EDGE APPROACH FOR EFFICIENT DECISION-MAKING PROCESSES

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

BACKGROUND AND MOTIVATIONAL SCENARIO

Massive amounts of data are constantly generated from the growing number of Internet of Things (IoT) devices. Nowadays, sensor-based data are used in many applications, like eHealth, smart manufacturing, intelligent traffic management, smart building systems. Such systems rely on monitoring of parameters (e.g., temperature, movement, heart rate, energy consumption, power production, radiation, pressure or air quality) coming from many sensors. Managing these systems generally requires (i) sensor data collection, (ii) data processing and (iii) acting based on the obtained results. Traditionally, such processing is done by employing geographically distributed and massive data centers. Recently, edge computing has been proposed as a solution to perform near-real-time decisions in the proximity of the user. Edge analytics rely on edge nodes (e.g., Raspberry Pi, edge gateway or micro data center) that can be deployed closer to IoT sensors and devices. By keeping data analytics close to the source of data, edge nodes can minimize the latency for decision-making processes as well as to improve energy efficiency of data transmission networks[1]. Furthermore, predictive analytics have a huge potential to revolutionize critical and proactive IoT systems such as accurate diagnosis of patients in eHealth, timely maintenance services and failures prevention in smart manufacturing and building systems.

However, meeting the strict latency and accuracy requirements of decision-making processes while performing predictive analytics in modern edge/IoT systems, imposes several issues. Limited (i) storage capabilities and (ii) scalability of edge nodes can hinder the accuracy of predictive analytics and timely decision-making for IoT systems. Also, users and IoT applications can require various prediction horizons (windows) as well as different accuracy thresholds.

We consider data generated by sensor-based monitoring, classified as **time-series** data. Based on a certain amount of historical time series and recognized patterns, it is possible to predict future values. The conclusions based on analyzed data generally assume that the future system behavior will follow a behavior (pattern or trend) from the measured past. Accordingly, we can use predictive analytics in the context of proactive decision-making.

Figure 1 represents an edge analytics model for smart building use cases. In step (1), data are collected from smart buildings. Due to the rapidly increasing amount of data generated, data can be transferred to the edge layer in step (2). Once a certain amount of data is transferred to the edge layer, the monitoring service cleans data and stores them to the limited edge storage (step 3). The available data are used in step (4) for near-real-time decision-making processes based on the results of predictive analytics, while checking the knowledge base that can be updated with dynamic user/application requirements. The knowledge base should contain recommendations on the best trade-offs between the quantity of the data and prediction accuracy as well as the most suitable forecasting methods for a particular dataset type regarding the prediction horizon. The analytics results are used to take operative decisions, e.g., commands sent to actuators in step (5). The mediator supports the edge layer and communicates with the cloud data repository for resource-demanding batch analytics (step 6).

[1] https://www.iea.org/reports/data-centres-and-data-transmission-networks

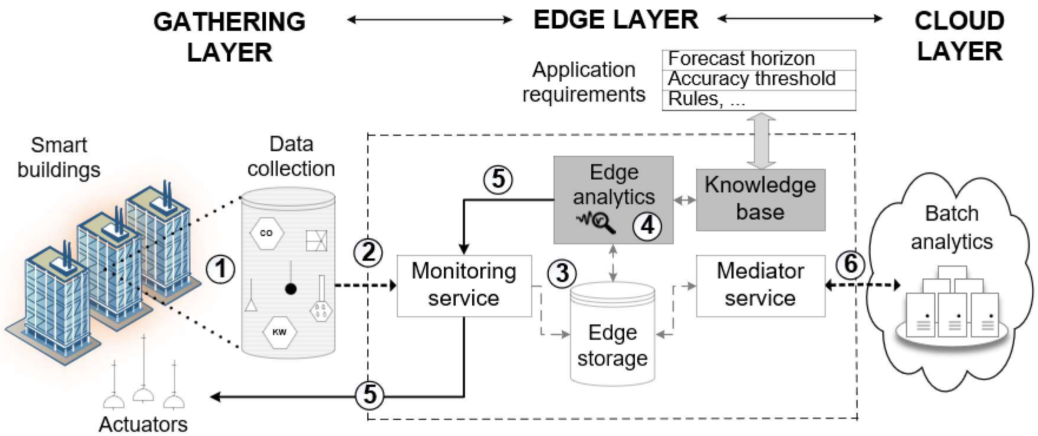

Figure 1. Edge analytics model for smart building use case.

<u>ASSIGNMENT</u>

Analyze and determine the best settings in the edge approach for efficient decision-making processes. The resulting recommended settings should ensure that the edge analytics component (step 4) can select an appropriate forecasting method having both high forecast accuracy and low latency by considering (i) target dataset type, (ii) required prediction horizon and (iii) the most suitable range of historical data as input. We consider the recommendations to be integrated into the knowledge base, in the future edge design.

"Which range of data and corresponding prediction horizon can ensure the best trade-off between prediction accuracy and computation latency", can experimentally be solved by **(i)** artificially defining different prediction horizons and ranges of training data using complete datasets, **(ii)** applying different forecasting methods for each of defined prediction horizons and training sets, **(iii)** observing forecast accuracies and latencies (running time), and **(iv)** consequently, making recommendations based on conditions and trade-offs that enable efficient decision-making, i.e., which forecasting method and setting can produce the best accuracy-latency trade-off regarding prediction horizon and given dataset type.

<u>Guidelines and details</u>

- The assignment is based on **time-series data**.
- A **prediction horizon** is the number of data points representing prediction length, that is, the amount of data into the future for which the forecasts are to be calculated.
- During the observation/experimentation phase, we assume that datasets are already complete (without missing values) and without outliers, so the forecasts can be evaluated.
- For the sake of simplicity, the **latency** is considered as a time consumption or running time of the forecasting procedure.
- **Edge predictive analytics** (step 4) for decision-making are often application dependent. However, our recommendations can be related to specific datasets and observations.
- Edge recommended settings for efficient decision-making must be determined considering:
    - different patterns of time-series data (different sensor-based datasets);
    - various forecasting methods/models that can be applied;
    - different length of forecast horizons in the experimentation phase;
    - a forecast accuracy measure to evaluate forecasted ranges of data points.
- Following settings can be considered:

1) <u>Workload.</u> We assume that:

- An edge node collects *univariate* time series data, that is, observations including only time stamps and measured values.

2) <u>Time-series forecasting.</u> 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 prediction horizons included as the parameter value in the forecasting process. A training set should contain more data than a test set.

Figure 2. Evaluation of the forecast accuracy.

- The setting of forecast methods typically requires specification such as training and test set, periodicity (if seasonality exists), prediction horizon (number of data points that will be forecasted), and other parameters depending on the selected method. If a dataset contains periodicity, then the selected forecasting method requires specification of that periodicity before making a forecast. If seasonality does not exist, periodicity is 1. There are multiple solutions to check this (optional), one of them is described in [8].
- **Consider lengths of the prediction horizons as different sizes of the test set.**
- Generally, there are two different types of time-series that can be seen from the time-series classification in Figure 3, namely, stationary data contain stationary characteristics where properties such as the mean or the variance do not change over time (e.g., white noise). Non-stationary data contain non-stationary characteristics where properties can change over time (e.g., including random walk, cycles, upward and downward trends).

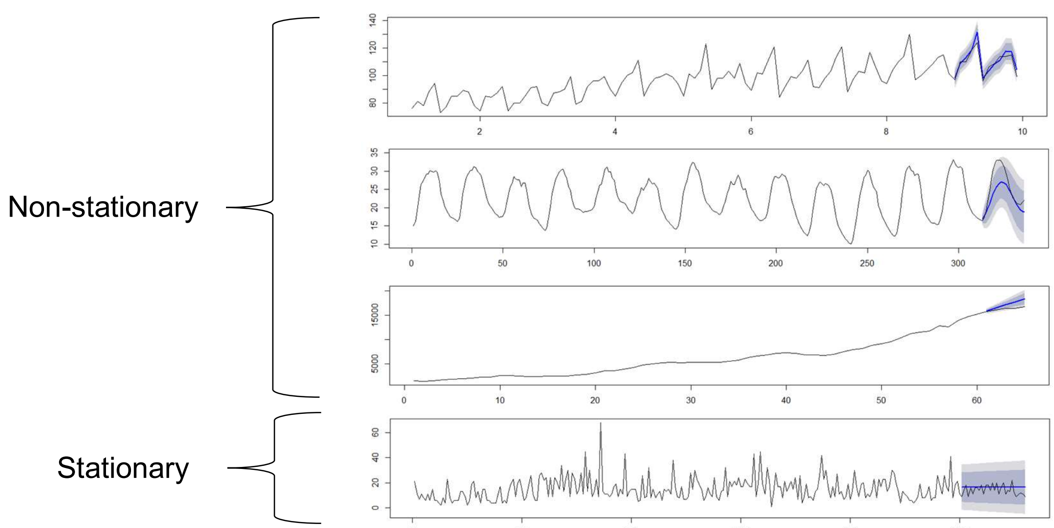

Figure 3. Dataset pattern classification with applied forecast method.

3) <u>Forecast accuracy</u>:

- Forecast evaluation can be done by using different accuracy measures such as Mean Absolute Scaled Error (MASE) and Mean Absolute Percentage Error (MAPE). There are several accuracy measures available, and you should select only one of them. (They are usually already implemented in the existing libraries and packages).

4) The experiments can be done using different tools, languages and libraries such as R (recommended, if you never worked with time-series before) [7][9], Python (pandas, numpy, prophet, sklearn, etc.), Keras, Weka, Matlab, Java, in which forecasting techniques/methods/models/(accuracy measures) are already implemented.

DATES AND TASKS:

---

*Due **March 23rd** (23:59): Building a Group for the Assignment*

✓ This is a project assignment of **max three** students building a group. You can add/register yourself to an available group number in the choice activity module on TUWEL course page. (you can update your choice before the activity module is closed)

---

*Due **April 20th**: Presentation: Preparation and Implementation Plan (5-10min)*

✓ Find two datasets (time-series): *Preferably from IoT systems/sensors; *Datasets should contain different characteristics (e.g., they can have seasonality, deterministic trend, or stationary properties). [1-5] might be a good starting point for checking; *Show graphs of selected datasets including characteristics such as dataset sources, metrics, range of values, size, number of data points, characteristics (volatility, mean), etc.

✓ Discuss and justify your implementation choices for programming language/framework, methodologies, libraries, packages, etc.

✓ Select and briefly describe two forecasting methods/models that you plan to use in your mechanism for the edge analytics.

✓ Select and describe one forecast accuracy measure that will be used for the evaluation.

✓ *The idea behind the first presentation is to: find potential datasets for experiments; select and get familiar with potential forecasting methods; get familiar with coding environment by importing datasets, plotting them, and extracting some simple characteristics; detect potential libraries and packages.*

✓ Send slides of given presentation with the subject: “EEDS Assignment P1”, to ivan@ec.tuwien.ac.at (Cc: ivona@ec.tuwien.ac.at).

---

*Due **May 18th**: Presentation: Simulation Workflow and Preliminary Implementation (5-10min)*

✓ Present your simulation workflow (you can also make a flowchart), i.e., describe how do you intend to: (i) define multiple prediction horizons; (ii) define different sizes of the training set; (iii) perform needed observations (i.e., in which settings prediction horizons can benefit using different ranges of available training data, based on observed accuracy and latency), etc. *We consider limited edge storage that cannot store all historical data, so in the case of big datasets found in the previous presentation, you are free to take a subset of such big dataset. *Also, we do not expect the prediction horizon bigger than available training data.

✓ Select one dataset and single prediction horizon (by choice), apply both forecasting methods, and show which method works better (forecast accuracy, running time). Then, for the same prediction horizon, check the performance using reduced size (by choice) of the training set. *Show graphs. *Show corresponding steps, code, etc. *In some cases, to apply a forecasting method, you will need to measure characteristics such as seasonality of data. [8-9] could provide techniques/methods for this analysis. Briefly describe them if you use any.

✓ Send slides of given presentation with the subject: “EEDS Assignment P2”, to ivan@ec.tuwien.ac.at (Cc: ivona@ec.tuwien.ac.at).

---

*Due **June 8th**: Presentation: Results of the Simulation, Validation and Discussion (5-10min)*

✓ Present simulation results as “recommendation settings” for both selected datasets, considering **(i)** performances of both forecasting methods, **(ii)** multiple defined prediction horizons (it is up to you how you define them, e.g., it can be predefined range of short, medium, and long horizons), **(iii)** forecast accuracies and running times, **(iv)** checking if different (reduced) sizes of the training set can achieve higher forecast accuracy and/or keeping the same (or lower) running time, for selected prediction horizons in both datasets. *Show important plots, performance results (accuracies, running times), comparisons.

✓ *The idea behind the third presentation is to show conclusions (or trade-offs) considering running times and forecast accuracies about potential recommended selection of appropriate forecasting methods and “best settings” of prediction horizons / training data that can be used for proactive edge near-real-time decisions.*

✓ Send slides of given presentation with the subject: “EEDS Assignment P3”, to ivan@ec.tuwien.ac.at (Cc: ivona@ec.tuwien.ac.at).

✓ Prepare and send the final report (*.pdf* up to 10 pages) with description and justification of the work done (integrating/summarizing materials from tasks/presentations), results and conclusion. (This report should be sent in the next 3-4 days after this final presentation)

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] “Kaggle datasets,” https://www.kaggle.com/datasets

[5] “UCI Machine Learning Repository,” https://archive.ics.uci.edu/ml/datasets.php

[6] 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.

[7] R. Hyndman, “Forecasting time series using R”, http://robjhyndman.com/talks/MelbourneRUG.pdf

[8] “Measuring time series characteristics”, http://robjhyndman.com/hyndsight/tscharacteristics/

[9] “Time series analysis”, https://cran.r-project.org/web/views/TimeSeries.html