Online Conditional Outlier Detection in Nonstationary Time Series

Siqi Liu[1], Adam Wright[2], and Milos Hauskrecht[1]

[1]Department of Computer Science, University of Pittsburgh

[2]Brigham and Women's Hospital and Harvard Medical School

Outline

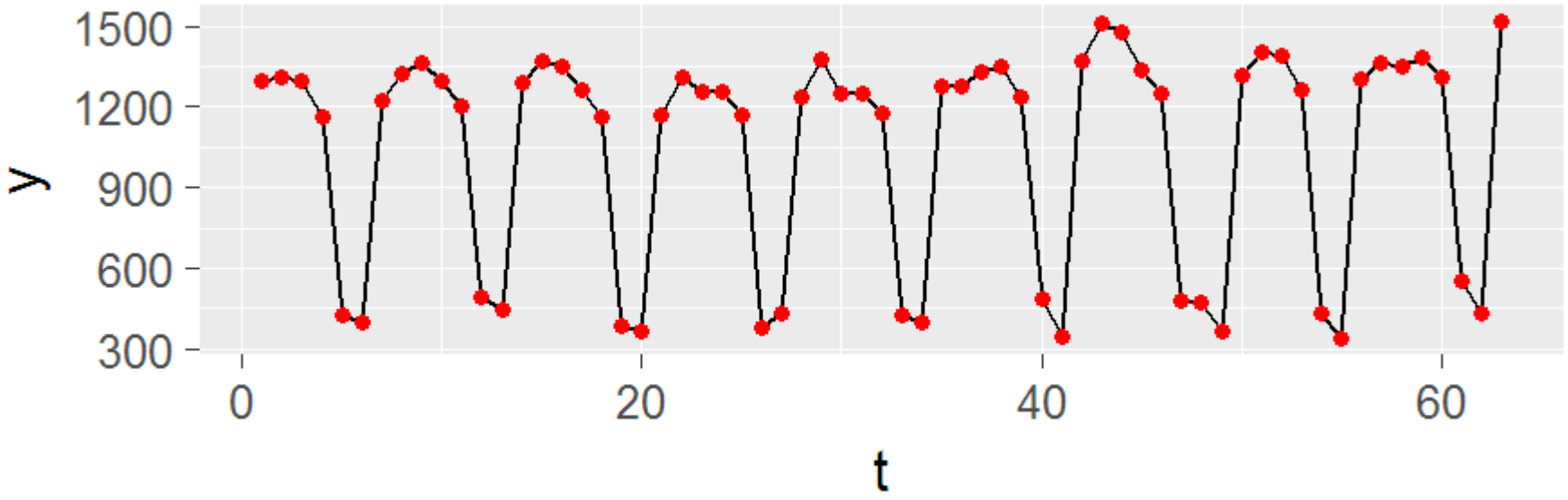
- **Introduction**
- Method
- Experiments and Results
- Conclusion

Time Series

- A time series is a sequence of data points indexed by (discrete) time.
- For example, a univariate time series

$$\{y_t \in \mathbb{R}: t = 1, 2, \dots\}.$$

- Generally, the points are **not independent** of each other.



Time Series are Everywhere

- Daily prices of stocks
- Monthly usage of electricity
- Daily temperature, humidity, ...
- Patient's heart rate, blood pressure, ...
- Number of items sold every month
- Number of cars passed through a highway every hour
- …

Monitoring Systems

- By monitoring some attribute of a target (e.g., the heart rate of a patient), we naturally get a time series.
- Analyzing the time series gives us insights about the target.
- In this work, we are interested in finding **outliers** in the time series in real time.

Outliers

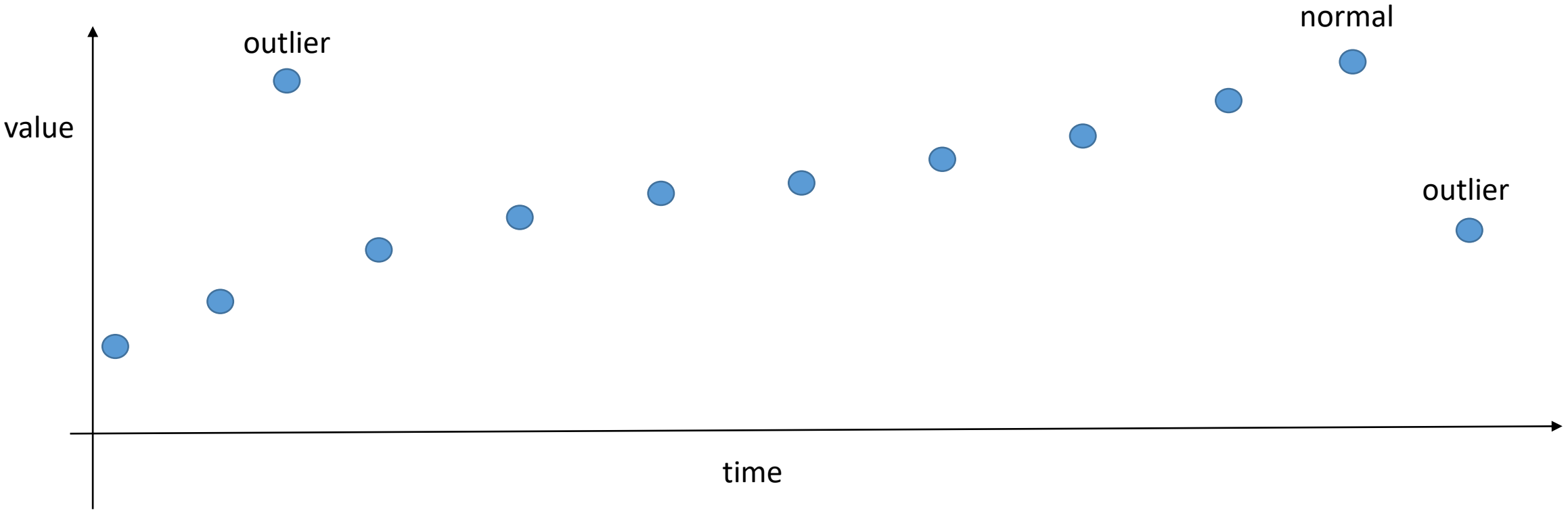
- Outliers are the points that do not follow the “pattern” of the majority of the data.
- More strictly, they are points that do not follow the probability distribution generating the majority of the data.
- Outliers provide useful insights, because they indicate anomaly or novelty, i.e., events requiring attention.
  - extremely low volume on a highway → traffic accident
  - unusually frequent access to a server → server being attacked
  - increasing use of a rare word on a social network → new trending topic

Challenge 1: Nonstationarity

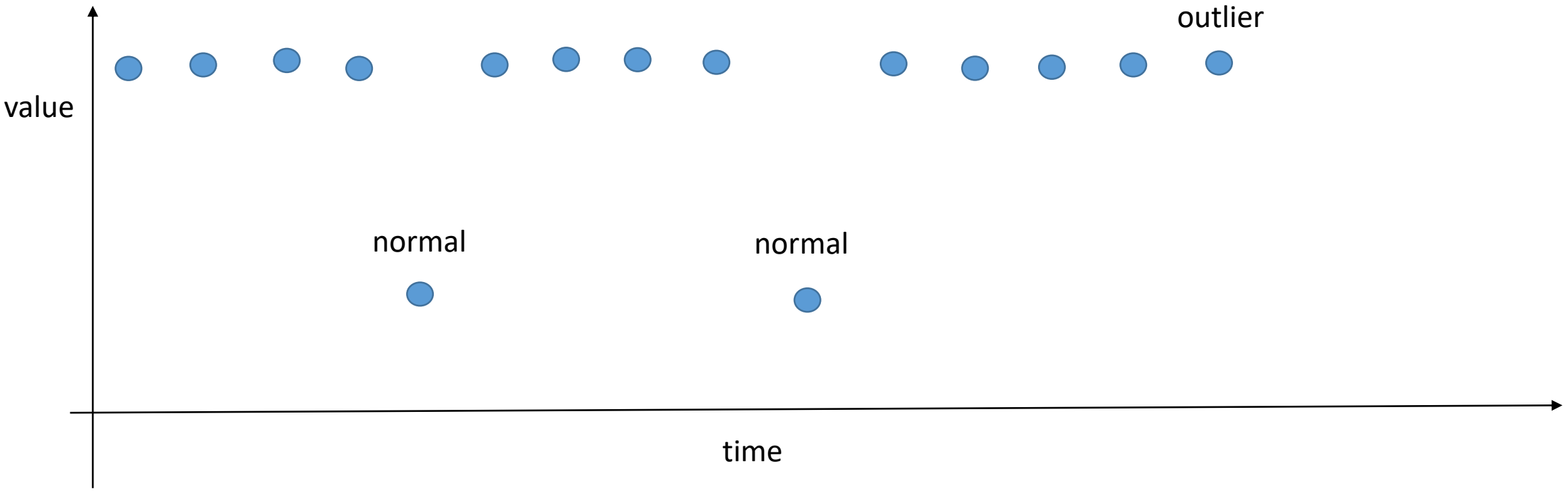
- Detecting outliers in time series is challenging because of the nonstationarity (i.e., the distribution of the data changes over time)
- Specifically, the changes could be
  - long-term changes
  - periodic changes (a.k.a. seasonality)
- These hinder outlier detection, because they result in false positives and false negatives

Challenge 1.1: Long-Term Changes

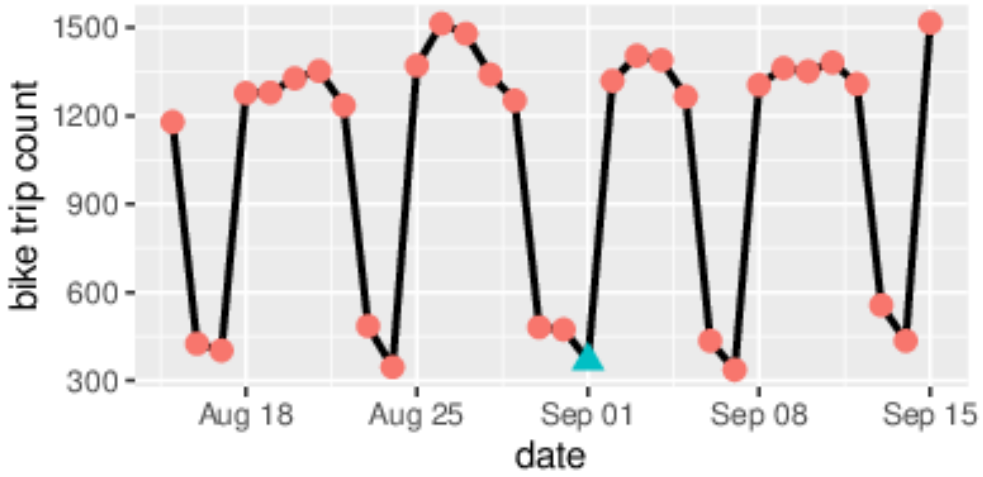
- An extreme value in the past could be normal now
- A normal value in the past could be extreme now



Challenge 1.2: Seasonality



Challenge 2: Context



- By considering the context, some “outliers” become normal; some “normal” points become outliers.

Existing Work in Outlier Detection in Time Series

- Existing work in outlier detection in time series usually assumes a model like autoregressive-moving-average (ARMA). (e.g., Tsay 1988; Yamanishi and Takeuchi 2002)
- These models cannot deal with nonstationary (seasonal) time series directly.
- A solution is to difference the time series, resulting in: autoregressive-integrated-moving-average (ARIMA).
- We use it as a baseline in our experiments.

Outline

- Introduction
- **Method**
- Experiments and Results
- Conclusion

Method: Two Layers

- A time $t = 1, 2, \ldots$ we **sequentially** receive the observations of the target time series

$$y = \{y_t \in \mathbb{R} : t = 1,2, \ldots\},$$

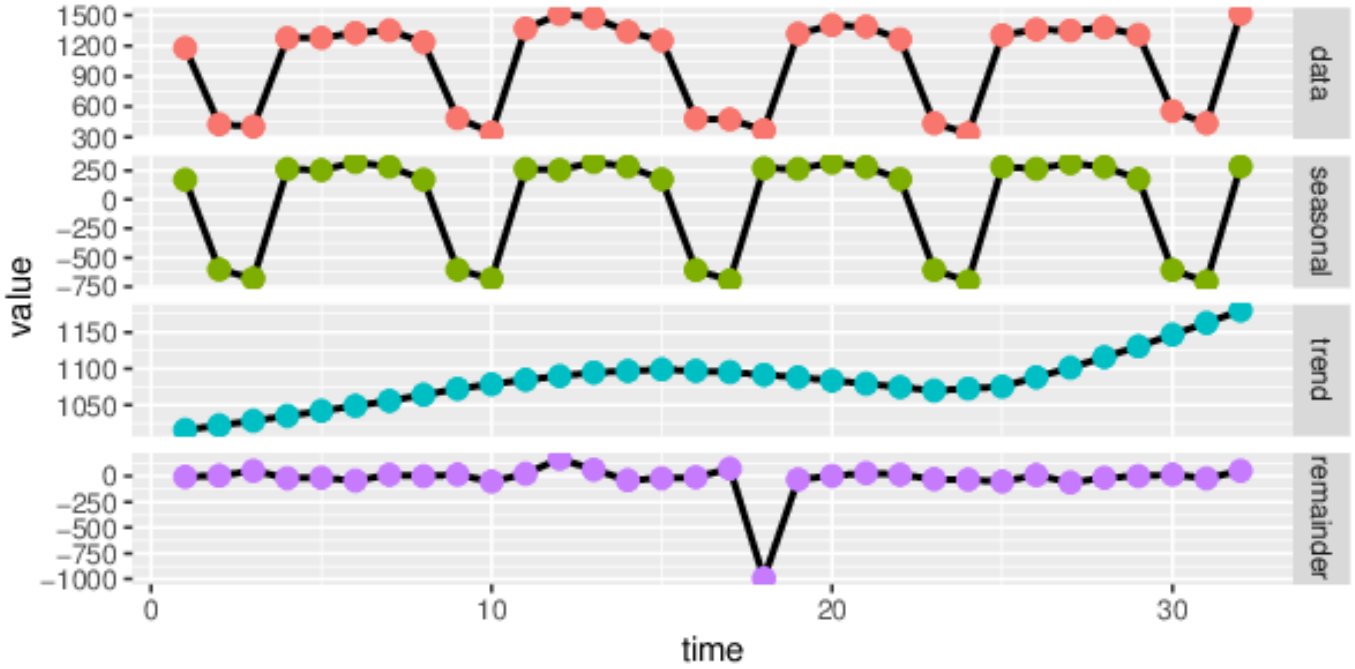
  and the associated context variables

$$x = \{x_t \in \mathbb{R}^p : t = 1,2, \ldots\}.$$

- Our model consists of two layers:
  - First layer uses a **sliding window** to compute a **local** score;
  - Second layer combines the local score with the context variables to compute a **global** score (which is the final outlier score).

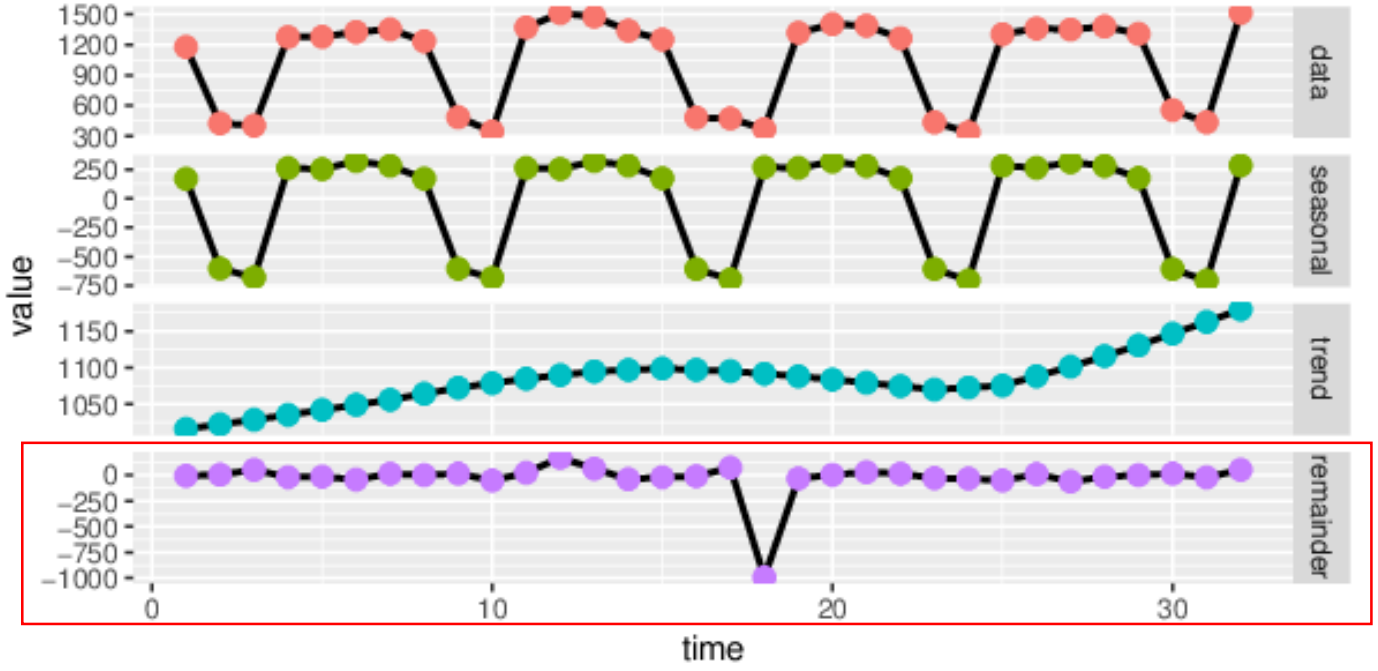
First Layer

- First, we decompose the time series (within a sliding window) into 3 components using a nonparametric decomposition algorithm called STL (Cleveland et al. 1990).



First Layer

- Then, we compute a local deviation score $z_t = \frac{y_t^{(R)} - \hat{\mu}_t^{(R)}}{\hat{\sigma}_t^{(R)}}$ on the remainder.



Second Layer

- At each time t, given (z_t, x_t), where z_t is the local score (first-layer output) and x_t is the context variables, keep updating a Bayesian linear model

$$z_t|w, \beta, x_t \sim N(x_t^T w, \beta^{-1}),$$

  with the conjugate prior

$$w, \beta \sim N(w|m_0, \beta^{-1} S_0) Gam(\beta|a_0, b_0).$$

- The model is built globally (aggregating all the information from the beginning), because
  - contextual variables may correspond to rare events (e.g., holidays), but we need enough examples to have a good model;
  - local scores are normalized locally, so no need to worry about nonstationarity.

Second Layer

- The final outlier score is calculated based on the marginal distribution of z_t given x_t and the history

$$z_t|D_{t-1}, x_t \sim St(z_t|\mu_t, \sigma_t^2, \nu_t),$$

where $D_{t-1} = \{(\mathrm{z_u}, \mathrm{x_u}) \mid \mathrm{u} = 1, 2, \ldots, t-1\}$.

Outline

- Introduction
- Method
- **Experiments and Results**
- Conclusion

Data Sets

- **Bike data** consists of the time series (of length 733) that records the daily bike trip counts taken in San Francisco Bay Area through the bike share system from August 2013 to August 2015 .
- **CDS data** consists of daily rule firing counts of a clinical decision support (CDS) system in a large teaching hospital. (111 time series of length 1187)
- **Traffic data** consists of time series of vehicular traffic volume measurements collected by sensors placed on major highways. (2 time series of length 365)

Injecting Outliers

- Outliers are injected into the time series by randomly sampling a small proportion p of points and changing their value by a specified size δ as

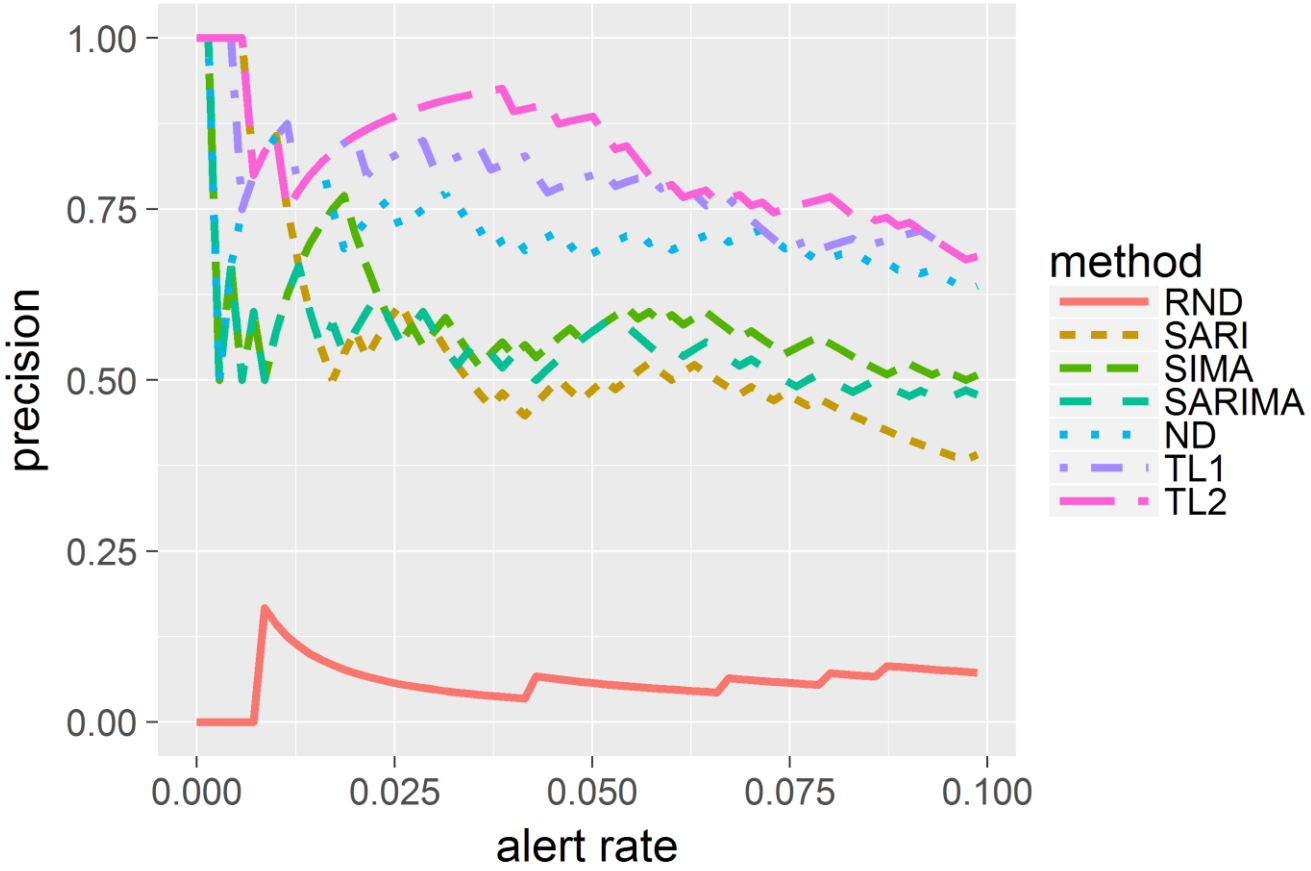
$$y_i = y_i \cdot \delta$$

  for each y_i in the sample.
- We vary p and δ to see the effects.

Methods Compared

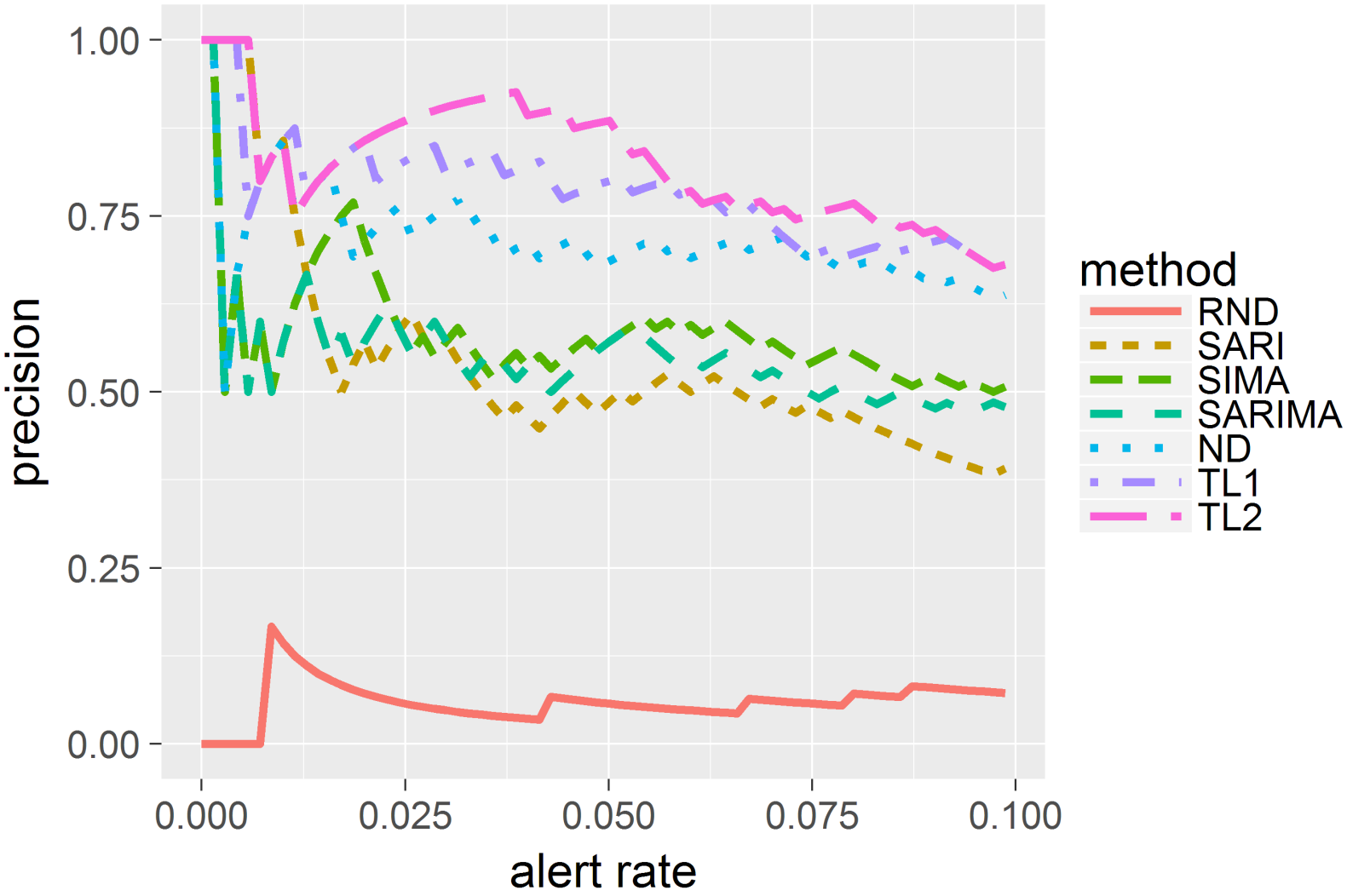
- RND - detects outliers randomly.
- SARI - $\mathrm{ARIMA}(1,1,0) \times (1,1,0)_7$, ARIMA with a weekly (7 day) period, (seasonal) differencing, and (seasonal) order-1 autoregressive term.
- SIMA - $\mathrm{ARIMA}(0,1,1) \times (0,1,1)_7$, ARIMA with a weekly period, (seasonal) differencing, and (seasonal) order-1 moving-average term.
- SARIMA - $\mathrm{ARIMA}(1,1,1) \times (1,1,1)_7$, ARIMA combining the above two.
- ND - our first-layer STL-based model, using absolute value of the output as outlier scores.
- TL1 - our two-layer model using holiday information as a contextual variable.
- TL2 - our two-layer model using holiday and additional information (if available) as context variables.

Evaluation: Precision-Alert-Rate Curves (Hauskrecht et al. 2016)

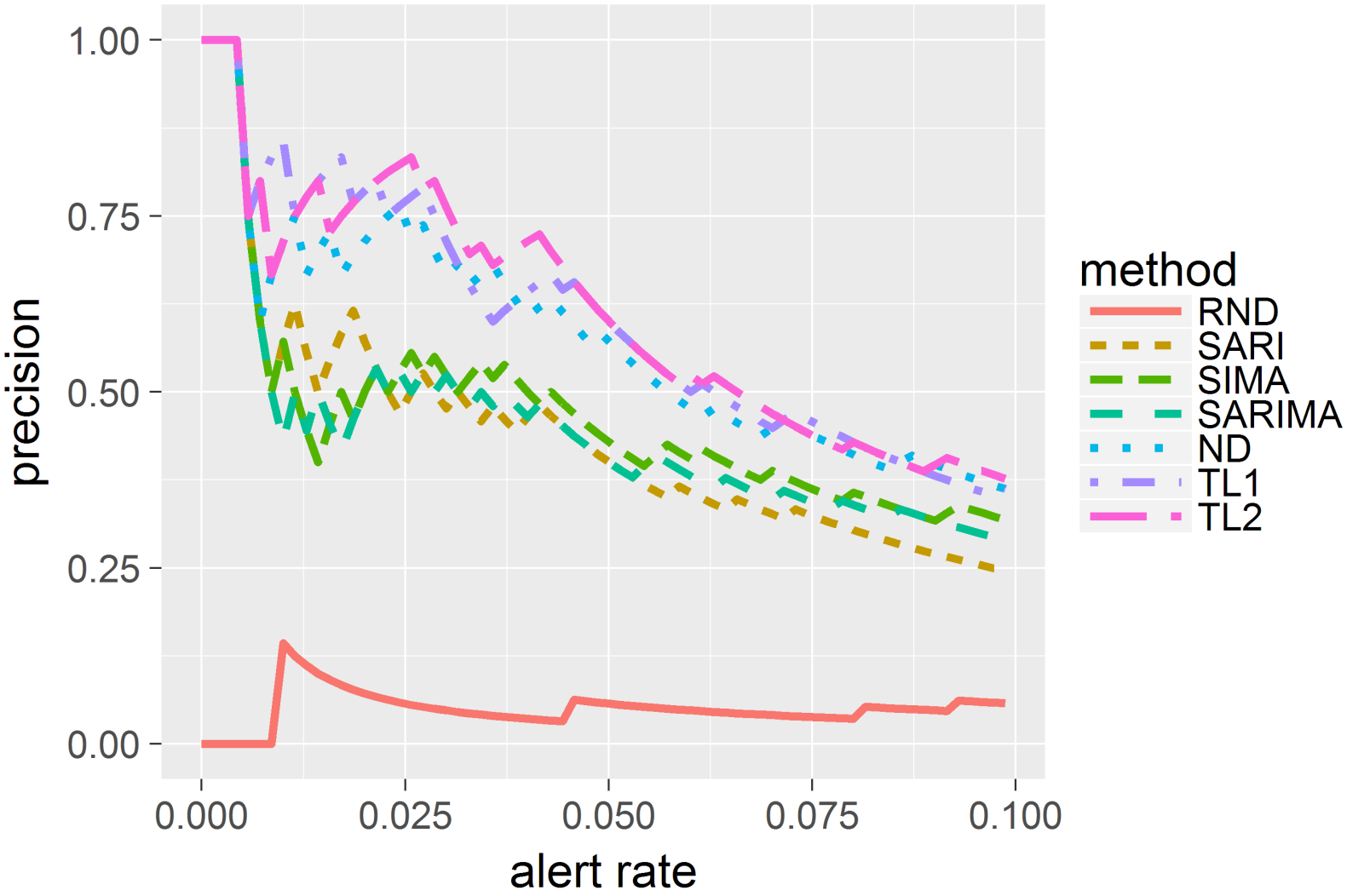


- Alert rate: the proportion of alerts raised out of all points.
- Precision: the proportion of true outliers out of alerts raised.
- We calculate AUC to compare the overall performance.
- Notice we focus on low-alert-rate region for practicality.

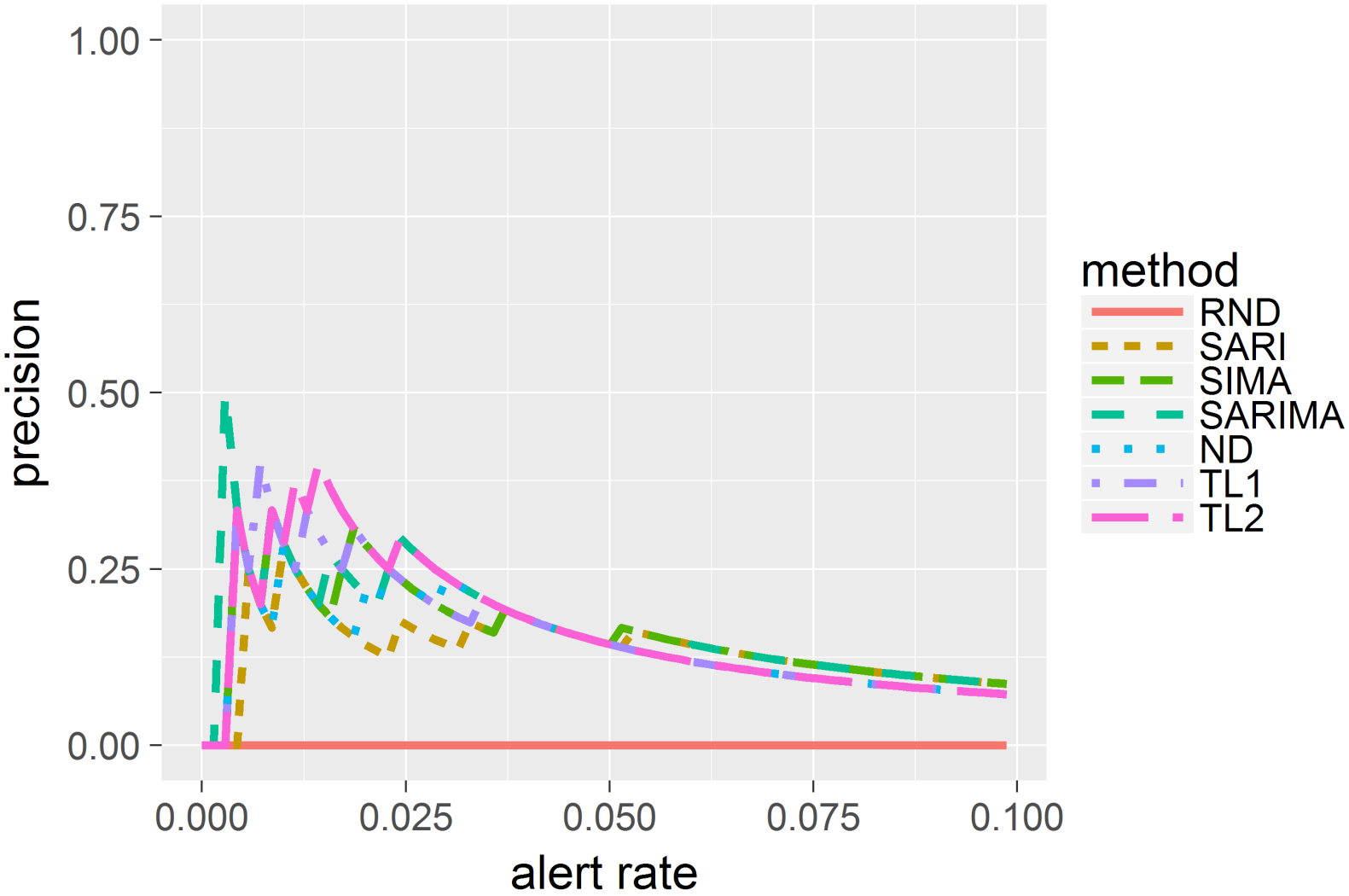
Varying Outlier Rate (0.1) - Bike Data



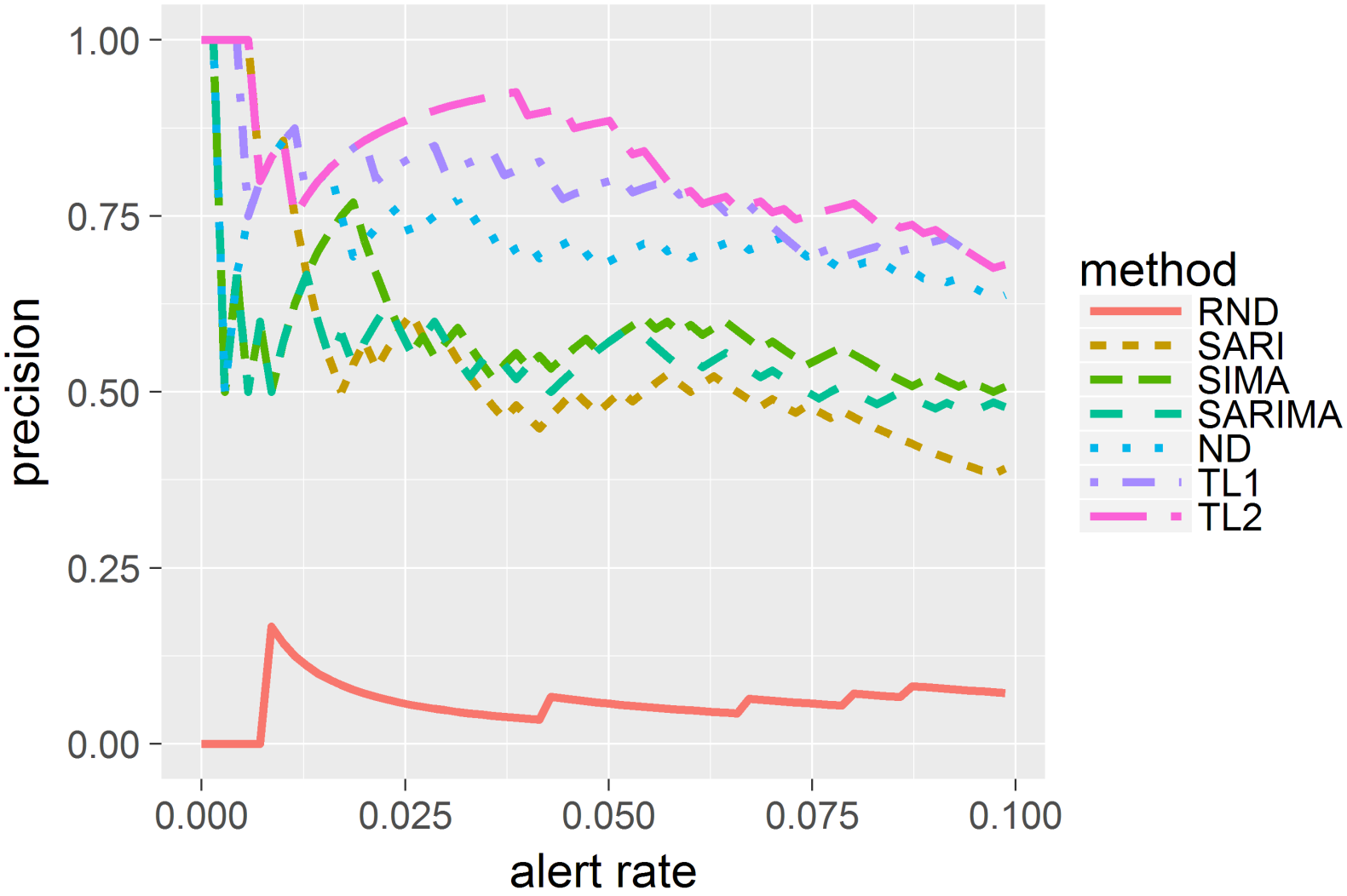
Varying Outlier Rate (0.05) - Bike Data



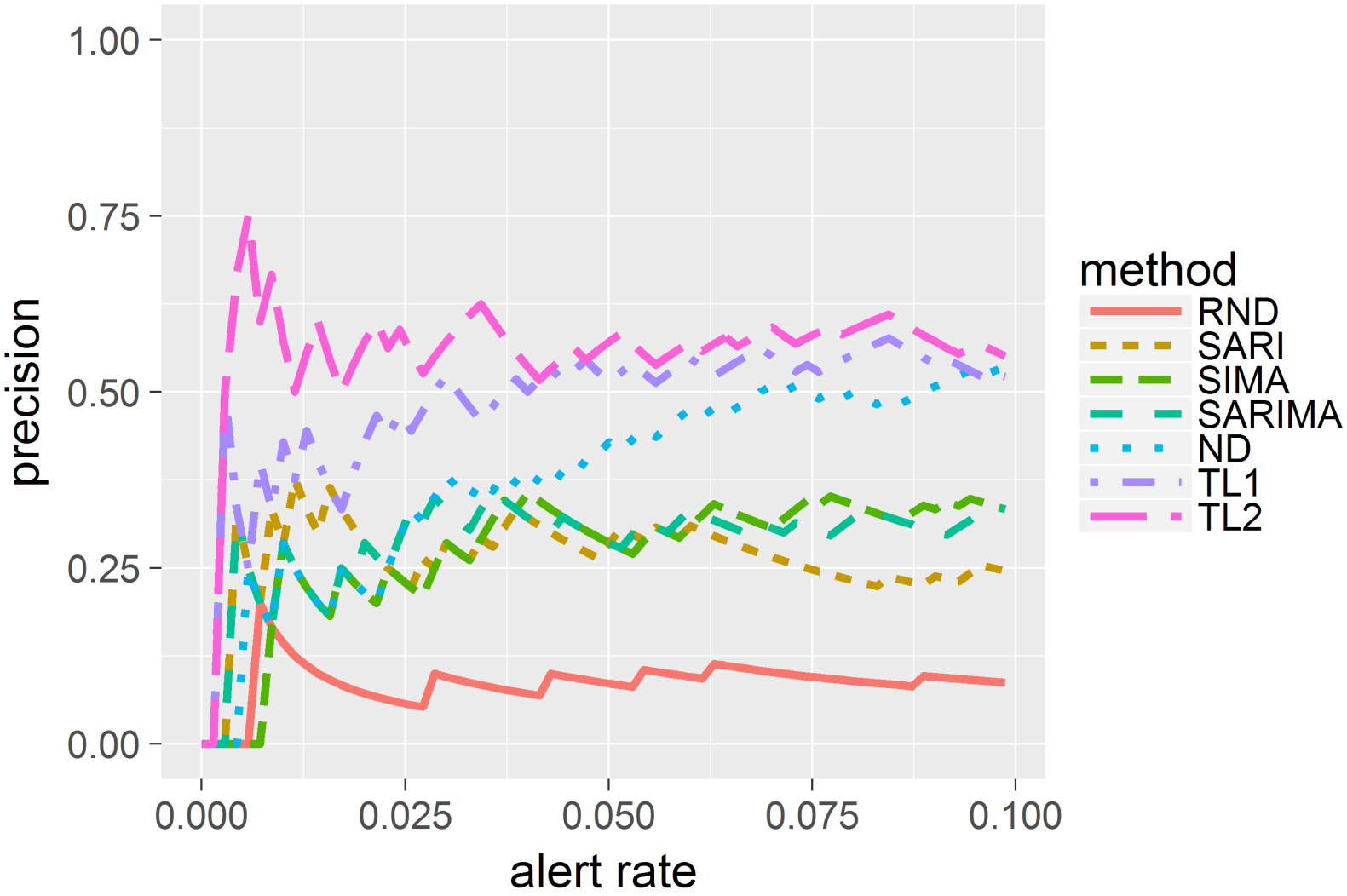
Varying Outlier Rate (0.01) - Bike Data



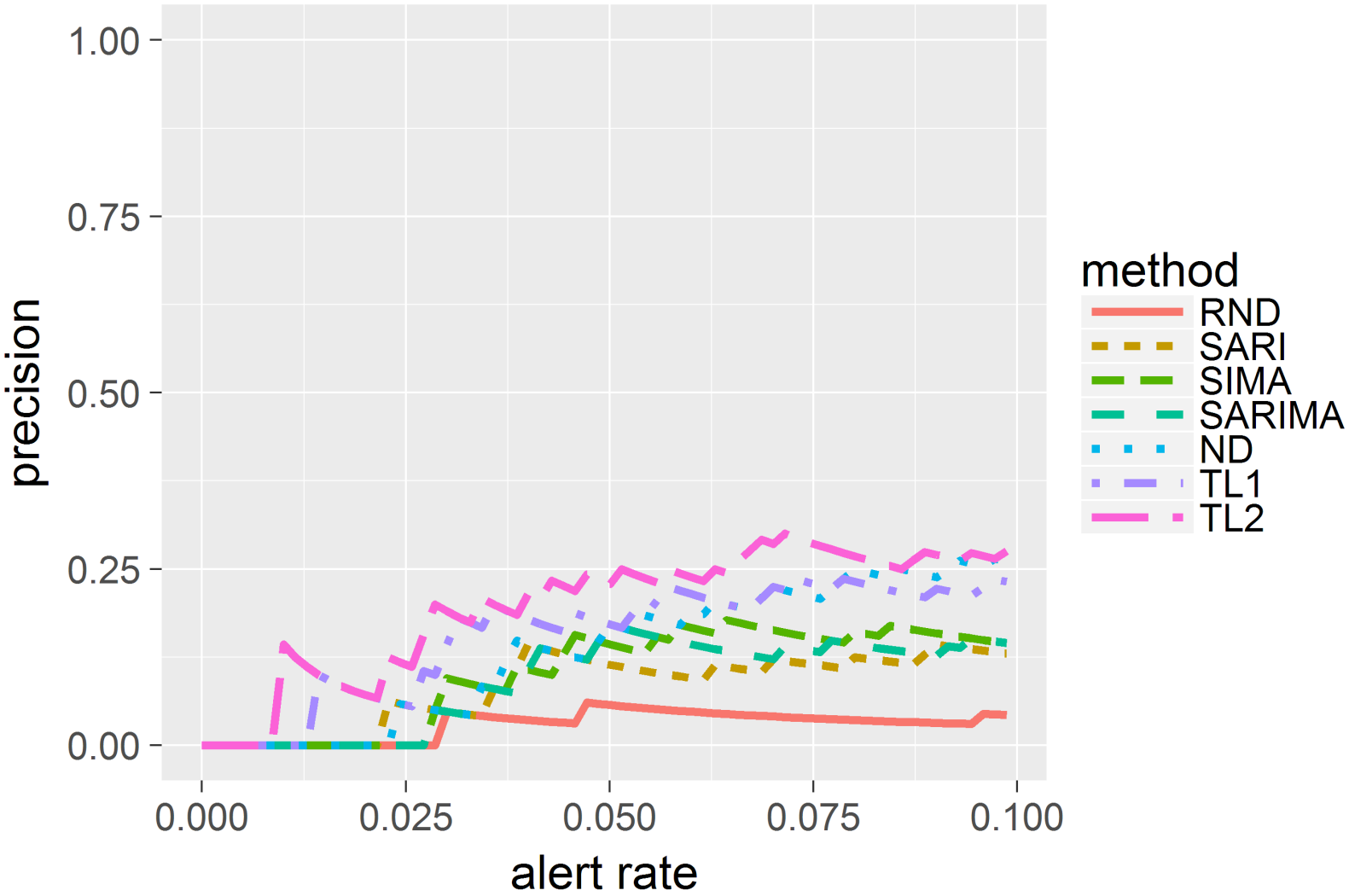
Varying Outlier Size (2.0) - Bike Data



Varying Outlier Size (1.5) - Bike Data



Varying Outlier Size (1.2) - Bike Data



Overall Performance

- By comparing the AUC, we have the following observations:
  - When the size of the outliers are small, all methods perform similarly to random.
  - In the other cases, our two-layer method is almost always the best method.
  - Even using only the first-layer can achieve similar or better results as the ARIMA-based methods.
  - Using additional information (e.g., using weather besides holiday info) improves the performance of the two-layer method.

Outline

- Introduction
- Method
- Experiments and Results
- **Conclusion**

Conclusion

- We have proposed a two-layer method to detect outliers in time series in real time.
- Our method takes account of the nonstationarity and the context of the data to detect outliers more accurately.
- Experiments on data sets from different domains have shown the advantages of our method.

Thank you!
Q & A