BigClue Analytics: Modeling Sensor Data and Anomaly Detection in IoT Systems

Dan Huru, Cătălin Leordeanu, Elena Apostol,
Mariana Mocanu, and Valentin Cristea

Faculty of Automatic Control and Computers,
University Politehnica of Bucharest, Romania
Email:alexandru.huru2208@cti.pub.ro, {catalin.leordeanu, elena.apostol,
mariana.mocanu, valentin.cristea}@cs.pub.ro

Abstract. The typology of most IoT projects requires data processing to be handled in a low latency scenario. The fact that we are dealing with a large number of devices, each producing considerable amounts of data, contributes to the difficulty of this subject. For this purpose we introduce BigClue, a middle-ware capable of collecting processing and modeling sensor data. The architecture contains a specialized component which is part of the middle-ware that has multiple features: data modeling, data prediction, anomaly detection and sensor correlation. To validate our work we used a scenario involving a greenhouse environment but the architecture can be adapted to other problem domains.

Keywords: IoT, machine learning, time series modeling, analytics, anomaly detection

1 Introduction

There is a considerable number of IoT use cases where real-time information must be processed and presented in a meaningful way. An underlying platform capable of handling these use cases must overcome several technical challenges, in very different scientific fields. Sensors can contribute with data to enable smart farming environments. For example information about humidity, temperature or light can be used in conjunction with weather data to understand and better monitor farm crops. Sensor information can also be used to understand the development of plants and verify their behavior with existing domain knowledge. In addition location data and statistical information can provide a better context (e.g. soil type, wind, air composition, animals). By gathering information from several farms, different strategies can be shared among farms while the farming process becomes more solid. Legal and compliance information services can be used by both farmers and agencies to manage agriculture better.

Houses and offices can make use of distributed sensors to automatically handle room heating, lighting and save energy with appropriate monitoring and alert systems. For example in [1] an energy-aware/cost ware platform for smart homes is implemented to monitor energy consumption. It also benefits real-time

information about tariffs by integrating smart grid web services. Another example is EnergyVisualizer [2], which offers a web interface to control different home appliances and their energy consumption. modeling sensor data can be useful in multiple ways. One approach is to model data in order to detect anomalies. Second, if the model is stable future values can be predicted.

The value added by data modeling systems in this scenario lies in the speed and accuracy with which the model is calculated. There is little use for an anomaly that is detected too late or if the predicted window is too short.

This paper reveals typical scenarios and introduce our middle-ware solution, in order to better understand the context of our research. BigClue is an end-to-end solution for data management suitable for IoT scenarios. The main contribution of this paper is to present BigClue Analytics, which is a middleware capable of data modeling and reasoning. It also demonstrates its capabilities in a smart farming context, specifically a greenhouse environment,

The paper is organized as follows. Section 2 reviews existing work while Section 3 presents an overview of the BigClue middle-ware with its analytics component and its features. Section 4 presents in detail the use case and the experimental results. In Section 5 we outline our work and propose future improvements.

2 Related work

2.1 Similar middle-ware

There is a considerable number of middle-ware providing end-to-end data management solutions. However, the majority do not support real-time processing and advanced reasoning. With the exception of MoCA[6] which provides an extra semantic context, most of the solutions which do provide real-time capabilities present limited reasoning techniques such as the use of rules. Such a solution is SCONSTREAM [7] which has a centralized architecture that processes streams of data coming from physical sensors and applies rules to detect events. It highlights the challenges that appear in an IoT environment from a real-time perspective.

MoCA[6] is a service based distributed middle-ware that employs ontologies to model and manage context. It consists of three key components: context providers, context consumers and context service. The providers are responsible for generating or retrieving context from other sources available to be used by the context management system while consumers consume the context gathered and processed by the system. The context service receives, stores and disseminates context information.

In addition to reasoning about anomalous events in real-time BigClue analytics uses techniques like data approximation, data sampling and parallel processing to offer an optimized solution.

2.2 Use case: smart farming

The subject of time series data modeling is widespread and heavily researched. However, its application in the context of smart farming has received less attention. With respect to anomaly detection in [3] the authors model a greenhouse environment and propose anomaly detection based on statistical functions. [4] offer content anomaly detection based on Gaussian predictor and contextual anomaly detection using sensor profiles. Other anomaly detection methods are proposed in [5] using a Gaussian mixture model.

Other types of reasoning in this context include rainfall prediction in precision agriculture using neural networks [8], weather forecasting for future farming [10] or root cause analysis for wine grape quality differences [9].

3 Solution overview

BigClue middle-ware is based on the Lambda architecture [12] and is described in Fig. 1

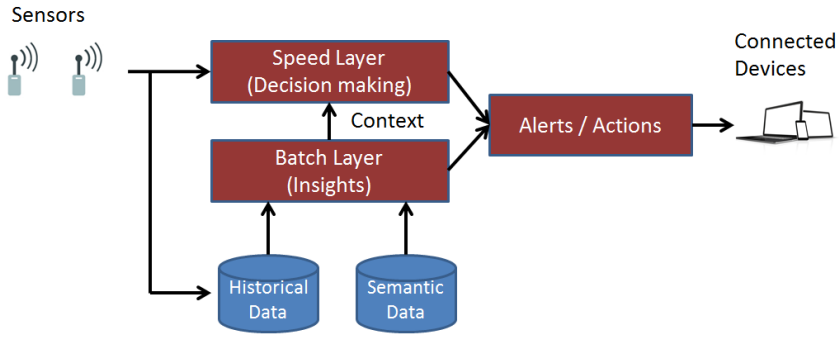

Fig. 1: BigClue architecture

The first step is to collect sensor data and route it to a central endpoint. Then this data is processed two times: once in the speed layer where fast decision making is required, second in the batch layer where a time series algorithm models the historical pattern and makes prediction. The results are used in the speed layer to raise alarms and recommendations to the user.

The main components of BigClue are:

- Message Queue - data is collected from multiple sources and replicated to multiple destinations
- Procesing engine - processing can take various forms (cleaning/rules/analytics)
- Storage engine - data needs to be stored in a persistent layer for later availability
- Visualization - data must be presented in useful way

3.1 BigClue Analytics

The BigClue Analytics component is part of the processing layer of BigClue. A high level architecture is shown in Fig. 2

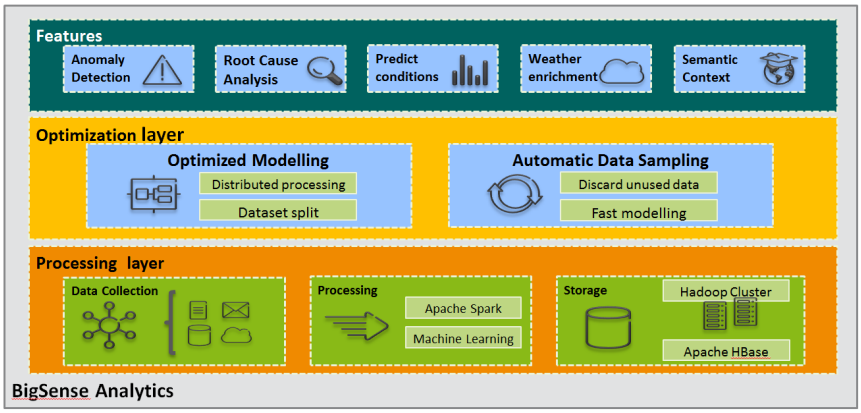

Fig. 2: BigClue Analytics architecture

It offers several features: Anomaly Detection, Prediction of future conditions, Weather data enrichment, Root cause analysis of events, Semantic Context. In addition to these features, BigClue Analytics benefits from using several optimization techniques: Optimized modeling, Automatic data sampling.The rest of this section contains details regarding each feature.

3.2 Anomaly detection

When monitoring an IoT system it is useful to: Identify measurement errors (faulty sensors), Detect unusual events (an door or a window accidentally left open) or to Predict future conditions.

Time series data may contain two types of irregularities: continuous multiple point events and single measure outliers. It is useful to detect both. The terminology we've used in our research maps outliers to measurement errors and anomalies to unusual events.

The anomaly detection service detects anomalous events in time series data such as temperature and humidity collected from sensors placed inside a greenhouse environment. The objective is to signal these issues to the users and offer feedback in the form of suggestions or automated actions. The type of detected anomalous events are split in two categories:Single-point outliers and Multi-point anomalous events.

3.3 Root cause analysis using sensor data correlation

Administrators are usually interested in quickly understanding the nature of the alerts. A proactive approach is to raise an alert and provide a possible root cause for the problem, thus assisting the administrator with the investigation. In this sense, alerts should be categorized in: faulty sensors, measurement errors, open windows, unusual weather events and so on. Also the system should be capable of filtering false or minor incidents that do not require intervention.

BigClue Analytics models multiple series produced by sensors, as represented in Fig. 3.

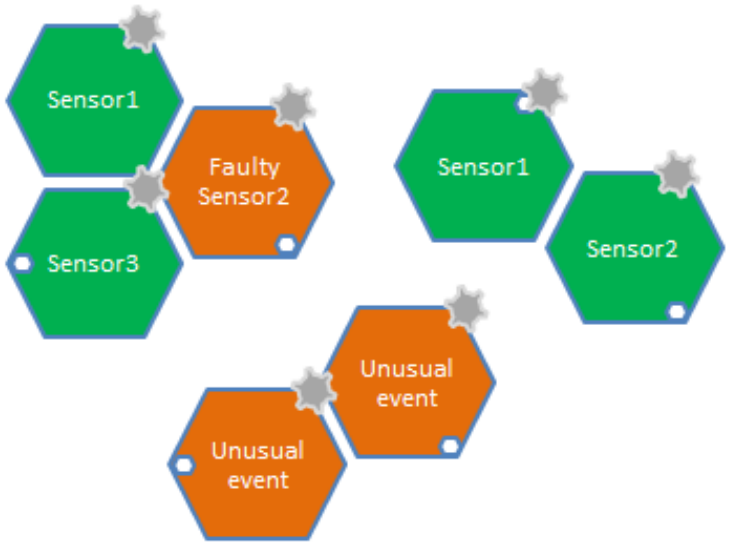

Fig. 3: Correlations between collocated sensors

When anomalous data is detected for one sensor, it is crosschecked with rest of the group of collocated sensors. If there is a matching anomalous event, then the system raises an alarm stating that an unusual external event has taken place (e.g. open door/broken window). Otherwise the system marks the sensor as faulty. This strategy is described in Algorithm 1.

4 Experimental results

4.1 Experimental setup

Our research started with ClueFarm [11], a typical use case of smart agriculture which aims to aggregate, process, analyze and present farm related information.

Algorithm 1 RCA logic with collocated sensors

```
while sensor1.getEvents.stream() do
  if (event is anomalous) then
    collocatedEvents = sensor1.getCollocatedEvents()
    if (collocatedEvents are anomalous) then
      PRINT An unusual event has occurred!
    else
      PRINT Measuring error!
      increment sensor1.errorCounter
      if (sensor1.errorCounter gt globalTreshold) then
        PRINT Sensor is faulty!
      end if
    end if
  end if
end while
```

One of the objectives of the project is to offer a monitoring solution of farm resources and the environmental conditions of greenhouses.

In this work we are considering two sources of time series: internal temperature and humidity, collected from greenhouse sensors. The data is collected for a 2 months' period (beginning of September till end of October) from a local greenhouse. The frequency of collected values is 15 minutes.The data set records contain:

$$sensorID|timestamp|temp|hum|externalmeasures$$

The sensors are collocated and they present high data correlation with external weather conditions. In both Fig. 4 and Fig. 5 a day/night shift pattern can be observed. A trend-line on the temperature values shows linear decrease over time in alignment with outside conditions in Fig. 6 since the measured interval starts from 4th of September and ends on 21st of October.

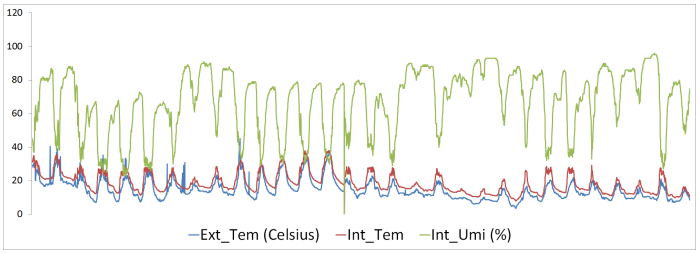

Fig. 4: 2 months of sensor data

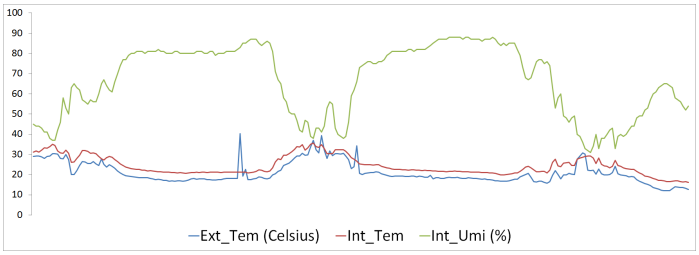

Fig. 5: 4 days of sensor data

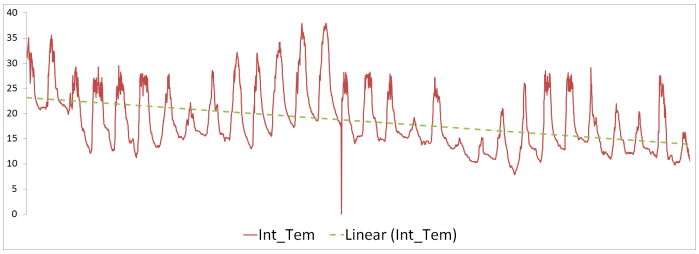

Fig. 6: Decreasing temperature trend

4.2 Algorithm choice

Using statistical functions A first attempt to solve the anomaly detection problem is to rely on basic statistical functions in Fig. 2. These functions are applied on a window of timestamps. Using mean and standard deviation we calculate vectors of min and max values. Values outside the limiting vectors are considered outliers (an arbitrary *stdFactor* is manually chosen). The results are shown in Fig. 7.

Algorithm 2 Outlier detection using statistical functions

```
Compute mean vector
Compute variance vector
STD ← sqrt(variance)
minValues ← (std − mean)
maxValues ← (std + mean)
if newValue between [MinValues, MaxValues] then
   NOK
end if
return  OK
```

This approach identifies several outliers but has considerable drawbacks:

– Misses subtle outliers

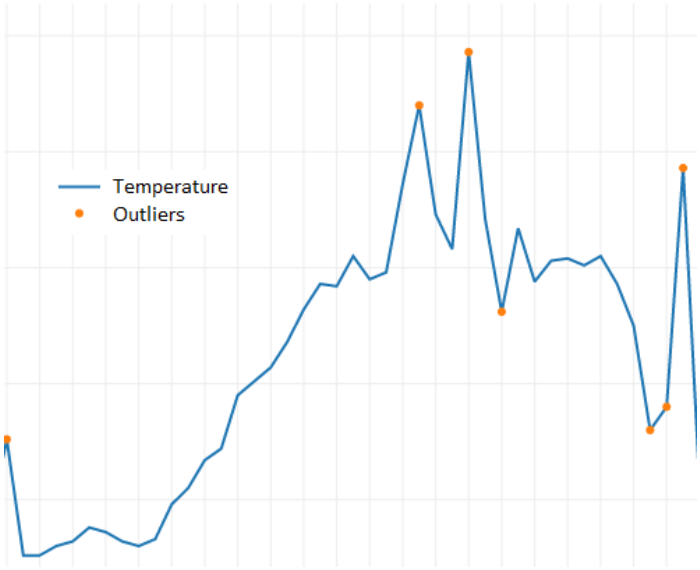

Fig. 7: Outliers in temperature data

- Doesn't detect multiple outliers in a row - If there is a continous stream of outliers, it will affect the window mean and variance
- Manually adjusting a parameter - depending on the value of stdFactor may cause too many false pozitives or true negatives
- No prediction is done

Linear regression For baseline purposes we modelled the data-set using basic linear regression. The data-set contains 700 measures from which first 500 measures are used as training set (in red) and the rest of 200 as the test set (in blue) in Fig. 8. We measured RMSE = 5.694187.

Signal decomposition Decomposition breaks the signal into seasonal, trend and remainder components as in figure 8. If the trend can be successfully modelled, once the signal is recomposed to its original form prediction can be done accurately.

In our case we've used both the additive and multiplicative models to decompose the signal and polynomial regression to train the model. Such an approach reasonably fit the training set as shown in Fig. 9 but failed to make accurate predictions on the test set.

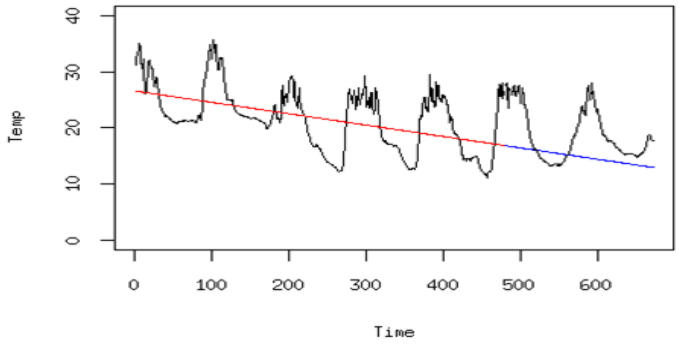

Fig. 8: Prediction with linear regression

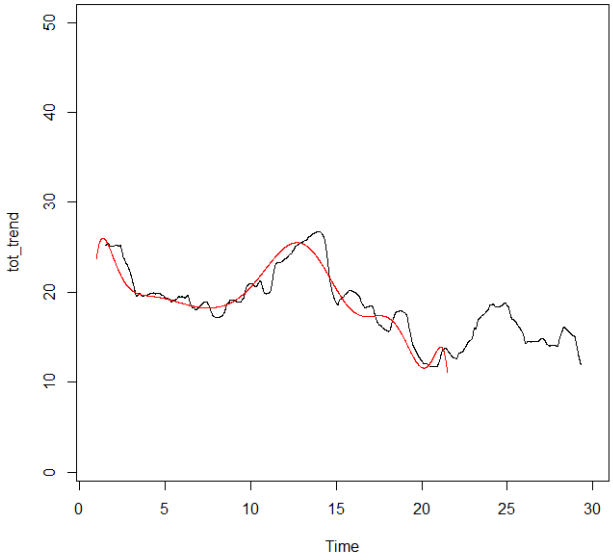

Fig. 9: Fitting with polynomial degree of 10

This is due to the sinusoidal nature of the trend which is not a good fit for a linear model. Even if the test set could be accurately predicted under some form, the polynomial degree still has to be chosen manually and this choice depends on the particularities of the dataset.

ARIMA ARIMA [13] is a popular time series analysis technique in many areas and is composed of the following:

- Auto-regressive process: Each observation is composed of a random error component(random shock, ϵ) and a linear combination of prior observations.
- Stationarity requirement: The parameters of the equation must be within a certain range
- Moving average process: Each observation is made up of a random error component (random shock, ϵ) and a linear combination of prior random shocks.
- Invertibility requirement: The moving average equation can be rewritten (inverted) into an autoregressive form (of infinite order) with the condition that the model is invertible.

ARIMA produced a good model and future estimates shown in Fig. 10. Evaluations on our dataset resulted in RMSE = 2.51002 for a 3-day prediction interval.

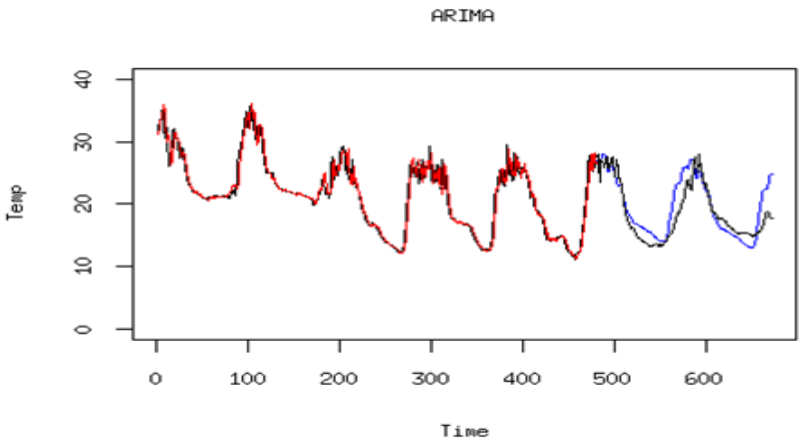

Fig. 10: ARIMA modeling

5 Conclusions and future work

This paper underlines several key points. First we presented the context of the paper and presented an overall picture of BigClue, a platform for data processing in IoT systems. Then we showcased BigClue Analytics, an analytics component that offers multiple features: anomaly detection, root cause analysis and weather data enrichment. We also described how the system uses different mechanisms to optimize computation time.

The anomaly detection mechanism achieved most accurate predictions using the ARIMA algorithm. Further optimizing the model computation we concluded that approximation techniques can be used to significantly reduce time. In this sense we showed that a sampling rate of 32 delivered acceptable prediction accuracy.

Future work will include the integration of a plant ontology to infer semantic value and enable refined insights. The ontology would be further developed by sharing it across multiple greenhouses.

A second point will involve black box testing. In this sense data from multiple greenhouses will be gathered in order to compare the results. Another step forward is to consolidate the results across multiple domains where different patterns of time series data are available.

Acknowledgment

The work has been supported by the project Data4Water: Excellence in Smart Data and Services for Supporting Water Management, number 690900/H2020-TWINN-2015, and the UPB project GEX AU 11-17-11 Activ. 4000.130: .

References

1. Kamilaris, A. and Pitsillides, A., 2011, May. Exploiting demand response in web-based energy-aware smart homes. In Proceedings of the First International Conference on Smart Grids, Green Communications and IT Energy-Aware Technologies (Energy 2011).
2. Guinard, D. and Trifa, V., 2009, April. Towards the web of things: Web mashups for embedded devices. In Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain (Vol. 15).
3. Eredics, P. and Dobrowiecki, T.P., 2012. Data Cleaning and Anomaly Detection for an Intelligent Greenhouse. In Applied Computational Intelligence in Engineering and Information Technology (pp. 123-134). Springer Berlin Heidelberg.
4. Hayes, M.A. and Capretz, M.A., 2014, June. Contextual anomaly detection in big sensor data. In Big Data (BigData Congress), 2014 IEEE International Congress on (pp. 64-71). IEEE.
5. Pang, H., Deng, L., Wang, L. and Fei, M., 2016, October. The Application of Spark-Based Gaussian Mixture Model for Farm Environmental Data Analysis. In Asian Simulation Conference (pp. 164-173). Springer Singapore.
6. da Rocha, R.C.A. and Endler, M., 2006. Middleware: Context management in heterogeneous, evolving ubiquitous environments. IEEE Distributed Systems Online, 7(4), pp.1-1.
7. Kwon, O., Song, Y.S., Kim, J.H. and Li, K.J., 2010, March. Sconstream: A spatial context stream processing system. In Computational Science and Its Applications (ICCSA), 2010 International Conference on (pp. 165-170). IEEE.
8. Bendre, M.R., Thool, R.C. and Thool, V.R., 2016. Big Data in Precision Agriculture Through ICT: Rainfall Prediction Using Neural Network Approach. In Proceedings of the International Congress on Information and Communication Technology (pp. 165-175). Springer Singapore.

9. Krintz, C., Wolski, R., Golubovic, N., Lampel, B., Kulkarni, V., Roberts, B. and Liu, B., 2016. SmartFarm: Improving agriculture sustainability using modern information technology. ACM SIGKDD DSFEW.
10. Bendre, M.R., Thool, R.C. and Thool, V.R., 2015, September. Big data in precision agriculture: Weather forecasting for future farming. In Next Generation Computing Technologies (NGCT), 2015 1st International Conference on (pp. 744-750). IEEE.
11. Serrouch, A., Mocanu, M. and Pop, F., 2015, June. Soil management services in cluefarm. In Parallel and Distributed Computing (ISPDC), 2015 14th International Symposium on (pp. 204-209). IEEE.
12. Hausenblas, M. and Bijnens, N., 2014. Lambda architecture. URL: http://lambda-architecture.net/. Luettu, 6, p.2015.
13. Ojo, I., 2010. Autoregressive integrated moving average. Asian Journal of Mathematics and Statistics, 3(4), pp.225-236.