Assignment - **Energy-efficient Distributed Systems (194.049)** - 2019S

EDGE-CLOUD MECHANISM FOR AUTOMATIC AND ACCURATE RECOVERY OF INCOMPLETE SENSOR DATA

Project Assistant Ivan Lujić (ivan@ec.tuwien.ac.at), Univ. Prof. Dr. Ivona Brandić (ivona@ec.tuwien.ac.at)

BACKGROUND AND MOTIVATION

Massive amounts of data are constantly generated from growing number of Internet of Things (IoT) devices. Nowadays, sensor-based data are used in many applications, like healthcare monitoring, intelligent traffic management, automated smart home and building systems. Such systems rely on monitoring of parameters such as temperature, movement, heart rate, electricity consumption, radiation and air quality, coming from many sensors. Managing such systems generally requires three steps: sensor data collection, data processing and taking actions based on the obtained results. In most of existing systems, such processing is done by employing geographically distributed and massive data centers. Recently, edge analytics has been proposed as a solution to perform near real-time decisions in proximity of the user. Edge analytic relies on edge nodes such as micro data centers or edge gateways, which are smaller scale cloud data centers, that can be deployed closer to IoT systems. By placing data analytics close to the source of data, edge nodes can minimize required latency for decision making processes.

Nevertheless, errors, missing values and outliers may appear in data collected by IoT sensors at the edge, due to (1) the highly distributed nature of IoT systems; (2) monitoring system failures; (3) data packet loss in sensor networks; (4) aging of the sensor; (5) changes in external conditions; or (6) periodic sensor failures, and thus affecting both near real-time and batch analytics. When performing data analytics on the edge, we need complete datasets to timely perform decisions. However, we still have to deal with limited storage capacities that can hinder accuracy of edge analytics and consequently decisions for IoT apps.

Figure 1 represents edge analytics model for smart buildings use case. In step (1), data are collected from smart buildings. Due to the increasing amount of data generated by smart buildings, data are transferred to the edge layer in step (2). Once certain amount of data is transferred, monitoring component receives data and detects outliers and missing values in step (3). Monitoring component notifies mediator component about incomplete data in step (4). Mediator keeps information about appropriate forecast method and ranges of historical data for particular gap length (number of missing values). Due to limited storage capacities, mediator might ask cloud repository for certain amount of historical data to help recovering wider gaps at the edge. Then, adaptive recovery process is performed in step (5), whose output is the dataset without gaps and cleaned from outliers. This output is then forwarded to edge storage and used by local analytics in step (6). The analytic results are either stored or used to take operative decisions (for example, commands sent to actuators) in step (7). Depending on applications, batch analytics can be performed in the cloud.

Generally, data generated by sensor-based monitoring are classified as time series data. Based on a certain amount of historical time series and recognized pattern, it is possible to predict future values. Accordingly, we can use predictive analytics in the context of recovering gaps in incomplete datasets.

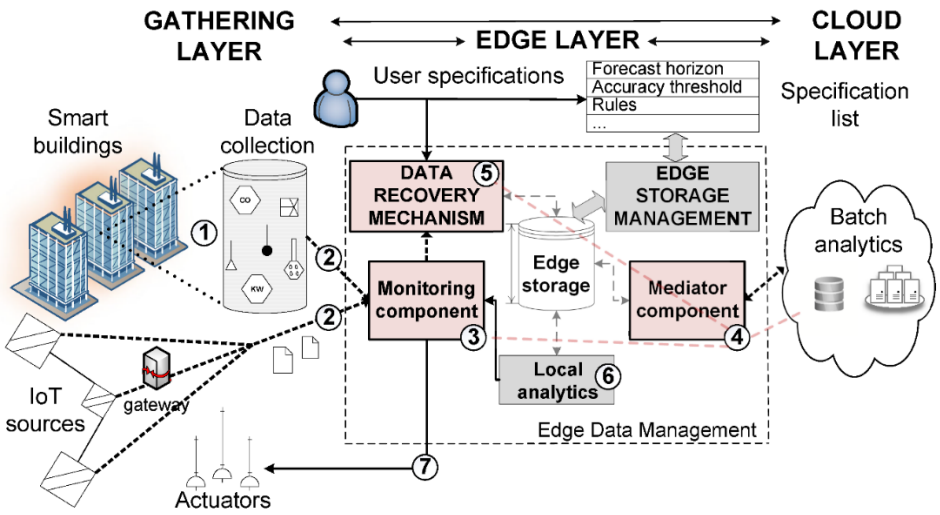

Figure 1. Edge analytics model for smart buildings use case.

ASSIGNMENT

Develop an edge-cloud mechanism for automatic and accurate recovery of incomplete sensor datasets based on detected gap length, such that data recovery mechanism (step 5) automatically applies recommended forecasting method on recomended range of historical data points that precede the gap. One of the solutions can be that mediator component (step 4) keeps a "recommendation information" for each dataset about (i) forecast method selection and (ii) which amount of historical data will best suit the recovery of certain gap length. This can experimentally be achieved by (i) artificially defining gaps within complete datasets, (ii) finding out which of used methods makes more accurate forecast (less error) on multiple gap lengths, (iii) at the same time observing which range of historic data points is most appropriate, (iv) making a conclusion, as a potential recommendation for the automatic recovery mechanism, i.e., which appropriate forecasting method and range of historical data can be used for the recovery process.

Additional information

- A **gap** is a sequence of one or more missing or invalid consecutive values, irregularly distributed in sensor-based time series data.
- During the observation/experimentation phase, we assume that collected datasets are already complete and without outliers, so we can evaluate forecasts afterwards.
- **Latency** as a time delay between edge node and cloud data center is neglected. We assume that all historical data are transferred and available in the cloud data repository.
- Near real-time analytics (step 6 in the figure) is application dependent, but making complete datasets before any kind of analysis is necessary for all edge deployed systems.
- In this assignment, an edge/cloud mechanism for automatic recovery of incomplete datasets must be developed by considering:
    - Different patterns of time series data (different datasets);
    - Various forecast methods/models that can be applied;
    - Different length of gaps (representing forecast horizon in the experimentation phase);
    - An accuracy measure to evaluate recovered gaps.
- Following settings and hints can be considered:
  1) Workload. We assume that:

- An edge node collects univariate time series data, that is, involving observations and analysis of only one variable (time stamps + values);

2) Time series forecasting. Features:
   - An experimental approach to time series prediction evaluation is consisted of dividing datasets into two parts, namely, a training and a test set (Figure 2). A training set, known as In-Sample data, is used to build up a model, and a test set, known as Out-of-Sample data, is used to validate the model built. The test data represent the artificially defined gap such that the amount of data points consisted in the gap gives needed forecast horizon, that is, the parameter value included in the forecasting process. **A training set usually contain more data points than a test set.**

Figure 2. Evaluation of forecast accuracy.

   - The setting of forecast methods typically requires specification such as: a training and a test set, periodicity (if seasonality exists), forecast horizon (number of data points that will be forecasted), and other parameters depending on selected method. If a dataset contains periodicity, then selected forecasting method requires specification of that periodicity before making forecast. If seasonality does not exist, periodicity is 1. There are many ways to check it (this is optional), and one of them is described in [6].
   - Use the length of forecast horizon as the size for your test set (gap length).
   - Generally, there are two different types of time series that can be also observed from the time series classification in Figure 3. Stationary datasets contain stationary characteristics where properties such as the mean or the variance do not change over time (for example, white noise). Nonstationary datasets contain nonstationary characteristics where mentioned properties change over time (for example including behaviors such as: random walk, cycle, upward and downward trends).

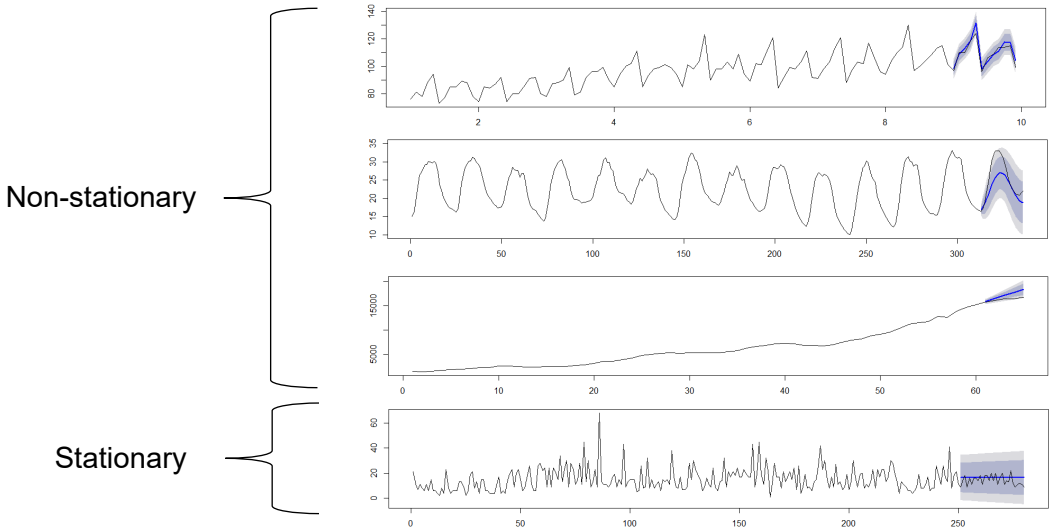

Figure 3. Datasets pattern classification with applied forecast technique.

3) Forecasts accuracy:
   - Forecast evaluation can be done by using different accuracy measures such as Mean Absolute Scaled Error (MASE) and Mean Absolute Percentage Error (MAPE), etc. There are several accuracy measures available, and we should choose only one of them. (They are usually already implemented in tools/libraries).
4) The experiments can be done using tool(s) and language(s) such as R [5][7], Matlab, Python (pandas), Weka, in which many forecasting methods are already built in.

DATES AND TASKS:

*Due **March 20nd** (23:59) Group assignment*

- ✓ Send an email with the subject: "EEDS Group Assignment" to ivan@ec.tuwien.ac.at (do not forget *CC:* ivona@ec.tuwien.ac.at) with **three** names building a group.

*Due **May 8th**: Presentation: Preparation; Implementation plan*

- ✓ Find two datasets (time series): *preferably from IoT systems/sensors; *if possible, both datasets should contain different characteristics (e.g., they can have seasonality, deterministic trend, stationary properties). ([1-3] might be a good starting point for checking.); *show charts of selected datasets including related information such as dataset sources, metrics, range of values, size, number of data points, characteristics (volatility, mean) etc.
- ✓ Select two forecasting methods/models you plan to use in your mechanism for adaptive recovery ([4-5] might be a good starting point for checking.); *Briefly describe them.
- ✓ Select one forecast accuracy measure that you will use for the forecast evaluation; *Briefly describe this measure – no need for formula explanation.
- ✓ Discuss and justify your implementation choices for programming language/framework, methodologies, libraries, packages, etc.
- ✓ Prepare brief descriptions and justifications for all your selections with appropriate arguments;
- ✓ Send slides of given presentation with the subject: "EEDS Assignment P1".

*Due **May 29th**: Presentation: Simulation infrastructure and preliminary implementation*

- ✓ Present your simulation infrastructure; Make a flow chart (or similar) diagram that describes your approach to the problem;
- ✓ Present how do you intend to perform needed observations (for example, how and where to observe forecast accuracy, how to decide which range of historical data points is needed). Justify your approach to iterative process of calculating ranges of data that give best forecast accuracy for used test data. *We do not expect more than 100 missing values in a gap.
- ✓ Select one data set and apply single forecast method on a gap length by choice, to find range of historical data that results generally in best forecast accuracy, according to your approach. *Show graph that describes behavior of forecast accuracy for different used historical data (preceding the gap) and showing selected/calculated amount necessary for chosen gap length. Illustrate/show corresponding steps/approaches/code etc. (*In order to apply selected forecasting method, maybe you will need to measure some characteristics, e.g., seasonality over data points before the gap. ([6-7] can provide techniques/methods for needed analysis). Briefly describe them if you used any.
- ✓ Send slides of given presentation with the subject: "EEDS Assignment P2".

*Due **June 19th**: Presentation: Results of the simulation, validation and discussion*

- ✓ Present simulation results as "recommendation information/chart" for both datasets from performed experiments (sequence of gap lengths, two forecast methods, forecast accuracies, potential ranges of necessary data points for gaps recovery).
- ✓ Present an experimental evaluation to see how recommendation is reliable: define three different gaps in one of datasets, (1) apply single method for all three gaps without specifying amount of historical data; (2) perform recovery based on recommendation information with specifying both forecast method and amount of used historical data. Compare total running time and average forecast accuracy between two approaches to check how good is your developed recommendation for automatic selection of appropriate forecasting method and range of historical data used for the recovery process.
- ✓ Send slides of given presentation with the subject: "EEDS Assignment P3".
- ✓ Send final report (.pdf) with description of the work done (integrating materials from tasks/presentations), results and conclusion. (in the next 2-3 days)

REFERENCES:

[1] "Umass trace repository," http://traces.cs.umass.edu/index.php/Main/Traces/
[2] "Comp-Engine time series," http://www.comp-engine.org/timeseries/
[3] "The R Datasets package," https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html
[4] N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn, "Self-adaptive workload classification and forecasting for proactive resource provisioning," in Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, ICPE '13. ACM, 2013, pp. 187–198.
[5] R. Hyndman, "Forecasting time series using R", http://robjhyndman.com/talks/MelbourneRUG.pdf
[6] "Measuring time series characteristics", http://robjhyndman.com/hyndsight/tscharacteristics/
[7] "Time series analysis", https://cran.r-project.org/web/views/TimeSeries.html