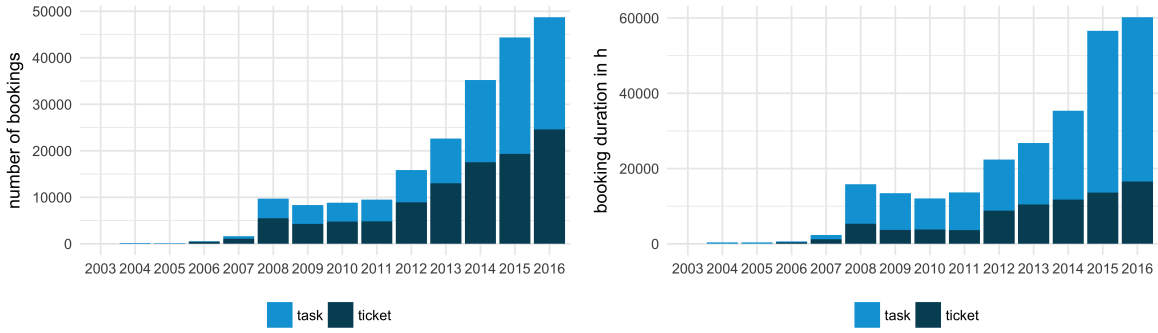

Figure 1: Number of bookings vs. total booking duration.

construction clients. Opposed to that, the logistics and IT sector have a much higher amount of working hours book on tickets.

In addition to the analysis of the differences between tasks and tickets, we also analyze the observable differences in the billing modes. Here, the manufacturing sector has the highest total booking duration. It is formed of 43% billable, 10% reportable and 47% internal bookings. Services constitute the second largest sector. Here, internal bookings account for 73% of the total workload. Remaining modes are billable at 25% and reportable at 2%. The dummy sector for in-house bookings constitutes the third largest sector. By definition it only reports internal bookings as the support requests can not be assigned to a client. The least amount of workload is booked in the retail sector. Here, the majority of requests (61%) is booked internally, followed by 30% reportable and only 9% billable bookings. Additionally, we also observe a heterogeneity regarding the billing modes inside the sectors on a business level. The distribution across the sectors and billing modes is summarized in Table 2.

Sector	Billable	Reportable	Internal	Total
Retail	9%	30%	61%	0.8%
In-house	0%	0%	100%	10.3%
Manufacturing	43%	10%	47%	56.2%
Services	25%	2%	73%	32.7%

Table 2: Billing modes across sectors.

Going further into detail, we observe a long tail distribution of working hours on the customer level. While the three biggest customers are responsible for over 50% of the workload, there is a multitude of customers with very sparse support requests.

Analyzing employee development, we find that the booking hours as well as the number of support requests are increasing over proportional to the number of workers. Hence, an increase in productivity due to process improvement can be concluded. Additionally, idle times can be reduced due to a better planning. Therefore, employees are able to not only handle more requests but also a higher workload. These findings are visualized in Figure 2.

Extracting the recurring seasonal patterns from the time series we find that the seasonal fluctuation is approximately 500 hours per month. Comparing this to the booking duration of 7000 at the end of 2016, the seasonality component only captures about 7% of the variance.

In order to train predictive models it is imperative to aggregate the data on a specific level Geurts (2002). Since our objective is to support the decision making on different management levels, forecasts for periods ranging from one day to one year have to be determined. Hence, we define three levels of aggregation: daily, weekly and monthly. We aggregate the target variable *booking.duration* for each level by summing

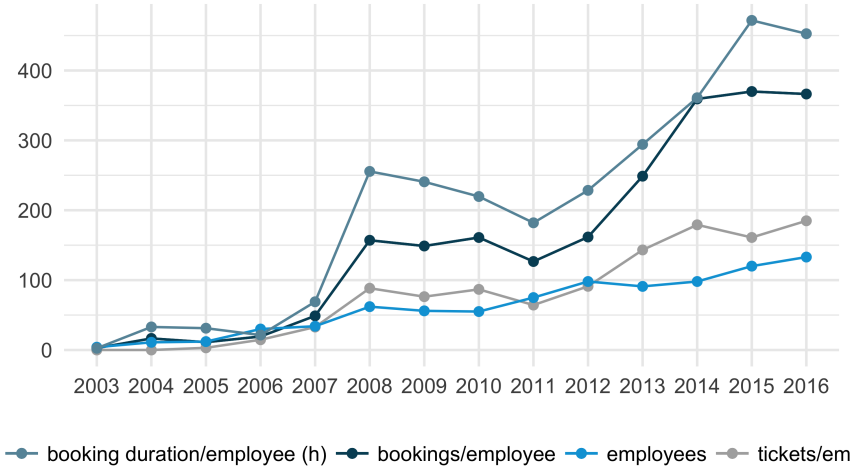

Figure 2: Development of employee number and average booking count / booking duration per employee.

up the single observations.

5 Model Setup

Prior to any modeling activities, a suitable evaluation metric has to be chosen. This metric has to account for the specific properties of the problem at hand. Following Davis et al. (2007), the metric selection is fundamental for the success –or failure– of every data science project. Regarding time series forecasting a variety of different metrics with different strengths and weaknesses is available. According to a classification broad forward by Hyndman and Koehler (2006), each measure is either a ”scale dependent measure," a ”percentage error measure" or a ”relative error measure."
In the forecasting task at hand, the scale of the workload varies over time. Hence, we chose to select a ”scale dependent measure". While the mean absolute percentage error (MAPE) is the most utilized quality measure, it comes with several weaknesses. According to Tofallis (2015), the MAPE has a tendency to prefer models underestimating the realized values. Hence, Armstrong (1978) suggest a symmetric version of MAPE called sMAPE. Since the original sMAPE has a range of $[-\infty, \infty]$ Hyndman and Koehler (2006) suggest using absolute values in the denominator, which is the version used in this paper. The metric is specified as

$$sMAPE = \frac{200}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

with y_i being the actual value, $\hat{y}_i$ the prediction and n the number of fitted points. Since for $y_i, \hat{y}_i = 0$ the function is undefined, the sMAPE is suggested to be only used for predicting values which are significantly higher than zero (Hyndman and Koehler, 2006). The range of sMAPE is $[0, 200]$. While this metric does not allow for an easy interpretation, it is well suited to compare the performance of different predictive models.

5.1 Modeling approach

Having established a suitable evaluation metric, a set of appropriate forecasting models has to be selected. To this end, we first select a set of statistical forecasting methods that later serve as a benchmark for the proposed machine learning approach. Following Bontempi, Taieb, and Le Borgne (2012), literature on time series analysis and forecasting is mostly based on these methods. Following Hyndman and

Athanasopoulos (2014), two simple approaches can be identified for statistical time series forecasting. The average or mean method is a simple approach mostly used as a benchmark for comparing it to more sophisticated models. It is also referred to as historic average as the prediction of future outcomes is equal to the average of the historical outcomes. As another simple benchmark approach, the naïve method sets all forecasts to the last observed value. Furthermore, the seasonal naïve approach can be utilized for highly seasonal data. Here, the forecasts are set to the values of the last value observed in the same season, which for instance could be the last observed value for the same month in the previous year. Autoregressive integrated moving average (ARIMA) is a more sophisticated approach of statistical time series forecasting. It is often applied in the relevant literature (Ho, Xie, and Goh, 2002; Khashei and Bijari, 2011; Zhang, 2003). However, ARIMA models can only be applied on stationary time-series without missing data Khashei and Bijari (2011). Being able to only capture linear patterns, the application of these models in real-world problems does not always reveal good forecasting results Zhang (2003).

To overcome this shortcoming, we suggest to apply advanced machine learning (ML) models. Such techniques are often referred to as black-box or data-driven models. They represent non-parametric and non-linear models that learn stochastic dependencies between historical and future data. While most ML methods can theoretically be applied for time series prediction tasks, only a small set of algorithms has gained popularity in this field. Friedman, Hastie, and Tibshirani (2001) and Hastie (2004) compare the performance of the five most popular methods (Artificial Neural Networks, Support Vector Machines, Multivariate Adaptive Regression Splines, K-Nearest Neighbours and Gradient Tree Boosting) regarding a set of characteristics (Natural handling of data of "mixed" type, handling of missing values, robustness to outliers in input space, insensitivity to monotone transformations of inputs, computational scalability (large N), ability to deal with irrelevant inputs, interpretability, predictive power). For each of the characteristics, the performance of the models is rated. While each of the methods show strengths and weaknesses in their performance, the gradient tree boosting approach exhibits the best performance overall. Additionally, the implementation XGBoost introduced by Chen and He (2015) is known for fast training times and high accuracy in predicting real world problems Hu et al. (2017) and Taieb and Hyndman (2014). For example on the data science competition platform Kaggle, out of all winning teams of the overall 29 challenges in 2015, in total 17 used XGBoost. Another advantage of this model other than the fast training times is the included automatic variable selection (Taieb and Hyndman, 2014). In particular, we decided to utilize XGBoost for the prediction task at hand for the following key reasons:

- Efficient training process
- High quality of predictions
- Robustness to overfitting

5.2 Traditional Forecasting as Benchmark

In order to define a competitive benchmark for the machine learning model, the four different statistical forecasting methods discussed in Section 5.1 are applied to the data set at hand. To this end, we use a rolling horizon evaluation with several forecasting horizons for all models on all three aggregation levels. The model quality is assessed for a one-day, three-day and seven-day ahead forecast for the daily data, a one-week, two-week and four-week forecast for the weekly data and a one-month, six-month and twelve-month forecast for the monthly data.

Table 3 summarizes the sMAPE metric for the different models and aggregation levels. Looking at the overall results, we clearly see the out-performance of the naïve and the ARIMA methods over the other forecasts. This finding even holds for all aggregation levels and forecasting horizons. While a strong improvement in forecast accuracy is generated by moving from daily to weekly aggregation, we see mixed results if the aggregation periods get longer. In general, we conduct that using a statistical forecasting method allows us to improve on the simple average method by up to 35% in terms of the sMAPE.

Aggregation	Forecast Horizon	Average Method	Naïve	Seasonal Naïve	ARIMA
daily	1 day	115.40	83.99	104.66	77.48
	3 days	116.14	103.01	102.09	81.52
	7 days	120.70	75.29	93.70	79.78
	mean	117.41	87.43	100.15	79.59
weekly	1 week	42.95	27.28	55.67	24.97
	2 weeks	43.45	29.84	54.69	27.17
	4 weeks	43.88	35.02	52.41	29.54
	mean	43.43	30.71	54.26	27.23
monthly	1 month	42.43	16.40	43.70	17.03
	6 months	48.04	21.41	42.94	25.05
	12 months	53.94	31.12	42.94	44.46
	mean	48.14	22.98	43.19	28.85
total	**mean**	**69.66**	**47.04**	**65.87**	**45.22**

Table 3: Rolling forecasting evaluation of statistical models.

5.3 Machine Learning Feature Engineering

So far our analysis only relied on past realizations of the target variable. To further increase the quality of the forecast, we now utilize a machine learning approach by building an XGBoost model. This model is able to generate valuable information based on multiple input variables. The process of developing these features is summarized under the term feature engineering. Following Domingos (2012), this process is critical to ensure the success of any data mining project. Going beyond basic raw features requires a significant portion of business and process understanding as well as creativity and luck. In a time-series setting, the features can either be time-based or non-time based. At first, we create the following set of time-based variables which serve as basis for the different aggregation levels:

day, **month**, **year**, **weeknumber** (numeric): These features a derived from the date variable.
weekday (categorical, *attributes*: Monday to Sunday.): Weekday of each specific date as an ordered factor.
isWeekend (categorical, *attributes*: *True*, *False*): Due to the observations of differing workloads, the distinction between weekends and weekdays is a promising feature (Mayrink and Hippert, 2016).
holiday (categorical, *attributes*: *holiday*, *bridgeDay*, *None*): Analyzing the raw data, we find a strong correlation between the booking duration and official German holidays. Additionally, bridge days which are days between a holiday and a weekend, appear to have an impact on the booking duration as well.
seasonal, **trend**(numeric): Seasonality and trend component of the daily booking duration time series.

For the weekly and monthly scales, the variables above have to be further aggregated:

holidays.in.week & **holidays.in.mth** (numeric): Total number of holidays in a week or month.
bridgedays.in.week & **bridgedays.in.mth** (numeric): Total number of bridge days in a week or month.
weekdays.in.mth, **weekenddays.in.mth** (numeric): Number of weekends and weekdays in a month. This is only meaningful for the monthly aggregation since a week has always the same number of weekend- and weekdays.
seasonal, **trend**(numeric): Seasonality and trend of the weekly or monthly booking duration time series, analogous to the daily aggregation.

In addition to the time-based variables, we create an additional set of explanatory features to increase the predictive power of the model. Hence, the following features are calculated for the relevant aggregation periods:

isTicket.true, **isTicket.false** (numeric): Count of bookings that are a ticket or a task. **total.bookings** (numeric): Total number of bookings for each day.
customer.bookings (numeric): Total number of bookings booked on customers. By implication, bookings on internal projects and tasks are excluded.
customer.durations (numeric): Total time booked across customers (w/o internal bookings).
billing.mode.* (numeric): Booking duration for each billing mode (*).
sector.* (numeric): Booking duration for each sector (*).
business.* (numeric): Booking duration for each business (*).

Since for the predictions of time series data solely date-derived features are known in advance, the forecasting can only be based on the behaviour of past data (Bontempi, Taieb, and Le Borgne, 2012). Hence, all variables that are not derived from the date, including the target variable *booking.duration.*, need to be shifted in time for the model to find stochastic dependencies between past and future data. Said shifting is predominantly referred to as lagging. In order to find the optimal lags, the autocorrelation for each aggregation level is analyzed.

As visualized in Figure 3, the daily model shows a high autocorrelation every seven days which endorses the weekly seasonal pattern. When looking at a higher lag horizon in the daily lags, the plot shows a mostly linear decline while keeping the overall seven-day pattern. Looking at the weekly booking time data, every 52 weeks or respectively one year, a slight increase in the autocorrelation can be observed. The graph shows an overall linear decline down to the 235-week lag. Opposed to the prior aggregation levels, in the monthly booking hour data no repeating peaking pattern can be observed. The graph shows an overall linear decline down to the 55-month lag. For all aggregation levels, the autocorrelation values start to get negative at about a 4.5 year horizon.

Based on this analysis, we choose the lags summarized in Table 4 for further modeling.

model	**lags**
daily	7, 14, 21, 28, 91, 182, 365, 730, 1,095, 1,460 days
weekly	8, 16, 52, 104, 156, 208 weeks
monthly	12, 24, 36, 48 months

Table 4: Specified lags for each model.

The use of high lags for all models contradicts the correlation analysis. However, since only the autocorrelation of the target variable and not the individual feature correlations were analyzed and XGBoost includes an automatic feature selection, we decide to use several higher lags.

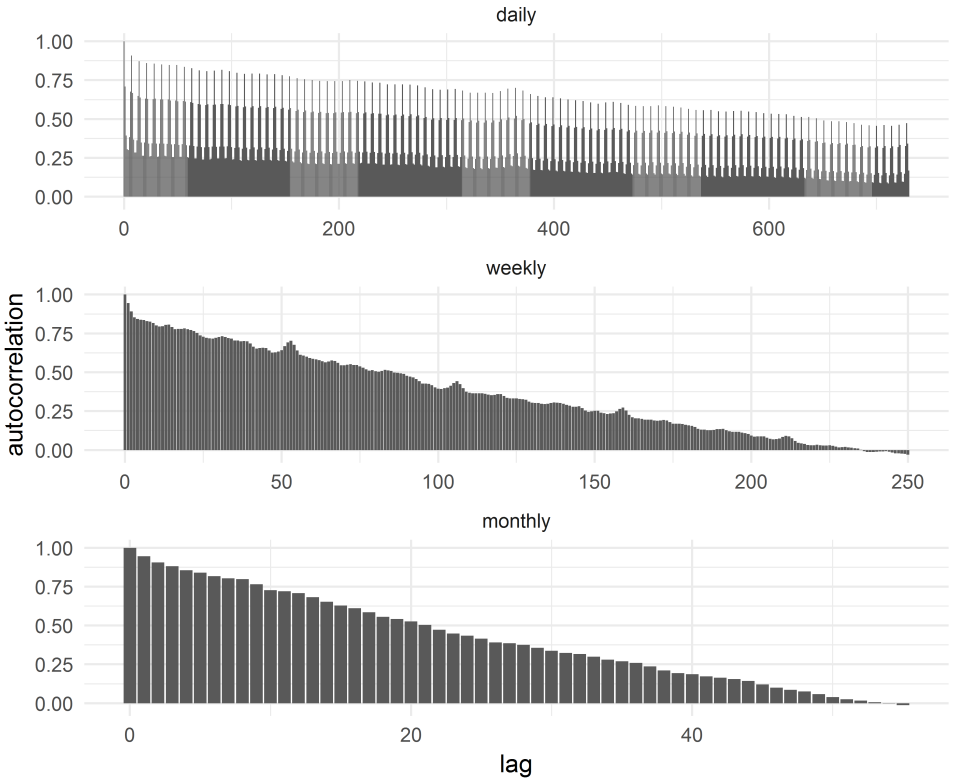

Figure 3: Autocorrelation of booking duration variable for each aggregation level.

Following Krollner, Vanstone, and Finnie (2010), the combination of several forecasting methods in order to achieve higher prediction quality is a promising approach. Hence, we decide to add the forecasts of ARIMA –the best performing statistical model– as an additional feature to the ML approach.

5.4 Model Refinement

Based on the features created above, we train nine separate ML models to tackle the prediction problem at hand. To leverage the full potential of the suggested approach, a single model is trained for each aggregation level (daily, weekly, monthly) and forecasting horizon. Additionally, the models are updated for each subsequent evaluation step following the rolling horizon approach. While this procedure provides us with well tuned models, it is computational expensive due to the size of the data set at hand.

The performance of the models can further be improved by selecting a good set of model parameters. In order to find a good set of parameters, we perform hyper-parameter grid search. To this end, over 400 possible parameter combinations are tested for each aggregation step. Due to computational limitations and in order to avoid overfitting, we decide to apply the same set of parameters for each forecasting horizon. Hence, the final hyper-parameter grid search has to be performed three times resulting in a total of 1,200 trained models. Table 5 summarizes and describes the tuneable parameters, the tested ranges and the selected values.

The quality for the nine ML models is evaluated using the same rolling horizon as in the statistical forecasting approach. The results are summarized in Figure 4. As expected, a decreasing accuracy can be observed with increasing forecast horizons. Additionally, the forecasting quality is better on higher aggregation levels.In terms of the sMAPE, the weekly and monthly models perform roughly twice as good compared to the daily aggregation level. This finding also holds in terms of forecast reliability, as the variance of the sMAPE is significantly reduced for the two higher aggregations.

Parameter	Description	Test Range	Daily	Weekly	Monthly
learning_rate	Shrinks the feature weights	0..1	0.5	0.01	0.005
gamma	Minimum loss reduction for partitioning tree leaf node	1..10	5	8	4
max_depth	Maximum depth of a tree	2..5	5	4	4
min_child_weight	Minimum number of instances in each node	1..10	7	6	5
subsample	Ratio of data to use for training	0..1	0.6	0.6	0.6
colsample_bytree	Ratio of columns to use for training	0..1	0.6	0.6	0.6

Table 5: Hyper-parameter optimization

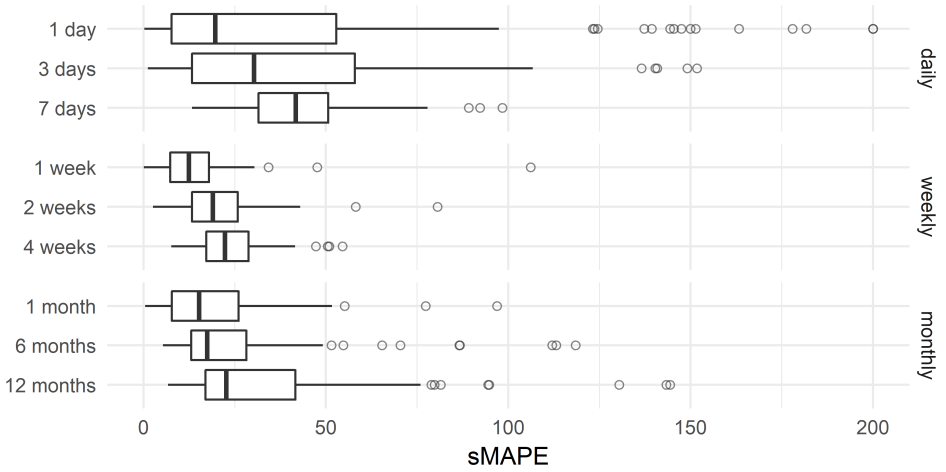

Figure 4: Comparison of sMAPE metrics throughout the models.

Table 6 compares the performance of the best statistical forecasting model ARIMA and the suggested ML approach. On the daily aggregation level, the biggest performance increased can be reported. Here, the average sMAPE is reduced by roughly 46%. With a reduction of roughly 27%, we also observe strong improvements on the weekly level. On the other hand, ARIMA and XGBoost perform at about the same quality on monthly aggregated data. Here, a significant increase in forecasting accuracy can only be observed for long forecasting horizons. The overall average performance in terms of the sMAPE shows a 34% better performance for the ML model.

6 Result Discussion and Interpretation

We developed and evaluated a system for support request forecasting in an IT consulting setting. We first determined several traditional statistical forecasts on several aggregation levels that serve as benchmarks for the later evaluation. Subsequently, we derived a meaningful feature set for a sophisticated ML forecast and performed hyper-parameter optimization.

In the age of big data, researchers as well as practitioners can no longer rely exclusively on standard statistical methods (e.g., ARIMA) to generate business insights from large data sets. Rather, the use of machine learning becomes inevitable as these approaches are better suited to handle thousands of variables

Aggregation	Forecast Horizon	ARIMA	XGBoost
daily	1 day	77.48	42.53
	3 days	81.51	42.33
	7 days	79.78	43.30
	mean	79.59	42.72
weekly	1 week	24.97	14.42
	2 weeks	27.17	20.99
	4 weeks	29.54	23.94
	mean	27.23	19.78
monthly	1 month	17.03	18.80
	6 months	25.05	25.32
	12 months	44.46	25.95
	mean	48.14	22.98
total	**mean**	**45.22**	**29.48**

Table 6: sMAPE comparison of XGBoost and ARIMA.

or work with unstructured data. Breiman (2001) and Shmueli et al. (2010) show that these approaches are of paramount importance in studies aiming at prediction instead of description. The main advantage of state-of-the-art ML algorithms is that they make less rigid statistical assumptions and are able to work with data-sets of very high dimensionality. Additionally, these methods are able to not only capture non-linear relationships but also pick up higher-order interaction effects between variables. On the downside, these black-box algorithms (e.g., gradient boosting machines) typically generate incomprehensible models and rules. Yet, the interpretability of the rules used by the algorithms is important if subsequent actions based on the predictions are to be taken by human decision-makers (Diakopoulos, 2014; Martens and Provost, 2014).

Answering the need for comprehensible prediction models as identified by Breuker et al. (2016), we analyze the importance of the features by means of the information gain. Figure 5 visualizes 20 most important features for all aggregation levels.

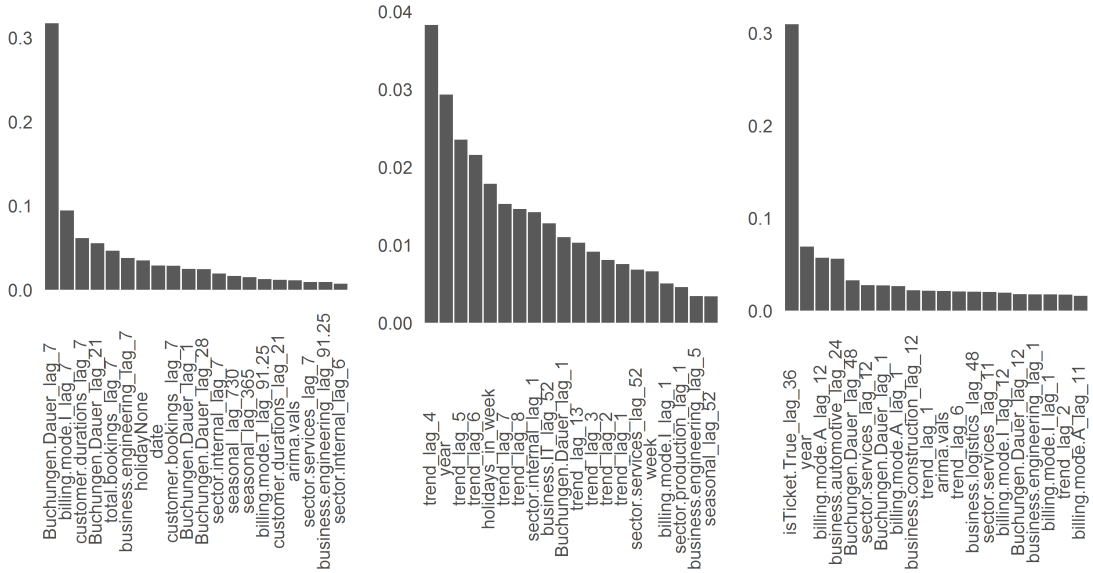

Figure 5: Feature importance for each aggregation level.

Looking at the feature importance of the daily model, the importance of the seven-day-lagged booking

duration (*Buchungen.Dauer*) provides clearly the most information gain. In the weekly model, more features are needed to leverage the full potential of the suggested approach. While the four-week-lagged *trend* is the most important feature, a total of nine *trend* features with different lags can be found in the top 20 features. The information gain in the monthly model shows a distribution similar to the daily aggregation with one feature holding the most importance. Here, the variable *isTicket.True* with a 36-month lag has the highest importance followed by the year variable.
For a better understanding of the model performance, the average employee-per-day error for the XGBoost model is calculated. For this, we assume an employee with an eight-hour work day on five weekdays (40 h/week). Calculating the mean absolute error per day, we find that the daily models average at an error of ~ 2.9 employees per day. The weekly models result in an average of ~ 3.4 employees per day and the monthly models in an average of ~ 2.6 employees per day. Considering the total of 137 employees in the business unit, the forecast should enable a high quality planning process.

7 Conclusion and Outlook

Using a large data-set from a IT consulting company we showcased the development, refinement and evaluation of a machine learning based forecasting system for incoming support tasks. Here, the objective is to improve short term as well as long term planning processes to improve capacity utilization and service levels.
After the extraction of raw booking data from several databases, we perform an extensive exploratory data analysis to identify patterns informing the subsequent modeling phase. During this analysis we clean the data set and remove redundant and sparse variables. Subsequently, an appropriate metric as well as appropriate statistical forecasting methods and a machine learning approach are chosen. Leveraging a powerful feature set, we show the out-performance of the suggested machine learning approach in comparison to traditional forecasting methods. On average, the machine learning model is able to increase the forecast accuracy by 34% depending on the forecast horizon and the aggregation level. Especially for operational short-term planning, the machine learning approach is far superior.
In future work, we intend to extend our case study in the following directions. First, the accuracy as well as the robustness of the predictions could be further improved by leveraging additional internal and external data sources (e.g., financial data or press releases) and creating more explanatory features. Second, manual model adjustments could be allowed in order to incorporate human knowledge into the model. To this end, the feature set could be enhanced by adding a variable with expert estimations. Third, the performance of the XGBoost model can be compared to other machine learning algorithms such as geometric semantic genetic programming (Castelli et al., 2016).

References

Aggarwal, Charu C. (2015). *Data Mining*. Cham: Springer International Publishing. ISBN: 978-3-319-14141-1. DOI: 10.1007/978-3-319-14142-8.

Aldor-Noiman, Sivan, Paul D Feigin, and Avishai Mandelbaum (2009). "Workload forecasting for a call center: Methodology and a case study." *The Annals of Applied Statistics*, 1403–1447.

Alpaydin, Ethem (2010). *Introduction to Machine Learning*. 2nd. The MIT Press. ISBN: 9780262012430.

Armstrong, J Scott (1978). *Long-range Forecasting: From Crystal Ball to Computer*. A Wiley interscience publication. John Wiley & Sons Canada, Limited. ISBN: 9780471030027. URL: https://books.google.de/books?id=7DAcAAAAIAAJ.

Arymurthy, Aniati Murni and Darmatasia (2016). "Predicting the status of water pumps using data mining approach." In: *Big Data and Information Security (IWBIS), International Workshop on*. IEEE, pp. 57–64.

Bontempi, Gianluca, Souhaib Ben Taieb, and Yann-Aël Le Borgne (2012). "Machine Learning Strategies for Time Series Forecasting." In: *eBISS*. Springer, pp. 62–77.

Breiman, Leo (2001). "Statistical Modeling: The Two Cultures." *Statistical Science* 16 (3), 199–215.

Breuker, Dominic, Martin Matzner, Patrick Delfmann, and Jörg Becker (2016). "Comprehensible Predictive Models for Business Processes." *MIS Quarterly* 40 (4).

Cankurt, Selcuk (2016). "Tourism demand forecasting using ensembles of regression trees." In: *Intelligent Systems (IS), 2016 IEEE 8th International Conference on*. IEEE, pp. 702–708.

Castelli, Mauro, Leonardo Vanneschi, Luca Manzoni, and Aleš Popovič (2016). "Semantic genetic programming for fast and accurate data knowledge discovery." *Swarm and Evolutionary Computation* 26, 1–7.

Cater-Steel, Aileen (2009). "IT service departments struggle to adopt a service-oriented philosophy." *International Journal of Information Systems in the Service Sector (IJISSS)* 1 (2), 69–77.

Cater-Steel, Aileen and Neil McBride (2007). "IT service management improvement-actor network perspective." In: *Proceedings of the 15th European Conference on Information Systems (ECIS 2007)*. University of St Gallen, pp. 1202–1213.

Cater-Steel, Aileen, Wui-Gee Tan, and Mark Toleman (2006). "Challenge of adopting multiple process improvement frameworks." In: *Proceedings of 14th European conference on information systems (ECIS 2006)*. European Conference on Information Systems, pp. 1375–1386.

Chakraborty, Kanad, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka (1992). "Forecasting the behavior of multivariate time series using neural networks." *Neural networks* 5 (6), 961–970.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System." In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: http://doi.acm.org/10.1145/2939672.2939785.

Chen, Tianqi and Tong He (2015). "Higgs boson discovery with boosted trees." In: *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, pp. 69–80.

Davis, Jason V, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon (2007). "Information-Theoretic Metric Learning." In: *Proceedings of the 24th international conference on Machine learning*. ACM, pp. 209–216.

Diakopoulos, Nicholas (2014). "Algorithmic Accountability Reporting: On the Investigation of Black Boxes." *Tow Center for Digital Journalism*.

Diao, Yixin, Hani Jamjoom, and David Loewenstern (2009). "Rule-based problem classification in it service management." In: *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*. IEEE, pp. 221–228.

Disterer, Georg (2012). "Why Firms Seek ISO 20000 Certification-a Study of ISO 20000 Adoption." In: *ECIS*, p. 31.

Domingos, Pedro (2012). "A Few Useful Things to Know about Machine Learning." *Communications of the ACM* 55 (10), 78–87.

Fayyad, Usama, Gregory Piatetsky-Shapiro, and Padhraic Smyth (1996). "From data mining to knowledge discovery in databases." *AI magazine* 17 (3), 37.

Foster, George (1977). "Quarterly accounting data: Time-series properties and predictive-ability results." *Accounting Review*, 1–21.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin.

Geurts, Pierre (2002). "Contributions to decision tree induction: bias/variance tradeoff and time series classification." PhD thesis. University of Liège Belgium.

Goby, Niklas, Tobias Brandt, Stefan Feuerriegel, and Dirk Neumann (2016). "Business Intelligence for Business Processes: the Case of IT Incident Management." In: *ECIS*, ResearchPaper151.

Gualtieri, Mike, April Rowan Curran, Key TaKeaways, and Modern Tools Bring Predictive Power To (2015). "The Forrester Wave™: Big Data Predictive Analytics Solutions, Q2 2015."

Gupta, Saurav, Nitin Anand Shrivastava, Abbas Khosravi, and Bijaya Ketan Panigrahi (2016). "Wind ramp event prediction with parallelized gradient boosted regression trees." In: *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, pp. 5296–5301.

Hastie, Trevor (2004). *Stanford University presentation on boosting*. URL: `http://jessica2.msri.org/attachments/10778/10778-boost.pdf` (visited on 10/29/2017).

Ho, SL, M Xie, and TN Goh (2002). "A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction." *Computers & Industrial Engineering* 42 (2), 371–375.

Hochstein, Axel, Gerrit Tamm, and Walter Brenner (2005). "Service oriented IT management: benefit, cost and success factors." *ECIS 2005 Proceedings*, 98.

Hu, Ke, Ashfaqur Rahman, Hari Bhrugubanda, and Vijay Sivaraman (2017). "HazeEst: Machine Learning Based Metropolitan Air Pollution Estimation From Fixed and Mobile Sensors." *IEEE Sensors Journal* 17 (11), 3517–3525.

Huang, Wei, Yoshiteru Nakamori, and Shou-Yang Wang (2005). "Forecasting stock market movement direction with support vector machine." *Computers & Operations Research* 32 (10), 2513–2522.

Hyndman, Rob J and George Athanasopoulos (2014). *Forecasting: principles and practice*. OTexts.

Hyndman, Rob J and Anne B Koehler (2006). "Another look at measures of forecast accuracy." *International journal of forecasting* 22 (4), 679–688.

Iden, Jon and Tom Roar Eikebrokk (2013). "Implementing IT Service Management: A systematic literature review." *International Journal of Information Management* 33 (3), 512–523.

Iden, Jon and Lars Langeland (2010). "Setting the stage for a successful ITIL adoption: A Delphi study of IT experts in the Norwegian armed forces." *Information systems management* 27 (2), 103–112.

Imgrund, Florian, Marcus Fischer, Christian Janiesch, and Axel Winkelmann (2017). "Managing the long tail of business processes." In: *Proceedings of the 25th European Conference on Information Systems (ECIS 2017)*.

Khashei, Mehdi and Mehdi Bijari (2011). "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting." *Applied Soft Computing* 11 (2), 2664–2675.

Kim, Taehoon, Dongeun Lee, Jaesik Choi, Anna Spurlock, Alex Sim, Annika Todd, and Kesheng Wu (2015). "Extracting Baseline Electricity Usage with Gradient Tree Boosting."

Krollner, Bjoern, Bruce Vanstone, and Gavin Finnie (2010). "Financial time series forecasting with machine learning techniques: A survey." In: *European symposium on artificial neural networks: Computational and machine learning*.

Maksai, Andrii, Jasmina Bogojeska, and Dorothea Wiesmann (2014). "Hierarchical incident ticket classification with minimal supervision." In: *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, pp. 923–928.

Marrone, Mauricio and Lutz M Kolbe (2011). "Impact of IT service management frameworks on the IT organization." *Business & Information Systems Engineering* 3 (1), 5–18.

Martens, David and Foster Provost (2014). "Explaining Data-Driven Document Classifications." *MIS Quarterly* 38 (1), 73–99.

Mayrink, Victor and Henrique S Hippert (2016). "A hybrid method using Exponential Smoothing and Gradient Boosting for electrical short-term load forecasting." In: *Computational Intelligence (LA-CCI), 2016 IEEE Latin American Conference on*. IEEE, pp. 1–6.

McBride, Neil (2009). "Exploring service issues within the IT organisation: Four mini-case studies." *International journal of information management* 29 (3), 237–243.

Müller, Oliver, Iris Junglas, Jan vom Brocke, and Stefan Debortoli (2016). "Utilizing Big Data Analytics for Information Systems Research: Challenges, Promises and Guidelines." *European Journal of Information Systems* 25 (4).

Nassif, Ali Bou (2016). "Short term power demand prediction using stochastic gradient boosting." In: *Electronic Devices, Systems and Applications (ICEDSA), 2016 5th International Conference on*. IEEE, pp. 1–4.

Pai, Ping-Feng and Chih-Sheng Lin (2005). "A hybrid ARIMA and support vector machines model in stock price forecasting." *Omega* 33 (6), 497–505.

Pierdzioch, Christian, Marian Risse, and Sebastian Rohloff (2016). "A boosting approach to forecasting the volatility of gold-price fluctuations under flexible loss." *Resources Policy* 47, 95–107.

Popovič, Aleš, Tomaž Turk, and Jurij Jaklič (2010). "Conceptual model of business value of business intelligence systems." *Management: Journal of Contemporary Management Issues* 15 (1), 5–30.

Shen, Haipeng and Jianhua Z Huang (2008). "Interday forecasting and intraday updating of call center arrivals." *Manufacturing & Service Operations Management* 10 (3), 391–410.

Shmueli, Galit et al. (2010). "To Explain or to Predict?" *Statistical Science* 25 (3), 289–310.

Taieb, Souhaib Ben and Rob J Hyndman (2014). "A gradient boosting approach to the Kaggle load forecasting competition." *International journal of forecasting* 30 (2), 382–394.

Tay, Francis EH and Lijuan Cao (2001). "Application of support vector machines in financial time series forecasting." *Omega* 29 (4), 309–317.

Taylor, James W (2008). "A comparison of univariate time series methods for forecasting intraday arrivals at a call center." *Management Science* 54 (2), 253–265.

— (2010). "Triple seasonal methods for short-term electricity demand forecasting." *European Journal of Operational Research* 204 (1), 139–152.

Tofallis, Chris (2015). "A better measure of relative prediction accuracy for model selection and model estimation." *Journal of the Operational Research Society* 66 (8), 1352–1362.

Tukey, John W (1977). "Exploratory data analysis."

Waller, Matthew A and Stanley E Fawcett (2013). "Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management." *Journal of Business Logistics* 34 (2), 77–84.

Zhang, G Peter (2003). "Time series forecasting using a hybrid ARIMA and neural network model." *Neurocomputing* 50, 159–175.