Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 11 – 17

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

Event prediction in an IoT environment using naïve Bayesian models

Bill Karakostas*

†

VLTN GCV, Antwerp, 2000 Belgium

Abstract

In many Internet of Things (IoT) scenarios, there is a need to predict events generated by objects. However, because of the dynamicity of IoT environments, it is difficult to predict with certainty if/when such events will occur. Probabilistic reasoning allows us to infer dependent probabilities of events, from other events that are either easier to detect or to predict. In this paper we propose an architecture that employs a Bayesian event prediction model that uses historical event data generated by the IoT cloud to calculate the probability of future events. We demonstrate the architecture by implementing a prototype system to predict outbound flight delay events, based on inbound flight delays, based on historical data collected from aviation statistics databases.

Peer-review under responsibility of the Conference Program Chairs

Keywords: IoT, event prediction; probabilistic reasoning;Bayesian networks; Big Data

* Corresponding author. Tel.: 3278484019; fax: 3278484019.
E-mail address: bill.karakostas@vltn.be

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.093

1. Introduction

Internet of Things (IoT), refers to the concept of interconnecting uniquely identifiable embedded computing devices using the existing Internet infrastructure. IoT is expected to offer advanced connectivity of devices, systems, and services. The number of devices connected to the Internet of Things (IoT) will reach 26 billion devices by 2020, according to Gartner Group (www.**gartner**.com/newsroom/id/2636073).

Internet of Things objects generate many events in their lifetime, as a result of changes in their state and in the state of their environment. Several middleware architectures for managing such events have been proposed, for example, in[1], addressing both spatial and temporal characteristics of IoT events.

Advance prediction of important events gives more time to other objects and systems in the environment to respond in a proactive manner. For example, in logistics the ability to predict delays during one transport stage gives sufficient time to the next transport stage to reschedule its activities.

Event mining techniques are often applied to time series data in order to detect event patterns and to predict future events[6]. Our approach applies to discrete events that occur in the IoT 'cloud' and trigger further events. In many IoT environments event causality is not known, and thus probabilistic approaches must be used to predict the probability of subsequent events occurring, based on the observation of other events.

In this paper we use the Bayesian approach to probabilistic reasoning[7], in order to calculate the probability of triggered events occurring based on the probabilities of the occurrence of triggering events. The use of Bayesian networks to ascertain the quality of data transmitted by the sensors has been proposed by researchers, for example, in [5].

The structure of the paper is as follows: The next section surveys key concepts of the Bayesian system of probabilistic reasoning, with emphasis on event prediction. Section 3 proposes a Bayesian model for probabilistic event prediction and an architecture that applies the model to historical IoT event data, to calculate probabilities of future events. Section 4 presents a case study on flight delay event prediction that illustrates the method and application of the proposed prediction model. Finally, section 5 concludes with a discussion of the significance of the proposed approach in the overall context of IoT research.

2. Bayesian Networks

Bayesian networks are used to model a domain containing uncertainty[3]. A Bayesian network is a directed acyclic graph (DAG) where each node represents a discrete random variable of interest. The parent-child relationship between nodes in a Bayesian network indicates the direction of causality between the corresponding variables, i.e. the variable represented by the child node is causally dependent on the ones represented by its parents. Each node corresponds to the states of the random variable that it represents and the probabilities of the node being in a specific state given the states of its parents (conditional probability). In a Bayesian model of events, an edge between two event nodes shows a probabilistic dependency between the events, i.e. $e_1 \rightarrow e_2$ indicates that the probabilities of event e_2 occurring are conditional on the probabilities of event e_1 occurring.

In general, for events *A* and *B*, where A depends on B, provided that $P(B) \neq 0$,

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

In many situations event *B* is fixed (P(B)=1), and we wish to consider the impact of it having been observed, on the probability of other possible events *A*. In such a situation the denominator of the above expression, the probability of the given evidence *B*, is fixed and we want to calculate the probability of A based on how the evidence of B regulates our prior (historical) knowledge about the probability of A.

In an IoT context, we assume an underlying (but unknown) set of causal relationships between objects that determines how events generated by objects trigger subsequent events. However, because the causal relationships are usually unknown, we use statistical knowledge about co-occurrences of events to create a conditional probability event model. We use a probabilistic dependency between events e_1 and e_2 to express the knowledge derived from statistical observations that event e_2's probability of occurring depends on the probability of event e_1 occurring. Then, using the Bayesian probability formula, the conditional probability of e_2 occurring, given the probability of occurrence of e_1 is

$$P(e_2 \mid e_1) = \frac{p_h(e_2) * p(e_1 \mid e_2)}{p\ (e_1)} \quad (1)$$

Where $p_h(e_2)$ is the historical probability of observing e_2, independently from other events, and $p(e_1 \mid e_2)$ indicates the strength of the influence of e_1 evidence on the probability of e_2.
If we assume with certainty that e_1 has occurred, $p(e_1)=1$ and formula (1) is simplified to

$$P(e_2 \mid e_1) = p_h(e_2) * p(e_1 \mid e_2) \quad (2)$$

In IoT environments, certain events can cause a 'chain reaction' of events, therefore we also want to be able to calculate the dependent probability of an event e' on an event e through a chain of *n* intermediate events e_1, e_n,…
i.e. the dependency $e \rightarrow e_n \rightarrow e_{n-1} \rightarrow \ldots \rightarrow e_1 \rightarrow e'$, where e_n is observed ($p(e_n)=1$) and there are known dependent probabilities $p(e_k \mid e_{k-1})$.

By applying formula (1) to calculate sequentially the conditional probabilities of $p(e_n|e_{n-1})$, …, $p(e_2|e_1)$ we can calculate the conditional probability of event e' on event e through a chain of n events under the assumption that the probability of e_{k+1} is only dependent on the probability of e_k in the chain of events $e_1,\ldots,e_n$.

3 IoT Event Prediction Architecture

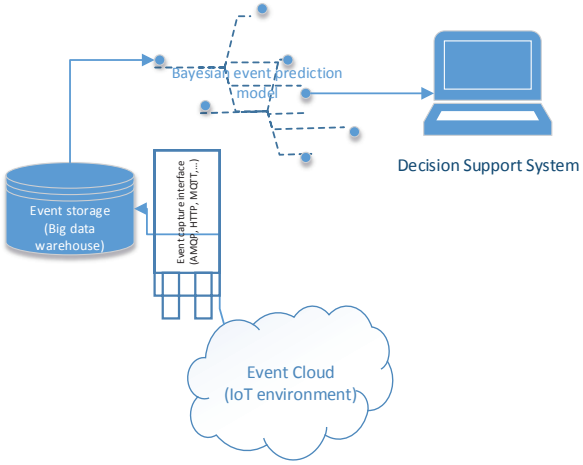

Figure 1: Generalised architecture for IoT event prediction.

Figure 1 shows the generalised architecture of an event prediction infrastructure for IoT. Events are generated by objects in the IoT cloud (network/s) and, through a capture interface that supports typical IoT M2M protocols, are stored in an event warehouse. Event data are then cleaned and classified according to the prediction (Bayesian and other probabilistic) models supported. Prior (historic) probabilities of event categories are calculated from events sets in the event warehouse. Probability dependencies are periodically recalculated as new events are added to the warehouse. The event prediction model is interfaced to a decision support system that can query the Bayesian model for events that have certain properties, for example, occurrence probability that exceeds a certain threshold. One important feature of the above architecture is that it offers a dynamically calculated dependent probability for events that is used for event prediction.

3. Bayesian Model for Flight Delay Prediction

3.1 Problem Domain

We applied the event prediction model described in Section 2, to the problem of predicting flight delays. Aviation is an interesting domain from an IoT viewpoint, for which data are easily obtainable. An airport environment has many objects with IoT properties such as sensors and systems for air traffic management. Many events are generated by an aircraft's systems before, during and after its flight, and also events generated by other objects at the airport or in the flight environment can trigger further events.

The objective of the experimentation was not to develop an accurate model for flight delay prediction, but to validate the steps of a practical methodology for constructing Bayesian networks for event prediction in IoT environments. Flight statistics data were collected from the web site www.flightstats.com which maintains detailed flight statistics as shown in Figure 2.

CDG Top 10 Departing Airlines

Airline						Delays			
Code	Name	Scheduled	Tracked	Departed	Cancelled	15-30	30-45	45+	On-time
(AF)	Air France	272	265	265	0	51	20	27	63%
(YS)	HOP!-REGIONAL	59	59	59	0	12	5	2	68%
(U2)	easyJet	47	47	47	0	13	1	7	55%
(WX)	Cityjet	19	19	19	0	5	4	1	47%
(LH)	Lufthansa	14	14	14	0	1	0	0	93%
(DL)	Delta Air Lines	11	11	11	0	2	1	1	64%
(BE)	Flybe	10	10	10	0	3	2	1	40%
(SK)	SAS	10	10	10	0	0	0	3	70%
(AZ)	Alitalia	9	9	9	0	2	1	2	44%
(VY)	Vueling	8	8	8	0	1	0	1	75%

Fig. 2: statistical airline performance from www.flightstats.com

4.2 Event Prediction Model

We developed a Bayesian event prediction model that calculates the probability of a connecting flight departing late, given the probability of the incoming flight departing late and/or the incoming flight arriving late. Being able to predict flight delays in advance is useful in many situations, as it allows replanning or rescheduling of other activities that follow up the arrival of a flight such as ground transportation.

We are interested in predicting the delay effect that an outbound flight departure delay will have on an inbound flight arrival, in order to be predict significant delays. For practical purposes, we are interested in delays longer than a certain threshold, as short delays are usually dealt by the time buffers built into the airport procedures, so that essentially they cause no or little disruption. More specifically, we are interested in departure delay probabilities that exceed a threshold, and might require re-planning or rescheduling of activities further down the transportation line.

Of course, a useful probabilistic model for airport flights, as well as for other IoT environments needs to consider many more delay factors in the dependency model. In the case of flights for example, weather conditions, traffic density (congestion) in the flight path and/or the departure and arrival airports, and other events such as personnel strikes, can contribute significantly to delays. An event prediction model also needs to be able to accommodate all these factors and their dependencies and calculate their probabilities on a continuous basis, as the state of the IoT objects change. Due to the complexity of the factor dependencies contributing to delays, event prediction model in such situations cannot give quantitative delay predictions. Instead, we expect to predict the type of the delay (according to some categorisation, i.e. short, medium or long). This approach can also be considered to be a naive

Bayesian classifier. Intuitively, the probability of an outbound flight departing late given a late departure of the inbound flight is calculated by the historical probability of the outbound flight department late, the strength of the probabilistic dependency of a delay on the inbound flight on the outbound flight, and the historical probability of delays in the inbound flight.
Our model being essentially a proof of concept one, however, considers airline and departure/arrival airports as the only known parameters.
We represent flight delays using the following general variables:
IFDL: inbound flight departure delay,
IFAD: inbound flight arrival delay,
OFDL: outbound flight departure delay.
By linking inbound flights to outbound flights, we establish the probability dependencies as:
IFDL→OFDL IFAD→OFDL and IFDL→ IFAD →OFDL

In the remainder of the section we only consider the IFDL→OFDL dependency, as for practical purposes it is more useful to predict the delay of the outbound flight departure from the delay of the inbound flight departure, rather than the inbound flight arrival delay, as the time interval between the two events is longer and provides more time to respond.
Formula (2) is now adopted for the flight delay prediction model as in formula (3)

$$P(outboundflightdelay \mid inboundflightdelay) = \frac{p_h(outboundflightdelay) * p(inboundflightdelay \mid outboundflightdelay)}{p_h(inboundflightdelay)} \quad (3)$$

We descritise the different delay values into three groups similar to the categorisation used in the flightstats website: L (low delays of up to 30 minutes), M (medium delays of 30-45 minutes) and L (delays over 45 minutes).
Next we expand the initial set of factors (IFDL, IFAD, OFDL) into the following variables to capture the above delay categories:

- SOFDD, MOFDD, LOFDD: Outbound flight departure delay (short, medium, long)
- SOFAD, MOFAD, LOFAD: Outbound flight arrival delay (short, medium, long)
- SIFDD, MIFDD, LIFDD: Long Inbound flight departure delay (short, medium, long)

Finally, based on formulas (2) and (3) we can calculate the following dependent probabilities:
P(LIFDD|SOFDD), P(LIFDD|MOFDD), P(LIFDD|LOFDD), P(LIFDD|SOFDD,SOFAD),
P(LIFDD|SOFDD,MOFAD),
P(LIFDD|SOFDD,LOFAD),P(LIFDD|MOFDD,SOFAD),P(LIFDD|MOFDD,MOFAD),
P(LIFDD|MOFDD,LOFAD),P(LIFDD|LOFDD,SOFAD),P(LIFDD|LOFDD,MOFAD),
P(LIFDD|LOFDD,LOFAD)

4 Case study

4.1 Data Collection

We collected flight delay data from flightstats.com for a period of 30 days from September to October 2015. To keep the data homogeneous, we restrict flight delay data to the same group of airports and airlines, as delays across airports can vary significantly. We collected historical flight delays for three airports (airport codes: FCO, CDG, BA) and three airlines that fly between these airports (airline codes: AF,AZ,BA). For each departing airport and airline, we collect probabilities of short delays (up to 30 minutes), medium delays (30-45 minutes) and long delays (over 45 minutes).

Table 1. Statistical delays of airlines per departure airport.

Airline code	Departing Airport	Short delay(S)	Medium delay (M)	Long Delay (L)
AF	CDG	19.7%	7.5%	10.2%
AF	LHR	19.1%	8.1%	1.5%
AZ	FOC	23.8%	3.1%	1.2%
BA	LHR	20.5%	8.8%	5.4%

Table 1 shows some of the delay statistics based on data available in flightstats website, used to calculate historical probabilities of the different delay categories for specific airlines and departing airports.

4.2 Conditional probability calculations

We calculate the strength of evidence probabilities, i.e. the dependent probability of experiencing a delay in an outbound flight given a delay in an inbound flight, from the statistical data collected from the flight statistics website. The strengths of relationships are as follows:

$SOFFD \rightarrow^{0.3} SIFDD$, $SOFFD \rightarrow^{0.2} LIFDD$, $SOFFD \rightarrow^{0.1} MIFDD$, $LOFFD \rightarrow^{0.3} LIFDD$, $LOFFD \rightarrow^{0.6} MIFDD$

For example, to establish, the probability of an AF flight from LHR to CDG having a long delay, given a long delay of the inbound flight, i.e. the probability P(LIFDD|LOFDD), we use the basic conditional probability formula (iix)

The historical probability of long delayed outbound flight from LHR to CDG, as shown in Table 1, is 0.015 .

The historical probability of long delayed outbound flight from CDG to LR, as shown in Table 1, is 0.102 .

The strength of the conditional probability, for this particular combination of flight and airport is calculated as 0.3.

Therefore, the conditional probability P(LIFDD|LOFDD) is calculated as 0.102 * 0.3 / 0.015= 2.04

This probability can be compared to conditional probabilities of other events, such as P(SIFDD|LOFDD) and P(MIFDD|LOFDD), to determine which category of delay has the higher probability for the outbound flight, for this particular case.

4.3 Results obtained

Table 2. results from testing the event prediction model with actual data

Outbound&inbound flight codes & dates	Outbound flight departure delay in minutes/Delay type	Outbound flight arrival delay in minutes/Delay type	Probabilities of flight dept. delays (S-M-L)	Inbound flight actual dept. delay in minutes/Delay type √:correct prediction
AF1068-1069, 13.10.15	41 (M)	51 (L)	0.86 0.1 0.057	35 (M)
AF1280-1281-3.10.15	28 (S)	18 (S)	0.3 0.105 0.04	0 (N) √
AF1580-1581 5.10.15	17 (S)	18 (S)	0.3 0.105 0.04	5 (S) √
AF1780-1781 5.10.15	28 (S)	27 (S)	0.3 0.105 0.04	0 (N) √
AZ322-323 5.10.15	24 (S)	18 (S)	0.35 0.16 0.04	49 (L)
BA560-561 5.10.15	24 (S)	19 (S)	0.3 0.105 0.04	3 (S) √
AF308-309 5.10.15	13 (S)	18 (S)	0.3 0.105 0.06	0 (N) √
AF322 5.10.15	24 (S)	18 (S)	0.3 0.105 0.06	49 (L)
AF326-327 5.10.15	96 (L)	99 (L)	0.2 0.2 0.3	31 (M)
BA432 5.5.10	53 (L)	21 (M)	0.2 0.2 0.24	61 (L) √
BA432 5.5.10	59 (L)	28 (M)	0.2 0.2 0.24	34 (M)
AZ7302-7301 5.5.10	24 (S)	44 (M)	0.35 0.16 0.04	13 (S) √
AF324-325 5.5.10	30 (M)	37 (M)	0.86 0.1 0.057	37 (M)

Table 2 shows the results of validating the event prediction model by comparing its output with actual flight delay results collected from the flightstats website. From table 2, we observe that the model assigns the highest probability to the correct delay event category in 53.8 % of the cases, which is better than random. The model's predictive power is weak, in particular in predicting the impact of medium or long delays in inbound departures, and thus further calibration of the model is required regarding the strengths of probability dependencies. A more elaborate model including additional factors such as time of day or season, impacting flight delays could also improve the

prediction accuracy.

5. Discussion and Conclusions

IoT environments are characterised by the large volumes of generated events and also by dynamicity, in terms of changing objects and interactions. Recently, systems such as publish-subscribe middleware have been designed to detect and disseminate events in IoT. In many IoT scenarios, it is important to predict events before they occur, in order to take corrective or preventive actions, as for example in the case of autonomous vehicles. In other situations events cannot be directly observed, and hence their occurrence must be predicted, based on the observation of other events. Other applications for IoT event prediction includes event-driven logistics chains comprising smart IoT objects[2], that need to synchronise inbound and outbound logistics activities.

A robust event prediction model requires a sufficiently large data set about historical IoT events. This dataset can be collected using global IoT infrastructures such as EPCIS[8]. Additionally, suitable Big Data technologies will be required for the probabilistic analysis of very large volumes of event data. As a future step in our research we plan to adapt existing Big Data analysis techniques to process the historical data required for our event prediction models, and to further investigate the research problems associated with the Bayesian approach to Big Data[4].

Acknowledgements

Research described in this paper has been partially supported by EU Project CORE “Consistently Optimised Resilient Secure Global Supply-Chains " Grant agreement no: 603993

References

1 Jin, Beihong and Chen, Haibiao. *Spatio-Temporal Events in the Internet of Things*. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, December 2010 pp. 353-358

2 Hribernik, Karl A. ,Carl Hans, Klaus-Dieter Thoben. *The Application of the EPC global Framework Architecture to Autonomous Controlled Logistics*. Third International Conference, LDIC 2012 Bremen, Germany, February/March 2012 Proceedings Editors: Kreowski, Hans-Jörg, Scholz-Reiter, Bernd, Thoben, Klaus-Dieter (Eds.)

3 Jensen, F. *An Introduction to Bayesian Networks*. UCL Press, London, 1996

4 Jordan, Michael I. *THE ERA OF BIG DATA*. THE ISBA BULLETIN Vol. 18 No. 2 June 2011

5 Mishra, PM. *Internet of Things and Bayesian Networks*. Available from http://www.analyticbridge.com/profiles/blogs/internet-of-things-and-bayesian-networks. July 9, 2014

6 Molaei, S.M. Keyvanpour, M.R. *An analytical review for event prediction system on time series*. 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA), 2015

7 Rish, Rina. *An empirical study of the naive Bayes classifier*. RC 22230 (W0111-014) November 2, 2001 Computer Science. IBM Research Report

8 Traub, Ken et al. *The GS1 EPCglobal Architecture Framework 2*. GS1 Version 1.6 dated 14 April 2014