Timeseries Prediction

Powersupply Stream of Italy between 1995 and 1998

Tamara Feiertag, Christine Waltl

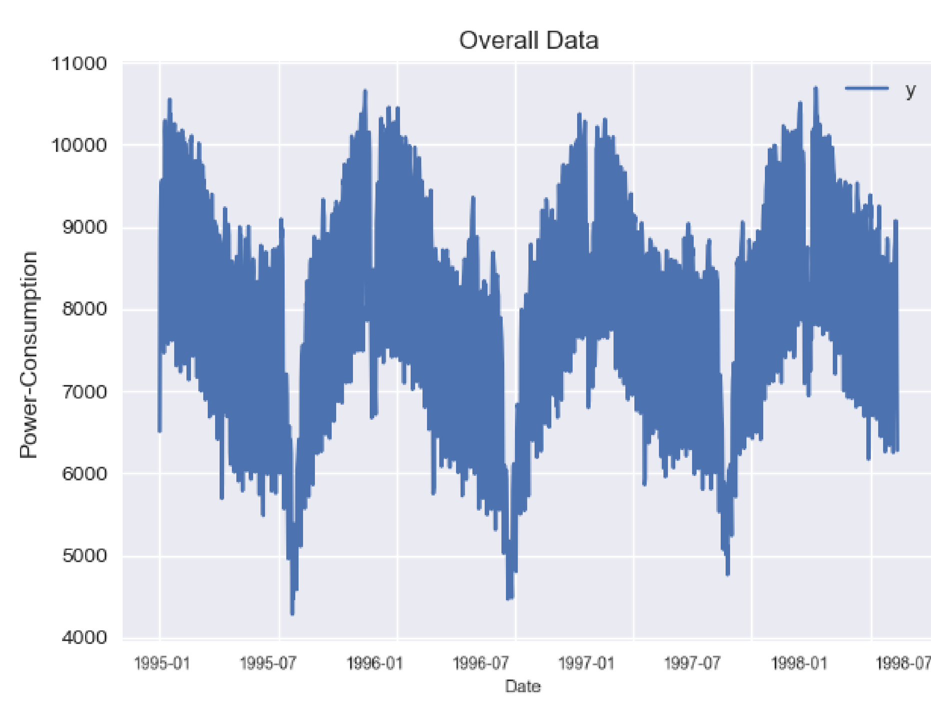
Extended Problem Statement

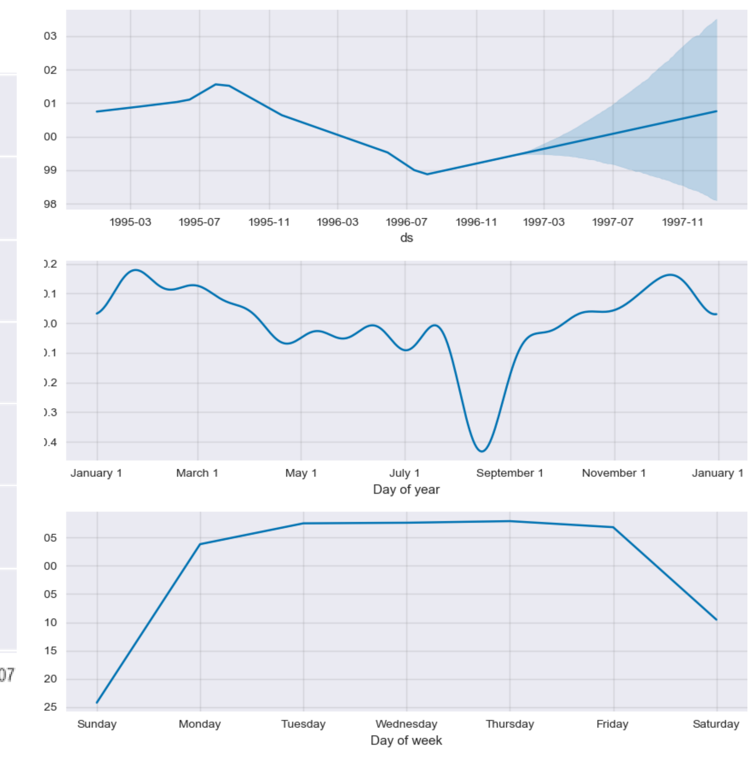
From historic daily power consumption data in Italy from 1995 to 1998, the goal is to predict future power demand. We worked with Autoregressive Integrated Moving Average Model (ARIMA), Holt-Winters, recurrent neural networks and Prophet in Python, and compared the results.

In addition, we were interested in whether or not public holidays have a strong impact on power consumption. Therefore, we created an algorithm that checks when the actual consumption deviates from our model, and maps those dates to possible events.

Dataset

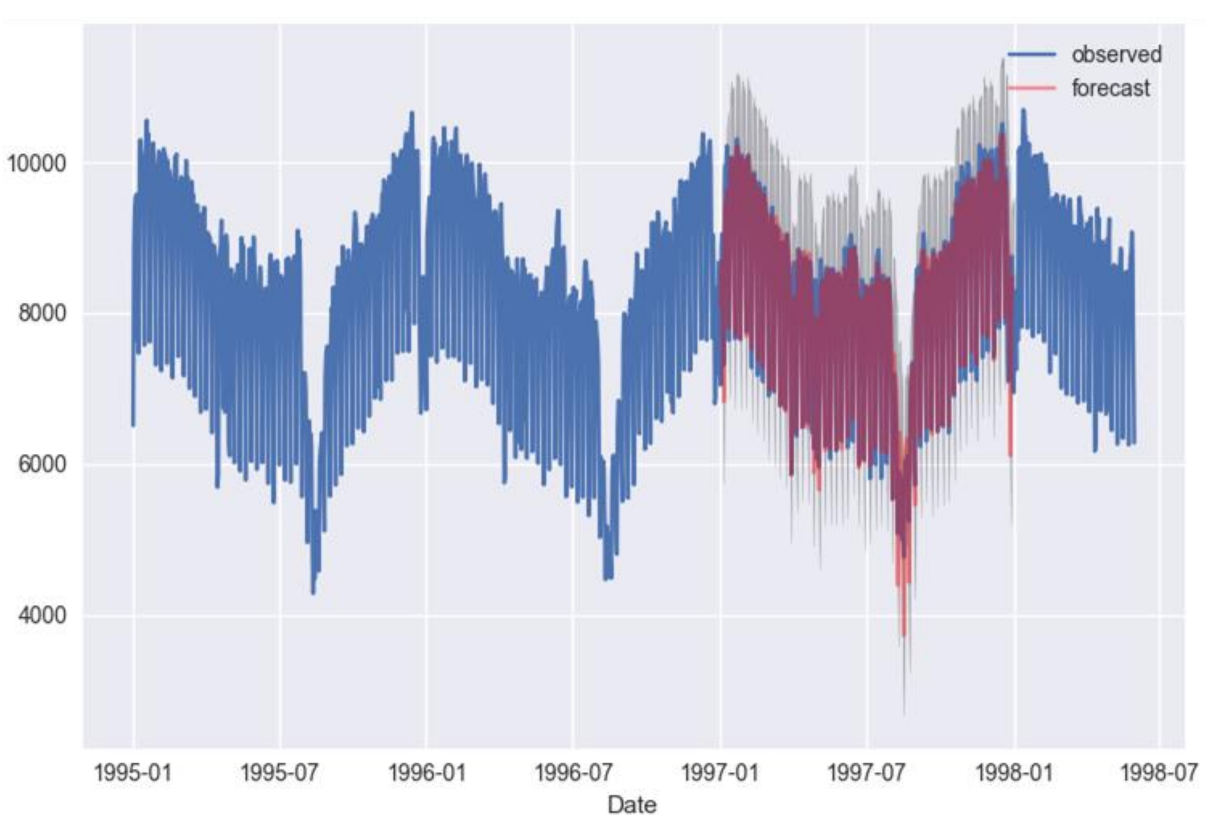
The power consumption data set is a univariate time series. Plotting the total power supply per day (left figure), reveals trend and seasonality within the data. The decomposition plot (right figure) of Prophet confirmed that.





Results

SARIMA(2,1,6)(1,1,6)$_7$

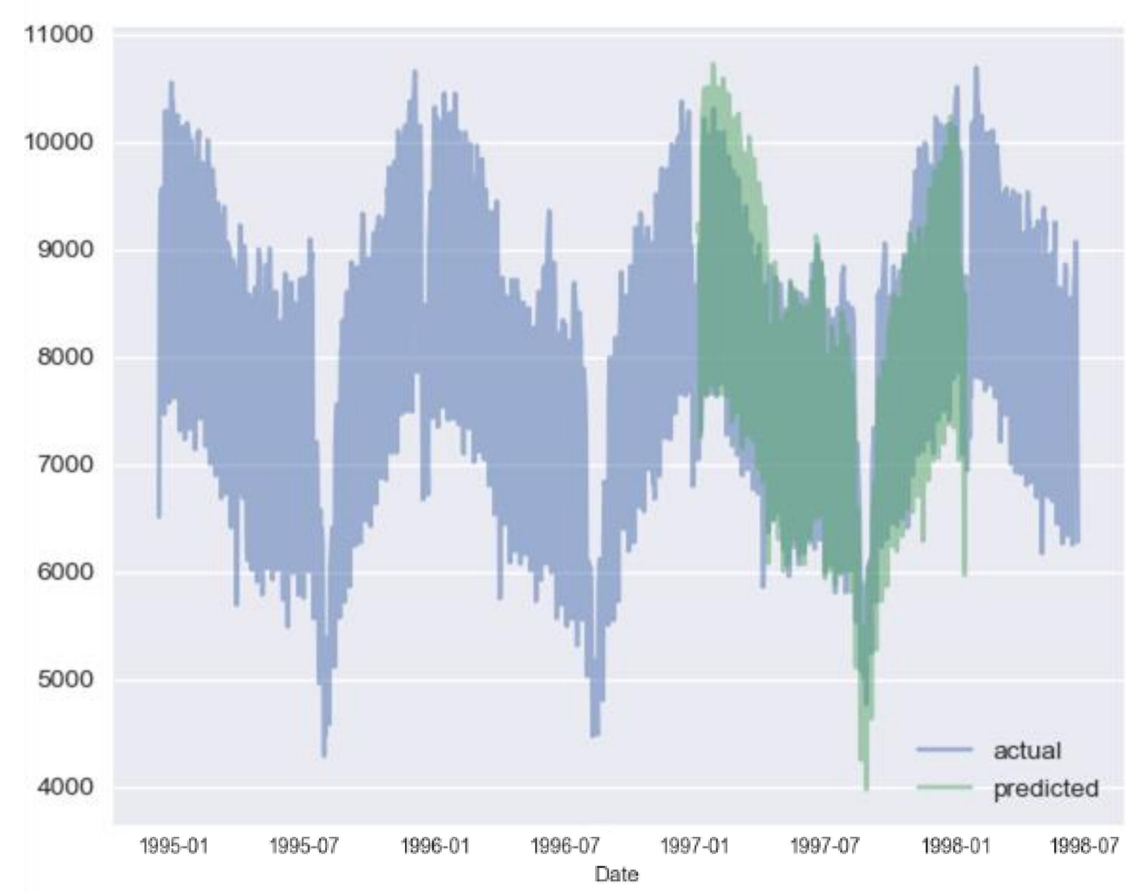


ARIMA(p,d,q)(P,D,Q)s

1. Remove trend and seasonality by differencing
   => until data stationary
1. Find parameters p, q by using
   a. ACF plot
   b. PACF plot
2. Find periodicity of data for s
3. Apply seasonal differencing
4. Compare parameter combinations to find seasonal parameters P, Q
   a. lowest AIC (how well model fits data)
5. Fit model on training data (2 y)
6. Plot residuals. If contain information, adapt parameters.
7. Do predictions & add back trend

RMSE: 337.81

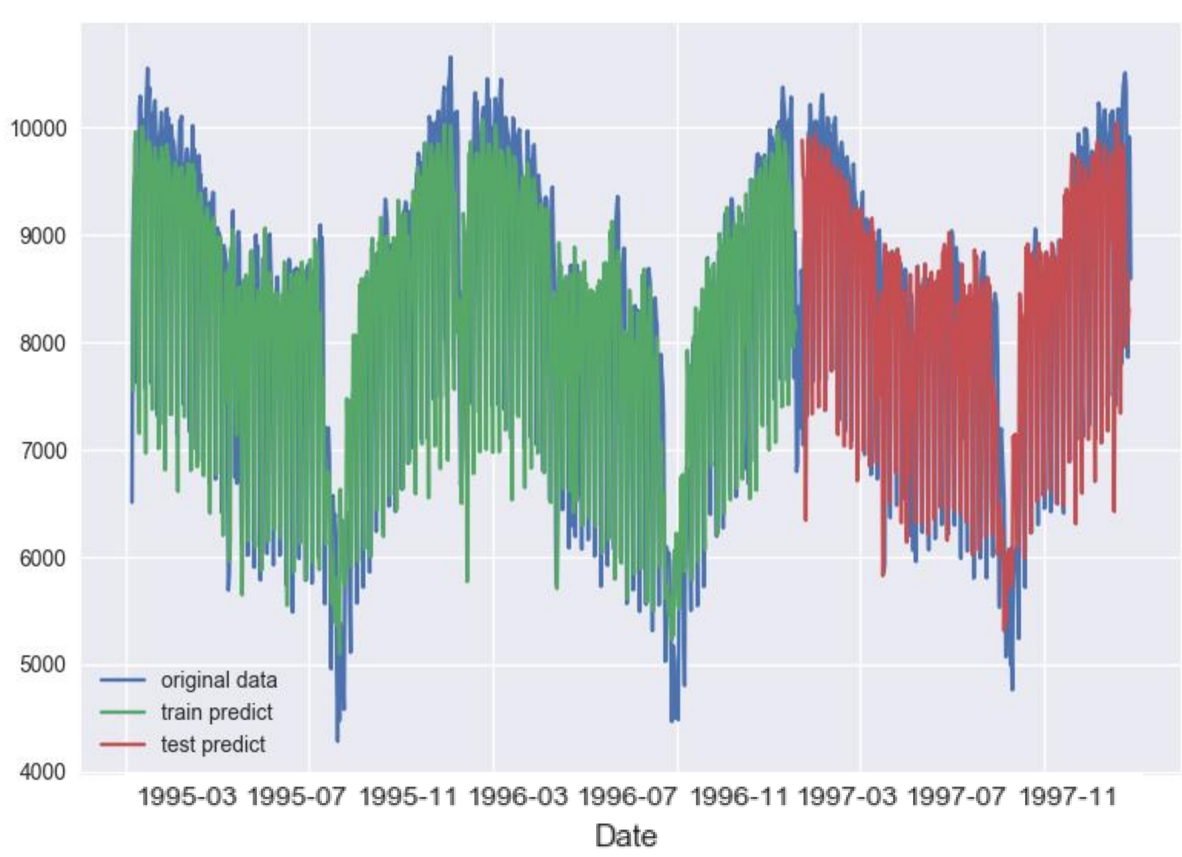
Holt-Winters Method (triple exponential smoothing)



1. Filter first two years
2. Apply log on data
3. Subtract trend
4. Find parameters for triple_exponential_smoothing(alpha, beta, gamma) by looping through combinations and compare the r2_score of HW which the combination would produce
   a. alpha = 0.99
   b. beta = 0.2386
   c. gamma = 1
5. Apply Holt-Winters method with parameters
6. Add back trend to prediction
7. Apply exp on prediction

RMSE: 555.36

RNN: Long Short-Term Memory

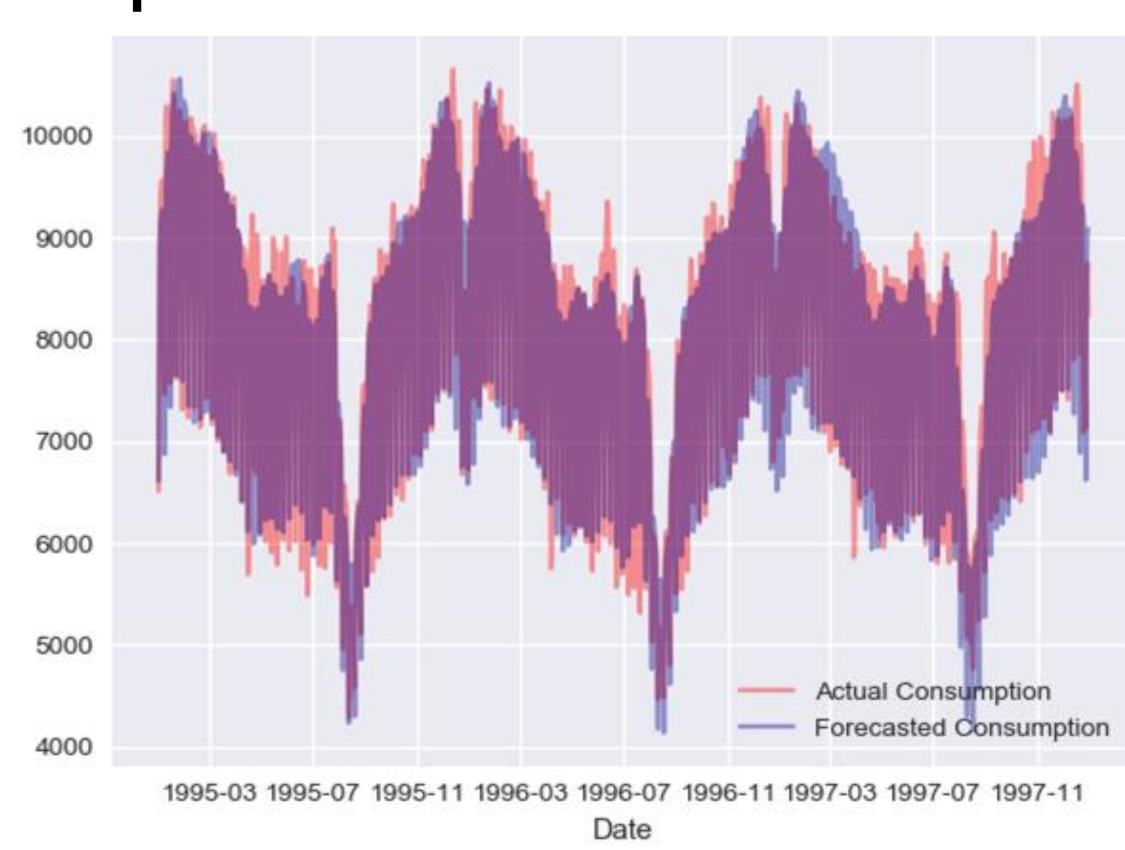


LSTM

1. Scale data within [-1, 1] for hyperbolic tangent activation func.
2. ~~Stationarize data~~
3. Transform to supervised learning problem
   a. inputs are lag observations
   b. output is current value
4. Design and fit LSTM
   a. tensorflow
   b. find best #neurons - #epochs combinations (MSE)
5. Finally, we used 2 neurons and 100 epochs
6. Predict test and training data
7. Scale predictions back

RMSE: 477.19

Prophet



Prophet Steps:

1. Filter first two years
2. Apply log on data
3. Fit prophet
4. Forecast
5. Apply exp on data

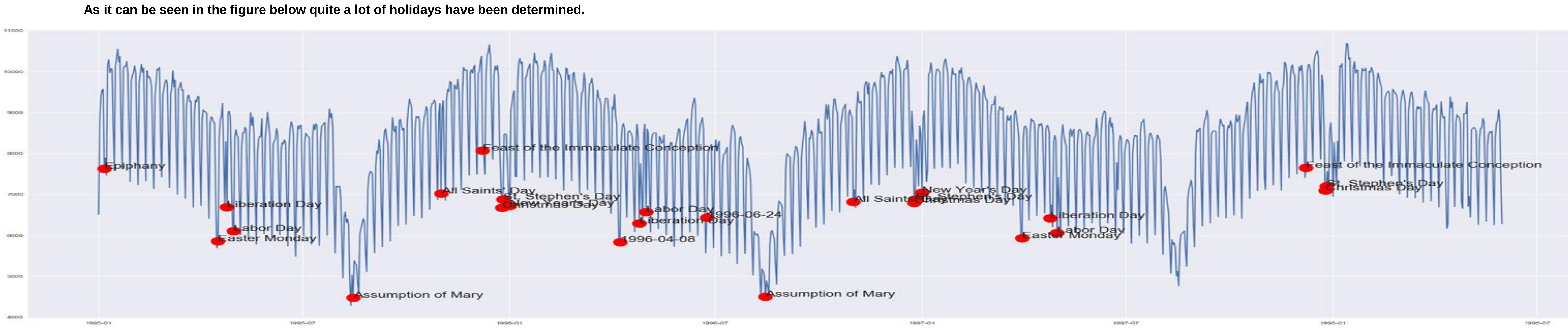
RMSE: 491.30

Event Detection with Prophet

Applied steps:

1. Filter first two years
2. Apply log on data
3. Fit prophet to have a model
4. Compare existing data with model
   a. All points with a deviation of 20% are considered as abnormal (=event)
   b. If the specific abnormal date was a public holiday then link to the name

As it can be seen in the figure below quite a lot of holidays have been determined.



References

[1] Github, 2018. *Prophet: Automatic Forecasting Procedure*. https://github.com/facebook/prophet (12.06.2018)
[2] Udacity, Inc. 2018. *Time Series Forecasting*. https://classroom.udacity.com (12.06.2018)
[3] Brownlee, J. 2017. *LSTM article collection*. https://machinelearningmastery.com (12.06.2018)
[4] Online OTexts. 2018. *Forecasting*. https://www.otexts.org (12.06.2018)
[5] Brockwell, Peter J., and Richard A. Davis. *Introduction to time series and forecasting*. springer, 2016.
[6] Shumway, Robert H., and David S. Stoffer. *Time series analysis and its applications*. springer, 2011.
[7] Hyndman, Rob J., and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.
[8] Hochreiter S., Schmidhuber J. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.