Learning how people respond to changes in energy prices

Ben Lim
bclimate@stanford.edu

Jonathan Roth
jmroth@stanford.edu

Andrew Sonta
asonta@stanford.edu

***Abstract*—We investigate the problem of predicting day-ahead energy consumption for three groups of households: those that receive notification that energy prices will increase the following day, those that do not, and a combined group. We construct features of past energy consumption values, type of notification, and weather data, and we use Ordinary Least Squares, Random Forests, Support Vector Machines, and Autoregressive Integrated Moving Average models to predict the day-ahead consumption. We find the Random Forest model to be the most successful in predicting the aggregated consumption of all households in the group. We find that, on average, it is hardest to predict day-ahead energy consumption for houses that do receive notifications, easier for the combined group, and easiest for houses that do not receive notifications.**

I. INTRODUCTION

Utilities are beginning to adopt time-of-use rate plans where electricity prices change throughout the day depending on both supply and demand. Rather than solely focus on meeting demand with added supply, utilities are attempting to curb demand by informing customers of periods of high electricity prices through event notifications to encourage consumers to use less energy. Although modest reductions in energy have been observed with these notifications, it is uncertain how they affect next day energy predictions.

This project aims to predict day-ahead energy consumption for the 3-hr pricing event period (*i.e.,* 4 pm – 7 pm) for three different groups of households: those who receive notification of pricing changes, those who do not receive notification, and the combination of both groups. The inputs to our algorithms are: energy consumption in that same 3-hr period from prior days, information about the type of notification received by households, and information about current weather conditions. We use Ordinary Linear Regression (OLS), Random Forests, Support Vector Machines (SVMs), and Autoregressive Integrated Moving Average (ARIMA) models to learn from the inputs and predict day-ahead energy consumption for the 4 pm – 7 pm period when energy prices change. The goal is to maximize accuracy for the aggregate consumption of each population—since utilities care about the total demand on the grid—as well as to understand if the presence of event notifications results in less accurate models.

II. RELATED WORK

Forecasting next day energy consumption is important to provide utilities with information for decision-making regarding unit commitment, economic dispatch, and power systems operation [1]. A growing interest is developing in predicting individual household consumption due to its connection with management of distributed energy systems and for use in forecasting bus level loads. Current research is exploring the impacts that various calendar effects and temporal scales have on forecasting accuracy to aid in future model development [1,3]. This research is showing that daily and weekly seasonality, combined with weather data, are especially important when limited historical data is present [1]. Hourly intervals have also shown to result in models with the most predictive power, as they are less noisy than finer temporal scales yet contain more information than daily intervals [2]. Furthermore, one year of historical load data has been shown to be sufficient to develop adequate forecasting models, eliminating the need for more historical data [1].

A variety of models have been examined to forecast load consumption. Regression based models are the most commonly used but new work has begun to explore more complex methods like neural nets and support vector machines [2,4]. These methods have shown to consistently outperform the regression based models, but more work needs to be done to explore model performance between different levels of building energy aggregation [3,5]. Random forests have also been utilized in this space and have shown low errors when predicting consumption for large numbers of buildings [6]. However, much of this research takes into account building characteristics to improve the model accuracy, but these works may be limited in their real-world extension to utilities, as these entities often do not possess that kind of data when forecasting demand. There is also a lack of research in understanding the impact on forecasting accuracy during various demand-response programs, like price event notifications. This project aims to provide some preliminary insight into forecasting accuracy for these event notifications, building off insights from this past work.

III. DATASET

The dataset used for our study comes from Pecan Street, which is a research and development organization in Texas [7]. The dataset includes hourly energy consumption for 1,436 residential homes in Austin, TX, daily weather values, and information on which subset of buildings receive notification that prices will increase for the 4 pm – 7pm period in the day following the event. For the remainder of this report, we refer to these days on which prices change for some households as "event days." Fig. 1 shows an example of the raw data from Pecan Street, in the form of one home's energy consumption values, between the hours of 4 pm and 7 pm, over the course of two full years.

To predict energy consumption for houses on event days, we constructed the following full set of features (Table 1):

Table 1: Feature descriptions

Feature name(s)	***Feature(s) description***
lag_0, lag_1, lag_6, lag_7, lag_13, lag_14	Consumption for 4:00 - 7:00 PM for X number of days before the notification in kWh (X indicated in variable name)
event	0/1 indicator for if the household received notification
event_type	1...5 categorical variable for type of notification received
pv_panel	0/1 indicator for if the household had solar panels on their roof
tz_offset	Timezone adjustment
temperature, temp2, dew_point, humidity, visibility, apparent_temperature, pressure, wind_speed, cloud_cover, wind_bearing, precip_intensity, precip_probability, max_temp	Various weather attribute forecast simulations for the day of the event
CDD, CDD2	Cooling degree days (measure of amount of energy needed to cool house) / CDD squared
HDD, HDD2	Heating degree days (measure of amount of energy needed to heat house) / HDD squared

Because we are trying to predict energy consumption on event days only, we preprocessed our data to only include 4 pm – 7 pm energy consumption on days that we were trying to predict, or days that were used as features as indicated in Table 1 above.

The events typically occur in summer months when utilities are expecting an abnormally hot day. In the dataset, there are 27 total instances of pricing events, which we split into training, development (dev), and testing sets for our analysis. Due to the time series nature of the problem, the splits into these three buckets were chronological. The train set includes 21 event days (78%) from June 20, 2013 to August 15, 2013. The dev set includes 3 event days (11%) from August 21, 2013 to September 3, 2013. And the training set includes 3 event days (11%) from September 9, 2013 to September 15, 2013.

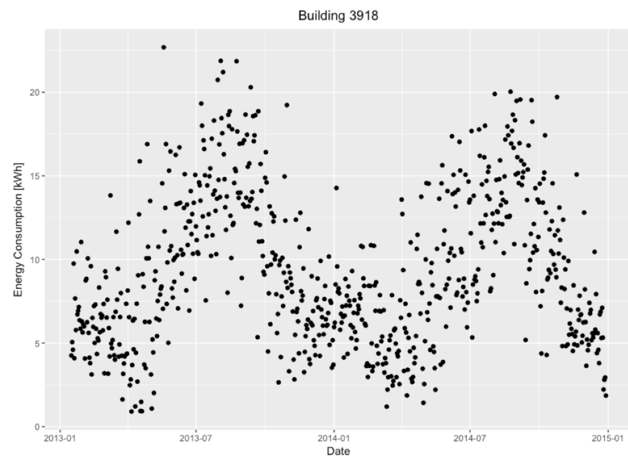

Fig. 1. Two-year energy consumption pattern for a single building in the Pecan Street dataset.

IV. METHODS & EXPERIMENTS

To make our predictions for day-ahead energy consumption, we tested four categories of methods: Ordinary Least Squares (OLS), Random Forests, Support Vector Machines (SVMs), and Autoregressive Integrated Moving Average (ARIMA) models. Each model category is discussed in detail below, with particular attention to the variations we used for each type of method. Each model variation was developed on the training set for future comparison on the dev set. Model development and testing, for all different models, was done in the R programming language using the following packages: e1071, glmnet, randomForest, and forecast.

A. Ordinary Least Squares (OLS)

$$\beta = argmin\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \qquad (1)$$

Three different techniques were used to select variables to include when fitting the model to the dataset. First, regularization and feature selection was performed using the lasso which applies a penalty in the cost function for non-zero coefficients using the L1 norm (as opposed to the L2 norm in ridge regression). Two other methods were examined using stepwise variable selection through bidirectional elimination using both the AIC and BIC criteria to select final variables. The BIC process is stricter and therefore results in a model with fewer selected variables than when using AIC.

The lambda parameter for regularization was selected using the one-standard error rule, as shown in Fig. 2. This finds the lambda associated with the minimum cross-validation error and then finds the max lambda that falls within one-standard deviation of this. This process is used to eliminate variables that

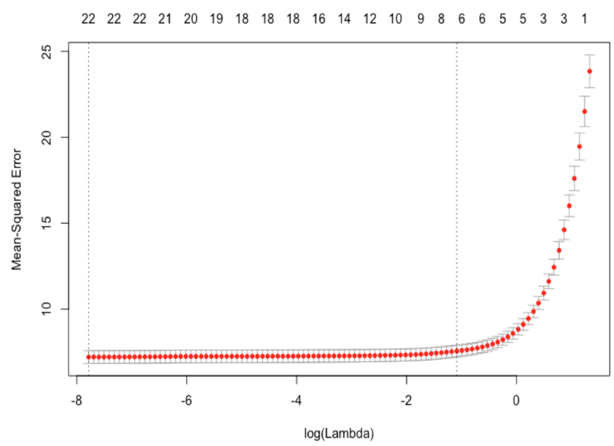

Fig. 2. OLS regularization search

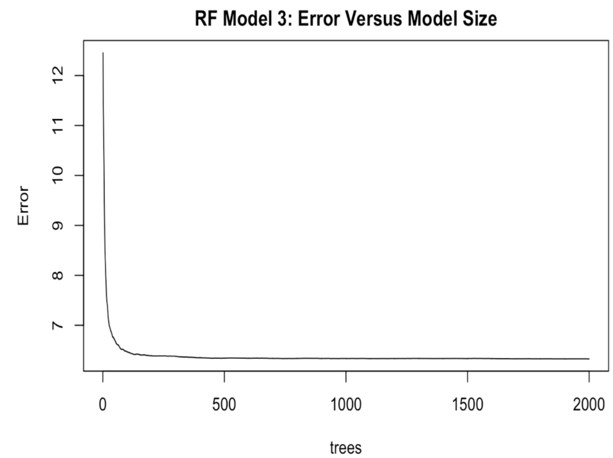

Fig. 3. Random Forests – Error (estimated on train set) vs. # of Trees

have a negligible effect on training performance, to help prevent overfitting.

B. Random Forests

$$\widehat{y_i} = \frac{1}{T}\sum_{t=1}^{T}\widehat{y_{it}}\,, e_i = (y_i - \widehat{y_i})^2 = L(y_i - \widehat{y_i}) \tag{2}$$

The Random Forests model is an ensemble method that utilizes many classification and regression trees to determine its final prediction. The training algorithm selects random samples with replacement (bagging) and fits trees to each of these samples and produces a final output by averaging the predictions from each of the trees. This methodology decreases the variance of the model and helps prevent overfitting, which is a common problem when only one tree is constructed. Each tree is sensitive to noise in the training set but the average of the many uncorrelated trees (through bagging) is not susceptible to this problem. To further prevent correlated and overly noisy trees, the user can select the number of variables tested at each split and the total number of trees to build (hyper parameters).

In constructing our model, we tried nearly a hundred permutations of the same model, where we varied which variables to choose from, the maximum number of variables tested at each split, number of trees (ranging from 100 to 2000 (Fig. 3)), and manual vs automated variable selection. We ultimately settled on three models that we felt were representative of the rest of the permutations, and the final model used 2000 trees, tested up to 12 variables at each split and employed correlation based variable selection.

The upper limit for the number of trees was capped at 2000 by our computational capacity.

C. Support Vector Machines (SVM)

Support Vector Machines are a class of algorithms suitable for either classification or regression problems. SVM regression is formulated through the optimization shown in (3) [8]

$$maximize \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i x_j) \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) \end{cases}$$

$$subject\ to \begin{cases} \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \tag{3}$$

As a result of this problem formulation, SVM regression requires tuning certain parameters, in particular the cost parameter C and the epsilon parameter. We performed a grid search—as shown in Fig. 4—on a range of both parameters to choose the best ones. The performance was measured through 10-fold CV on the training set. The best parameters are found to be: C = 4, epsilon = 0.1.

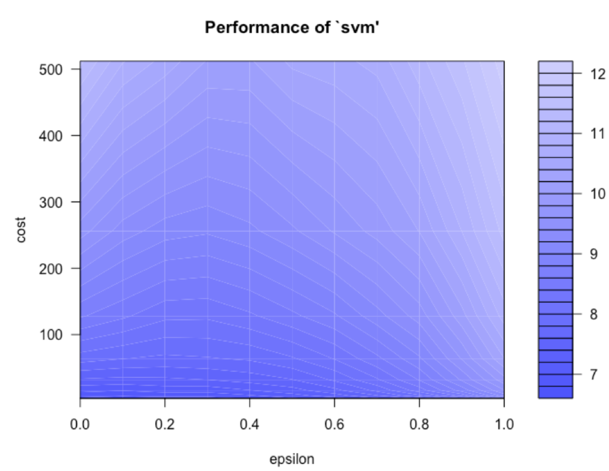

Fig. 2. Grid search for SVM cost and epsilon parameters

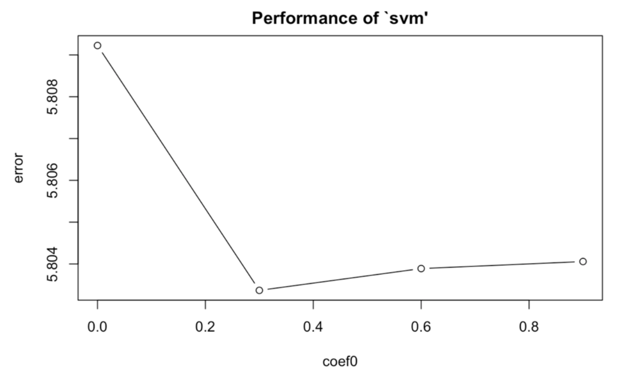

Fig. 4. coef0 tuning for SVM with polynomical kernel

The construction of SVMs allow for nonlinear transformations of the data that be computed very efficiently through the use of kernels, indicated by the [k(x,x)] function in the optimization. Three kernels were tested and tuned if necessary, as discussed in the following subsections.

1) Polynomial kernel

The polynomial kernel was implemented using the following formulation:

$$k(u,v) = (\gamma u'v + coef0)^{degree} \tag{5}$$

Tuning of the gamma, degree, and coef0 parameters was done by testing ranges of each parameter. An example of tuning for the gamma parameter is shown in Fig. 5.

2) Radial kernel

The radial kernel was implemented using the following formulation:

$$k(u,v) = \exp\left(-\gamma|u - v|^2\right) \tag{6}$$

Here, the gamma parameter was tuned in a similar process.

3) Linear kernel

The linear kernel was implemented using the following formulation:

$$k(u,v) = u'v \tag{7}$$

No hyperparameters needed tuning for the linear kernel.

D. Autoregressive Integrated Moving Average (ARIMA)

$$\phi(B)(1-B)^d x_t = \theta(B)w_t \tag{8}$$

The ARIMA model is popular in time series analysis and forecasting due to its ability to quickly adapt to sudden changes in the trend. It regresses on its own lagged values, adjusts for drift in the series, and accounts for the the regression error terms that are linear combinations of its past values. This is especially valuable when the data correlates with its past values. Three different variations were examined, all using an automated parameter selection process to determine the number of lags, the degree of differencing, and the order of the moving average based on AIC. The first model aimed to individually forecast household consumption based solely off its historical data. The other two models were constructed on the resulting time series for the total energy consumed each day for all the households in the dataset. This alternative dataset helped eliminate a lot of the variance that exists for individual households but does not allow for individual forecasts. One model was constructed using only the time series data from this dataset while the other included exogeneous weather variables (also known as an ARIMAX model). Since these models are specifically designed to forecast time series data, the dev error is calculated by first forecasting the consumption for one date, and then including that day of data before forecasting the subsequent day of interest (*i.e.*, the data used to predict future consumption grows with time).

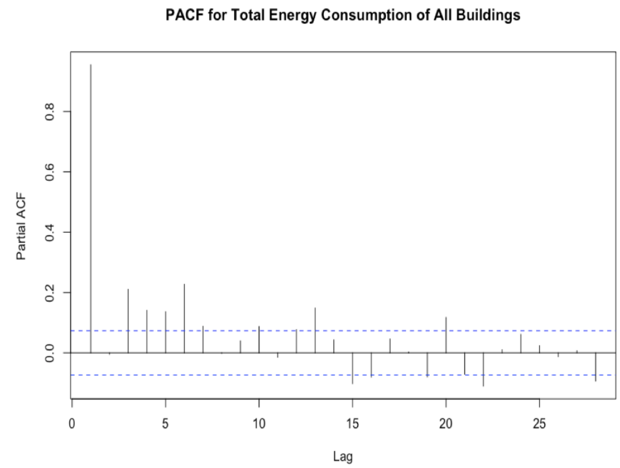

Fig. 3. Partial autocorrelation function (PACF)

The partial autocorrelation function (PACF) was found for the aggregate building energy consumption to help determine which lags would be included in the autoregressive part of the model, as shown in Fig. 6. As we can see, lags of 1, 2, and 7 days correlate with day-ahead energy consumption values. This graph was also useful for validating the constructed dataset that was used for the OLS, random forests, and SVM models since we included lags that were shown to positively correlate in this PACF graph.

V. Results & Discussion

The results for our different models are shown in Table 2. To measure the performance of each model, we used three different metrics, each metric reported for each model and for each of the three groups:

1. Mean Squared Error on the dev set. The errors are calculated for each of the houses in the group analyzed.
2. Mean Squared Error on the training set, calculated for each of the houses. This is calculated in order to get a sense of whether the model is overfitting on the training set.

3. Aggregated percent error (9). This is the percent error calculated by summing all the predictions for the houses in the group, and comparing that value to the sum of all the actual energy consumed by the houses. This is included because grid operators care more about the total demand on the grid, rather than energy consumption patterns for individual houses.

$$\frac{predicted - actual}{actual} \tag{9}$$

Table 2: Summary of estimated model results

Model Type	Control Group	Event Group	Combined Group
OLS - Regularization	5.26 (5.80) 3.39%	6.42 (7.48) 3.41%	5.59 (6.41) 3.99%
OLS - Stepwise (AIC)	8.63 (5.40) 9.27%	7.35 (7.10) -1.31%	8.51 (6.22) 8.08%
OLS - Stepwise (BIC)	5.25 (5.43) 3.17%	6.77 (7.13) 4.40%	5.58 (6.25) 3.57%
Random Forest (Model 1)	5.38 (1.00) 5.59%	6.39 (1.28) 3.33%	5.69 (1.13) 4.34%
Random Forest (Model 2)	5.35 (1.16) 4.49%	6.34 (1.45) 3.30%	5.64 (1.26) 4.09%
Random Forest (Model 3)	5.93 (1.03) 4.03%	7.33 (1.30) **-0.99%**	6.50 (1.13) 2.83%
SVM (Radial Kernel)	5.35 (5.18) **1.00%**	6.36 (6.85) 3.22%	5.56 (6.08) 2.02%
SVM (Polynomial Kernel)	5.25 (5.48) 1.39%	6.50 (7.05) 3.11%	5.65 (6.26) **1.72%**
SVM (Linear Kernel)	5.23 (5.59) 2.16%	6.27 (7.28) 4.56%	5.61 (6.38) 2.77%
ARIMA (using single building data)	21.10 - -9.98%	12.52 - -1.58%	17.07 - -13.98%
ARIMA (All Buildings summed)	- - 2.76%	- - 3.94%	- - 3.01%
ARIMA (All Buildings summed w/ weather data)	- - 3.36%	- - 3.58%	- - 3.39%

In each box: Dev Set Mean Squared Error
Train Set Mean Squared Error
Percent Error (aggregate of full building portfolio)

Three models are highlighted in Table 2 based on their performance: the third random forest model (RF3), the SVM with a radial kernel, and the SVM with a polynomial kernel. We chose these models using the third metric—the aggregated percent error—since we wanted to take the perspective of the utility in our analysis. Utilities are concerned with providing the correct amount of electricity to the grid, which can be viewed as energy demand aggregated across all the houses on the grid, hence we wanted to consider our models performance on the aggregate.

RF3 had the overall lowest performance on the "Event" group, and comparable performance to the SVM-polynomial model on the "Combined" group. For these reasons, we chose this model as the best. The parameters for this model are the following: 2000 Trees, 12 Variables tested at each split, and an optimized variable selection process based on correlation scores. We evaluated the final performance of this model by computing its aggregate percent error on each group in the test set. Table 3 shows these performance metrics for this model.

Table 3: Test Errors for Random Forest Model 3

Test Error	*Control group*	*Event Group*	*Combined Group*
Random Forest (Model 3)	-0.65981%	1.8036%	0.4752%

A few very important observations can be made from looking at the overall performance of all 12 models tested. In general, the models tended to perform best on the "Control" group, worst on the "Event" group, and somewhere in between on the "Combined" group. (There is some randomness in our results, so this is not unilaterally true, but it does appear to be a trend.) This result makes intuitive sense, given that it is hard to predict how people respond to notifications that their energy prices will be significantly higher during a predetermined window of time, and that they have enough notice to actually make adjustments to their actions in order to potentially reduce energy consumption in the prediction window.

In this project, one of our goals was to gain a better sense of whether or not it is indeed more difficult to predict energy consumption behavior for people who do receive price notifications, and our results indicate that this is true to some extent. We did, however, find that some models are able to perform well even with this information about price notifications. This is the main reason we chose RF3 as our best model, because it did well on the "Event" group, and comparably well on the "Control" and "Combined" groups.

VI. Conclusions & Future Work

In this project, we built models to predict day-ahead energy consumption for different groups of individuals: those who receive notification that their energy prices will increase, those who do not, and a combined group of both subsets. We used OLS, Random Forests, SVMs, and ARIMA models to make predictions, and we tested the accuracy of these models both on the consumption of individual households and on the aggregated consumption. We found the Random Forest to be the best model, and we note that it is, in general, harder to predict day-ahead energy consumption for those who do receive notifications.

Given more time to work on this project, we would like to spend more time thinking about which variables have the most importance in predicting energy consumption. Furthermore, we would like to find ways to improve the accuracy of our models on the "Event Group," as this is the group that is currently the hardest to predict, since it is difficult to model how people change their behavior after they receive new information.

Contributions

J.R. found the raw dataset and constructed the dataframes. J.R., A.S., and B.L. conceived the goals and structure of the project. J.R. and A.S. built the OLS models. B.L. built the Random Forest models. A.S. built the SVM models. J.R. built the ARIMA models. J.R., A.S., and B.L. interpreted the results, designed the poster, and wrote the report.

References

[1] Peter Lusis, Kaveh Rajab Khalilpour, Lachlan Andrew, Ariel Liebman, Short-term residential load forecasting: Impact of calendar effects and forecast granularity, In Applied Energy, Volume 205, 2017, Pages 654-669, ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2017.07.114. 669, ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2017.07.114.

[2] Rishee K. Jain, Kevin M. Smith, Patricia J. Culligan, John E. Taylor, Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy, In Applied Energy, Volume 123, 2014, Pages 168-178, ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2014.02.057.

[3] L. Suganthi, Anand A. Samuel, Energy models for demand forecasting—A review, In Renewable and Sustainable Energy Reviews, Volume 16, Issue 2, 2012, Pages 1223-1240, ISSN 1364-0321, https://doi.org/10.1016/j.rser.2011.08.014.

[4] Lukas G. Swan, V. Ismet Ugursal, Modeling of end-use energy consumption in the residential sector: A review of modeling techniques, In Renewable and Sustainable Energy Reviews, Volume 13, Issue 8, 2009, Pages 1819-1835, ISSN 1364-0321, https://doi.org/10.1016/j.rser.2008.09.033.

[5] A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah, R. Saidur, A review on applications of ANN and SVM for building electrical energy consumption forecasting, In Renewable and Sustainable Energy Reviews, Volume 33, 2014, Pages 102-109, ISSN 1364-0321, https://doi.org/10.1016/j.rser.2014.01.069.

[6] A. Lahouar, J. Ben Hadj Slama, Day-ahead load forecast using random forest and expert input selection, In Energy Conversion and Management, Volume 103, 2015, Pages 1040-1051, ISSN 0196-8904, https://doi.org/10.1016/j.enconman.2015.07.041.

[7] http://www.pecanstreet.org/studentresearch/

[8] http://www.svms.org/regression/SmSc98.pdf