Information Abstraction for Heterogeneous Real World Internet Data

Frieder Ganz, *Member, IEEE,* Payam Barnaghi, *Senior Member, IEEE,* and Francois Carrez

***Abstract*—Everyday around 2.5 quintillion bytes of data is created. There is also a growing trend towards integrating real world data into the Internet, which is provided by sensory devices, smart phones, GPS and many other sources that capture and communicate real world data. The term Internet of Things (IoT) refers to billions of devices which produce and exchange data related to real world objects (i.e. ”Things”). This paper focuses on how to optimise the data exchange between the sensory devices and applications in IoT and Cyber-Physical systems. In particular, a method to construct higher-level abstractions of data at local gateways is proposed. This will reduce the traffic load imposed on the communication networks that provide the real world data. The proposed method is based on an information processing algorithm where gateways analyse the data collected from the sensors and create higher-level abstractions. We enhance the Symbolic Aggregate Approximation (SAX) algorithm that is used as a building block of the abstraction creation framework, into an optimised version for sensor data, called SensorSAX. We extend the Parsimonious Covering Theory (PCT) that is usually used for medical purposes with a probabilistic parsimonious criterion in the temporal domain in order to infer abstractions based on time-dependent sensor data. The proposed method is analysed and evaluated over a real world dataset and the results are discussed in terms of the data size reduction, accuracy and latency needed to create the abstractions.**

***Index Terms*—Information Abstraction, Internet of Things, Wireless Sensor Networks.**

I. Introduction

THE Internet is facing a data overload. Every day around 2.5 quintillion bytes (10^{18}) are created [1]. This data comes from social media, audio and video content, document repositories, digital libraries, news websites, and many other sources of online information. In addition to the current produced data, it is predicted that in the next 5-10 years there will be around 50 billion Internet connected devices that will produce 20% of non-video Internet traffic [2]. The devices include sensor nodes, smart phones, GPS and many other sources that capture and communicate real world data. This large volume of data requires sophisticated mechanisms to satisfy the data communication and data management needs for the future Internet.

In this paper, we introduce a new paradigm for handling sensor data to decrease traffic in the real world data networks, by providing and exchanging abstractions that represent patterns, events and occurrences or other machine interpretable concepts, instead of communicating the raw sensory data. We differentiate the abstraction process into two stages. First the raw sensor data is abstracted into a low-level abstracted form represented as SAX patterns, that contain aggregated information based on the mean values of data from variable data windows. Afterwards the SAX patterns are used to infer higher-level abstractions inferred through our abductive and temporal reasoning component. This paper contributes three novelties to achieve the goal of reducing communication traffic by providing data abstractions, namely:

1. A version of the Symbolic Aggregate approXimation algorithm (SAX [3]) for the symbolic representation of time-series data optimised for sensor data. SAX leads to dimensionality and numerosity reduction and is the building block for many pattern and outlier detection algorithms ([4], [5], [6], [7]). The SAX method is described in Section IV.A). SensorSAX, an adapted version for sensor nodes introduced in this paper, exploits a variable encoding rate instead of a constant rate based on the activity in the streaming data and allows higher compression and fewer errors in reconstructing the original raw data.

2. An abductive reasoning model based on the parsimonious covering theory [8] in which sensors report different data that serve as the input for our model. Based on the data obtained from sensors, we abductively rule out the most unlikely phenomena that could have been caused by the sensors observations. The model is described in Section IV.B). The model is used to infer from the symbolic representation (SensorSAX) of the sensor data into higher-level abstractions such as ”warm”, ”dark” or ”no-attendance”.

3. Using the outcome of the extended non-temporal Parsimonious Covering Theory (PCT) in a temporal domain by introducing a Hidden Markov Model that includes the temporal dimension of the data. By taking the changes of states over time into the abstraction process it is possible to detect events that occur over time. This concept is discussed in Section IV.C)

A. Motivation

The recent advancements in integrating physical objects into information networks are opening a new paradigm. IoT networks are enabled by various devices and resources that collect data in the form of observations and measurements from the physical environment and communicate the data

Manuscript submitted January 31, 2013; revised May 29, 2013; accepted June 21, 2013

Frieder Ganz, Payam Barnaghi and Francois Carrez are with the Centre for Communication Systems Research at the University of Surrey, Guildford, United Kingdom. Email: F.Ganz@Surrey.ac.uk.

via the Internet to other devices or nodes that are interested in the data. Wireless Sensor Networks (WSN) are seen as one of the key enabling technologies to create networks of cooperating connected physical objects. The data collected by sensors needs to be communicated and to be made available to end-users (e.g. enterprise applications, business process services, web applications, software agents, human users and objects). The data communication between the capillary sensor networks and the Internet can be direct end-to-end connections or it can be facilitated via a gateway node. A gateway node can process or aggregate the data, process and respond to the data queries and make the data available through the network. In case that sensor nodes in the capillary networks are resource constrained, (which means that the full IP protocol stack is not feasible or could hinder the efficiency of the communications) gateway nodes can act as a bridge between the nodes and higher-level IP networks.

The raw data collected by sensor devices and communicated is crucial element to integrate the physical world observations into the cyber world or create what is referred to as the Real World Internet [9]. However, data consumers are often not interested in the raw sensor data and thus data is collected and processed to create domain-level abstractions and knowledge that can be extracted from the real world data. For example, a user may be more interested in the information that the door is open than in the raw analogue sensor readings that lead to this conclusion. This fact can be used to save the communication traffic between the network nodes by processing the sensory data locally at node or gateway level and creating higher-level abstractions that can be then communicated to the user.

B. Objectives

In this work, we focus on local data processing and creating higher-level data abstractions that can be communicated globally. We store and process the raw sensory data in the gateway and make it available via service interfaces. The gateway is an intermediary node that enables the communication between device networks (e.g WSN) with the core network (i.e Internet). The raw data can be accessed and communicated over the Internet by accessing the gateway service interfaces. However, our main focus is on processing the data locally at the gateway and representing it as abstractions that are smaller in size. This reduces the size of the data that needs to be communicated instead of offering the raw data (unless it is directly requested).

Abstract representations refer to an occurrence or a pattern in data, and they are also used to provide a multi-granular access to the data. Users can receive the abstractions and then if they were interested in the raw data, then it can be communicated to them (the records or intervals that are required). This reduces the size of data emerging through the deluge of connected devices and allows the gateways to manage the flow of data and send higher-level information or extracted knowledge from data that is processed locally, instead of flooding global networks with raw data from the real world.

We have implemented the proposed mechanism on a gateway component that we developed in our previous work [10]. The proposed solution supports device heterogeneity by introducing a gateway component that enables the collection of data from several heterogeneous sources via different hardware interfaces (described in Section 3). We have integrated our abstraction mechanism as part of the processing steps in the gateway. To evaluate the SensorSAX algorithm we use TMote Sky nodes [11] to compare reconstruction error rate and power consumption of the different algorithms. To evaluate our abstraction creation process, we use the sensory data collected online from the UK Channel Coastal Observatory resources[1].

We use an abductive reasoning mechanism (described in Section 4) to create abstractions from the meteorological data (i.e. in this case we use it for tidal monitoring). The inferred abstractions are then compared with those reported by a tidal reporting program to evaluate the accuracy and quality of our results. Overall, the proposed methods can reduce the size of communicated data from local weather stations by 80%, with preserving key information and patterns of data.

The rest of the paper is organised as follows: First we present current approaches in data communication optimisation and emphasise the novelty of our approach. Section 3 describes our gateway architecture, which serves as the host component to run our proposed algorithm. Section 4 introduces the proposed framework to create abstractions based on the raw sensor data gathered by the gateway. Section 5 provides an evaluation and discusses the results. Section 6 concludes the paper and describes future work.

This paper extends our previous work reported in [12] by 1) introducing a probabilistic extension of the initial static model, 2) designing a temporal reasoning model and 3) developing a discretising algorithm for streaming sensor data. In the current paper, we also provide extensive evaluation of the algorithms on a real world dataset and demonstrate the efficiency of the proposed solutions based on an experiment using real sensor nodes.

II. Related Work

Traffic aggregation and data compression are the main solutions used to reduce communication traffic in IoT. In this section we review some of the common solutions and highlight the uniqueness of our data abstraction approach.

Data aggregation or data fusion can help by removing redundant data from the gathered sensor streams. Chen *et al.* [13] give an overview about approaches that create a summarised data stream of a set of sensory data streams and use the aggregated data for transmission. The aggregation of the data usually relies on the mathematical sum, max, min, average and count aggregate functions [14].In large distributed WSN this usually happens via clustering algorithms; however this can lead to loss of important data that has been masked due to the aggregation in lower layers. Chen et al. also review different approaches of information extraction such as Kalman-Filtering, Neural Networks and Probabilistic Models, however these methods are not evaluated over real datasets.

[1] http://www.channelcoast.org/

TABLE I: Feature identification of different approaches

	Compression Technique	Approach	Information Abstraction
Chen et al.	Tree-based Aggregation	In-Network	Kalman-Filtering Neural Networks Probabilistic Models
Wang et al.	Sparse Data Transmission	In-network Centralised	Abnormal Reading Detection
Yun et al.	Binary Encoding	Centralized	-
Stocker et al.	Bandpass FFT	Centralized	Machine-Learning
Ganz et al.	Data discretizing	In-Network Centralised	Reasoning

A different approach to reduce the communication traffic in communicating the sensory data is to reduce the size of the messages. This can be realised using data compression algorithms. However, compressing the data itself could lead to a loss of information (in lossy compression) and the compression techniques can require higher power consumption as compression requires data processing before transmission and in long term observations (e.g. environmental monitoring applications) compression techniques data can still create large amounts of data [15].

Wang *et al.* [16] use compression techniques with adaptive sensing for WSN. This can be exploited to transmit only very dense (and therefore small in size) data and then reconstruct the overall ”data” by applying a reverse function. However, the constructed data relies on several incomplete data-streams and therefore this can lead to a huge loss in the quality of the reconstructed information. Wang et al. introduce an abnormal reading detection mechanisms that is able to find outliers out of the reconstructed data, however the approach is not able to detect and highlight events that occur on a regular basis. A new way to reduce data communication in Real World data networks is to extract the information which is relevant or important for the user before transmitting it. However, determining what is required or what is important to the user from heterogeneous data sources and in the wide range of applications that IoT and Cyber-Physical systems can use is not a trivial task. So it is important to define methods that can create higher-level abstractions that can be general purpose and then to use abstract reasoning models that can transform these abstractions into machine interpretable or human understandable knowledge.

Stocker et al.[17] introduce a system to acquire knowledge that is represented in a semantic database by abstracting from the physical sensor layer and the sensor data layer. At first, the data is pre-processed by applying a bandpass filter to the raw sensor data and filtering the relevant frequencies. The bandpass filter is implemented using fast fourier transformation and summarising the values of a time window to provide input for the next step which is detection and classification using machine learning. The authors use a multilayer perceptron (MLP) neural network classifier to detect and classify different events and abstractions. The outcome of the classification process is then transferred into the semantic database. The limitation of the approach is the long non-automated learning process that requires domain experts to feed the model with sample data for supervised learning processes. In this paper we introduce an approach that infers abstractions based on pattern representations.

Yun *et al.* [18] introduce a similar approach that exchanges information using ”signatures” instead of the raw data. The signatures are combinations of properties measured during an event such as ”bright light” and ”loud sound” during the explosion of a bomb. The signatures consist of a string representation indicating that something is present or absent at a particular sensor. This is efficient in terms of data communication as only binary data has to be transmitted such as ”NYY” standing for ”No” - light is not present and ”Yes” sound and temperature are present; however is not precise in terms of defining various patterns of data and its extendibility and scalability is also limited. Moreover, the approach gives only insights about the local sensor readings and does not allow to extract information in global networks. In our approach, we discretise the data and create pattern representations of the source data which can be used to describe the transient states of a sensor data stream (e.g. low noise (A), medium noise (B), high noise (C) resulting in ABC instead of binary No). Then these patterns are fed into a probabilistic reasoning model to create higher-level abstractions from the data that is emerging from several sensor data streams.

In Table I we show the various work and their main features regarding compression/aggregation technique, location of the approach and the information abstraction method. Most approaches use mechanisms that are only applicable to certain use-case domains such as discrete fourier transformation for signal data and binary encoding for information that has only two states. However through the tractability of our SensorSAX mechanism, granularity is automatically adapted based on the volatility of the data, as described in Section IV-A.

III. System Model

We have implemented our solution on a gateway component developed in our previous work [10]. The gateway component provides a connection between heterogeneous sensor devices with low processing and communication capabilities and higher-level services and applications on the Internet. The gateway can be equipped with several air interfaces such as IEEE 802.15.4, IEEE 802.11 and Bluetooth to support a variety of communication links over the physical layer. The network protocols such as 6LoWPAN [19] and Zigbee protocol stack [20] are also supported to enable network layer communications. In Figure 1, the main components and basic workflow of the system are depicted. The sensor nodes transmit either raw data or run our SensorSAX module and transmit aggregated information to the data collection component on the gateway. The data collection layer includes different interfaces and is able to collect and store data from heterogeneous nodes. After collecting the data, it is forwarded to the processing layer. In case the data was not aggregated and discretised by a SensorSAX module, the gateway applies the pattern creation and discretisation process by applying the SensorSAX algorithm. The abstraction process is then performed by using abductive and temporal reasoning methods. The abstracted data is finally accessible through the data provision layer where different mechanisms such as web-services or a graphical user interface can be used to access the data.

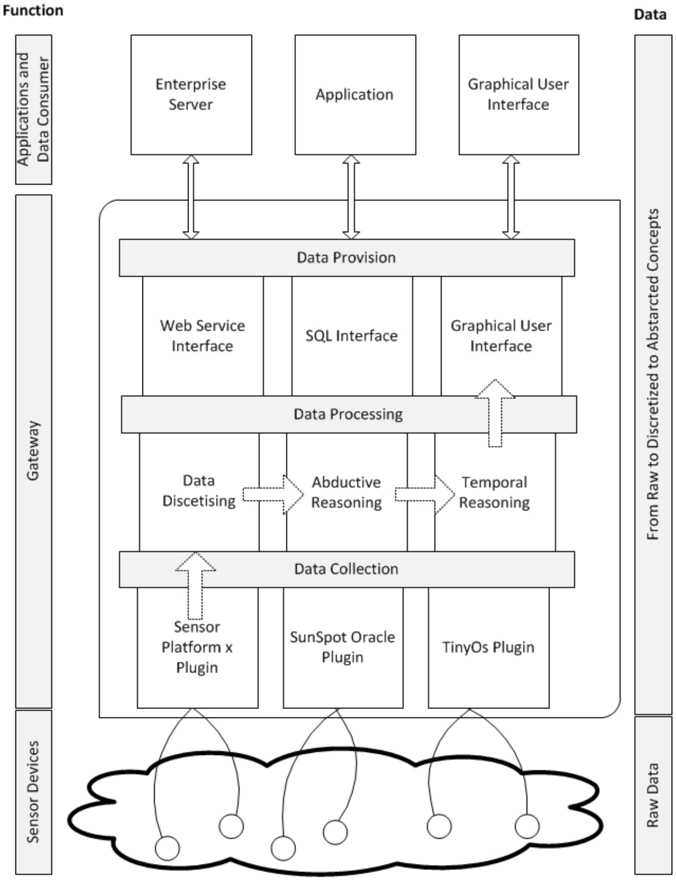


Fig. 1: The System Overview

A. Data Collection

The data collection tier provides wrappers for different hardware and software platforms and supports communication between different sensory devices and the gateway. The wrappers are implemented as plug-ins for different data sources. The data collection layer provides an abstract view and hides the complexity of underlying devices (i.e. sources) and the proprietary hardware and software specifications. The wrappers include a protocol for negotiation and association of sensory devices to the gateway [10]. This modular design makes it feasible to automatically setup and connect a large amount of heterogeneous sensor sources. In the current gateway different plug-ins are implemented for TinyOs [21], Contiki [22] enabled devices and Oracle SunSpot [23] nodes. Further plugins for WiFi or the Zigbee standard can be developed by attaching the respective hardware interfaces to the gateway and developing the wrapper plugins to handle the communication between the standard interface and the data collection component. The architecture also supports large-scale setups where gateways can establish connections between other gateways in WSNs. For large scale Gateway-to-Gateway (G2G) communication scenarios, the communication process to exchange and update information among the gateways for data processing including the reasoning is described in our previous work and not the main focus of the paper.[24] For evaluation purposes of the abstraction creation process, we also implement a plug-in to access a large sensor data repository (UK Coastal Observatory) via a provided API to get more representative results.

B. Data Processing

The data processing layer provides caching, storage and processing functionalities. In this paper we focus on this layer and describe how our proposed solution is implemented in this part of the gateway. The data processing tier uses caching and storage functions to minimise direct interaction with the sensory devices. The important aspect of the gateway is the ability to process and interpret the data and create higher-level abstractions. This provides a local computing and data abstraction and a global data/concept communication paradigm which allows more efficient communication and integration of large sensory data.

The data processing tier consists of three sub components: data discretising, abductive reasoning and temporal reasoning. The data discretising component is used to discretise the raw sensor data into lower-dimensional representations. This component utilises an extended version of the Symbolic Aggregate approXimation (SAX) algorithm [3], called SensorSAX optimised for sensor data, to convert continuous data (e.g. {1,2,3,4,5,4,3,2,1}) into a compressed discretised representation (e.g. {a,b,b,a}). The abductive model stores the mapping between discretised representation and abstractions (e.g {a,b,c,d} represents ”attendance” in a room). The abductive reasoning and temporal component infers the current observations and determines which abstractions are the most plausible ones.

C. Data Provisioning

The data provision tier provides different interfaces for the data access such as Web service interfaces, and APIs to query and retrieve the abstracted concepts for traffic efficient communications. It also provides direct raw data access if it is required.

IV. Abstraction Creation Model

The aim of abstraction creation in this work is to generate higher-level abstractions from raw sensor input and transform the raw data to smaller size machine interpretable or human understandable concepts. The abstraction creation process consists of three main components: Data Discretising, Abductive Reasoning and Temporal Reasoning, as shown in Figure 1 and Figure 2.

The data discretising transforms the raw continuous sensor data into a dimensionality reduced “pattern” representation. The patterns are then evaluated and linked to abstractions (with relevance probabilities explained in Section IV-B). This allows a possible abstraction to be found, based on the observations. To abstract an event or phenomena its current and past states

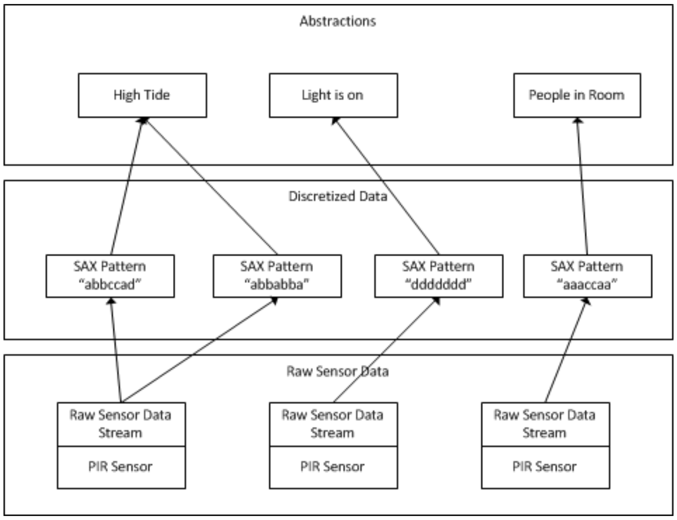


Fig. 2: Processing from raw data to abstraction

are taken into account. To include the temporal domain we propose a temporal reasoning model (described in Section IV-C).

A. Pattern Creation for Resource-constrained environments

The SAX algorithm is mainly used as a building block for detecting patterns and outliers, data aggregation, clustering and classification from and in time-series based data ([4], [5], [6], [7]). Some of the main advantages of the algorithms are the high compression rate while retaining the main features of the original data and providing a distance measurement function with high correlation on the distance function of the original data. The algorithm is divided into three steps: Normalisation, Piecewise Aggregation Approximation (PAA) and discretising of the aggregated data. During normalisation, the data is processed to have a standard deviation of 1 and an average of 0 to enable comparison of data from different sources and reducing the numerosity of the sensor data.

```
function PAA($outputLength, data$)
    $w := length(data)/outputLength$
    $output$                                        ▷ Output Vector
    while $pointer < length(data)$ do               ▷ Iterate data
        $segment := data_{(pointer,pointer+w)}$
        $output_n = mean(segment)$
        $pointer = pointer + w$
    end while
    return $output$
end function
```

Fig. 3: The original PAA function

```
function SensorPAA($minOut, maxOut, \rho, data$)
    $w_{min} := length(data)/minOut$
    $w_{max} := length(data)/maxOut$
    $w_{init} := w_{max}$                           ▷ Initial Length
    $output$                                        ▷ Output Vector
    $pointer := 0$
    while $pointer < length(data)$ do               ▷ Iterate data
        $w = w_{init}$
        $segment := data_{(pointer,pointer+w)}$
        if $stdDeviation(segment) < \rho$ then
            $w := w + 1$
        else
            $output_n = (w, mean(segment))$
            $pointer = pointer + w$
            $continue$
        end if
        if $w >= w_{min}$ then
            $output_n = (w, mean(segment))$
            $pointer = pointer + w$
        end if
    end while
    return $output$
end function
```

Fig. 4: The modified PAA function in the SensorSAX algorithm

Piecewise Aggregation divides the original data of length N into n equally sized windows by taking the mean of each window. This results in a reduction of data size from N to N/n data points. A shorter window length n results in a better reconstruction of the original data, however more data space is needed to store the data and eventually higher energy consumption by higher communication costs. The original PAA algorithm is depicted in Figure 3. The function takes two input parameters; the list data containing the raw values and the desired output length. The function iterates over the data and takes the mean of the list segments to achieve the desired output length. The output is a list that contains the segment means. For instance, running the function with the parameters data:[1,2,3,4,5,6] and outputLength:2 leads to an output of [2,7.5].

The data is then split vertical according to the alphabet size a. The break lines are distributed vertically according to the Gaussian distribution, more lines closer to 0 (as 0 is the mean of the distribution). The amount of break lines is dependent on the alphabet size, the more letters are used, the more break lines will be used to split the data vertically, each break line standing for one letter. Each window is assigned a letter, depending on which break line the average of the window resides under. The complete process of discretising raw data into a SAX representation is depicted and described in Figure 5. The drawback of the original approach is in the constant window length, the sax developers state that it is infeasible to choose the right parameters n (window length) and a (alphabet size) because of their high data dependency [3]. The authors therefore empirically choose the best parameters, based on several different datasets and the experimentation of different combinations of n and a. However, in sensor environments it is often the case that there is no or less activity in the sensed data and therefore it is not necessary to transmit data with the same window length. We adapt the piecewise aggregation step of the SAX algorithm to have a variable window length, depending on the volatility of the data. This leads to a

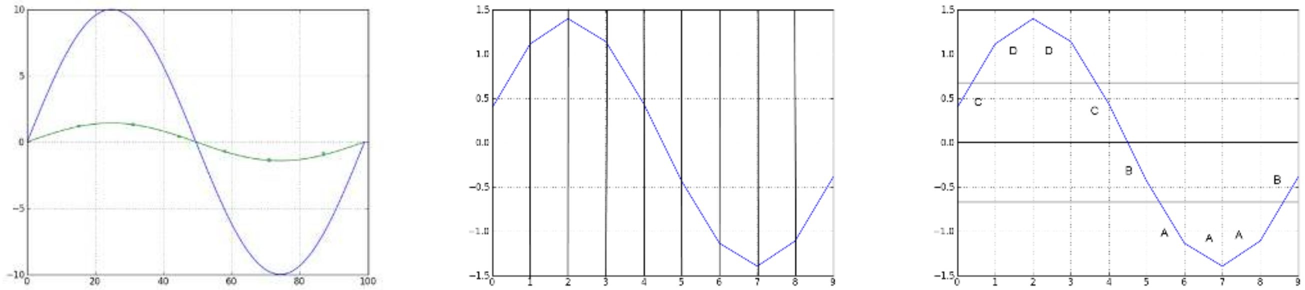


Fig. 5: Dimensionality Reduction Process of SAX

(a) The blue line represents 100 data points of a sine curve. The green (dotted) line is the curve after normalisation

(b) The normalised curve is divided into 9 windows. Each window represents the mean of 100/9 datapoints from the original data. That leads to a compression rate of n/N 9/100

(c) The output windows of the PAA is then divided vertical. Each segment is assigned to a letter. The segment in which the curve is in per window forms the SAX word, in this case CDDCBAAAB

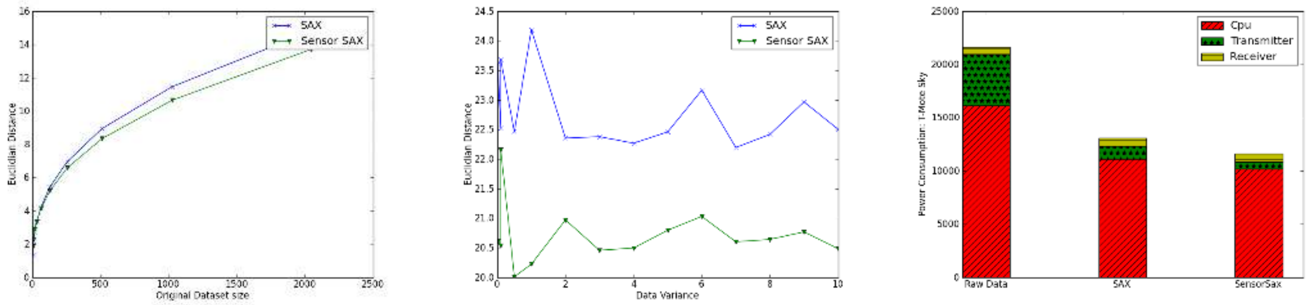


Fig. 6: Evaluation of SAX and SensorSAX

(a) The reconstruction error rate of SAX and SensorSAX over different random datasets with different size. To be comparable the output window length of SAX is the average length of the variable SensorSAX windows.

(b) The reconstruction error of SAX and SensorSAX over different random datasets with different activity and variance from 0.01 up to 10

(c) The energy consumption of a TMote Sky Sensor node for each sensor hardware component. Transmitting Raw Data, SAX transformed data and SensorSAX transformed Data

better reconstruction rate of the original data and less energy consumption, as less data is transmitted, as shown in Figure 6 c). Instead of a fixed window length we use the parameters minOut as minimum length of the output (N/n), maxOut as maximum output and ρ as a threshold value for the sensibility to reduce the output length for less active data. Our extended algorithm calculates the current activity in the observed data set based on the standard deviation, and either chooses a larger window length if there is low activity to reduce transmission cost, or a shorter window length which will lead to higher transmission costs but better reconstruction of the original data, including important features that are needed by the abstraction creation process in the following sections. The extended PAA algorithm is depicted in Figure 4 that accepts the input parameters: minOut, maxOut, p and data, and produces an output list with an average length of maxOut/minOut. In Figure 6 a) several random generated datasets with different length were generated and then discretised. In average, (over 100 random datasets per dataset size) SensorSAX has a better reconstruction rate of the original data. To measure the error in reconstruction we used the euclidean distance. In Figure 6 b) several random datasets with different variance in the data were generated. Independently from the volatility of the data, SensorSAX leads to a better reconstruction rate. To evaluate the power consumption on a real world node, we used TMote Sky [11] nodes that transmit sensory data over a certain time window. We compared energy consumption of raw data transmission, SAX processed data transmission and SensorSAX processed data transmission. Figure 6 c) shows that SensorSAX performs the best in energy consumption, even though more mathematical steps to compute standard deviation are required.

B. Abductive Reasoning

We use the Parsimonious Covering Theory (PCT) [8], an abductive logic framework, to transform the sensor data into abstractions. The parsimonious covering theory is predominantly used in the medical domain to model and infer the disorder of a patient based on observations made by a doctor. It uses an abductive approach which is based on partial observations. To give a brief example: A doctor asks the patient during the diagnosis, many different questions (How old are you?, Do you have fever?) to abductively rule out diseases unrelated based on the response given by the patient.

We exploit this approach in our abductive reasoning model. Sensors are reporting different readings. Based on the state of the sensor we abductively rule out the most unlikely abstractions that could have been caused by the present observations such as warm, high-water level and pressure. The novelty of our approach is to provide a probabilistic approximation of the different possible abstractions. Abductive reasoning infers the most likely explanation given a set of incomplete or partial observations. In contrast to deductive reasoning where a conclusion in a complete and sound system can always be inferred, abductive reasoning only gives an educated guess about the most likely explanation in an uncertain environment. The advantage of abductive reasoning is that for partially-observable concepts and incomplete observations a conclusion can be given, whereas deduction would need the complete observations to draw conclusions.

As far as we are aware, the use of PCT in information retrieval was first reported by Syu *et al.*[25] and the use of PCT for sensory data was first reported by Henson *et al*'s work in [26]. Henson *et al.* assume that the raw sensory data is presented in terms of higher-level observations and then apply PCT to these observations to draw conclusions and create a list of possible events. PCT uses two criteria to find an explanation for some observations: 1) A coverage criterion and 2) the parsimony criterion. The coverage criterion creates a set of explanations which includes each observation in the explanation set. To reduce the explanations, the parsimony criterion selects the best explanations. Many different parsimony criteria have been developed, such as the single-disorder, minimum cardinality, and irreducibly criteria. However, setting these criteria and transforming the raw sensory data into observations is not a trivial task and is not deterministically measurable. The following describes the probabilistic parameters that are used in our work to select the most likely explanation for a given set of observations.

We define our extended PCT model as follows: The abductive model uses two finite sets to define the scope of an abstraction. They are set A, representing all possible **abstractions** and set O, representing all **observations** that may occur when one abstraction is present (Observations are discretised patterns from the SensorSAX algorithm). To find the causations between abstractions and observations, we define a relation C, from A to O. The relationship $\langle a_i, o_j \rangle$ indicates that a_i is one of several possible abstractions of an observation o_j. Let's assume that "ABCD" and "EFGH" are observations o_1, o_2 and "high tide" an abstraction a_1. The fact that the observations "ABCD" and "EFGH" are possible signs for the abstraction "high tide" is denoted as $\langle a_1, o_1 \rangle \langle a_1, o_2 \rangle$. The sets A and O and the relationship C create the knowledge-base of the model. The knowledge-base is a simple (because of a maximum depth of 1) Bayesian network. For our example we assume that we have o_1,o_2 and a_1 and their relationship in our example knowledge-base (partly depicted in Figure 7) .

To find a possible abstraction based on the observations a 4-Tuple is defined, which is shown as $P = \langle KB, O^+ \rangle = \langle A, O, C, O^+ \rangle$ where $O^+ \subseteq O$ is the set of current observations made by active sensors.

We find two functions in the PCT, namely $causes(o_j)$ representing all possible abstractions of a given observation and $effects(a_j)$ representing all possible observations of a given abstraction. $causes(o_1)$ would lead to an abstraction e.g "high tide", whereas $effects(a_1)$ would return "ABCD" and "EFGH".

A set of abstractions $A_I \subseteq A$ is called a *cover* of a set of observations $O_j \subseteq O$ if $O_j \subseteq effects(A_I)$

a_1 is a cover for the observation set o_1 and o_2.

This definition makes it likely that there could be different abstractions for a set of observations. To find out the plausible abstractions, PCT follows the concept of "Occams' Razor"[2] that refers to selecting among explanations and chooses that one that makes the fewest assumptions. A cover is called a parsimonious cover when its abstraction covers O^+, but also satisfies being parsimonious. This means that only the simplest explanations are chosen. This leads to the definition of simplicity in the context of choosing the explanations.

To define the simple explanations we use a likelihood weighting used to determine the plausibility of a hypothesis by calculating its probability and comparing with the probabilities of the other hypothetical abstractions which cover the current observations. We use the utility functions introduced by Peng and Reggia [27] and extended it for discretised data.

In order to define the weights and calculate the likelihood several probability factors are introduced, as shown in Figure 7. A probability p_i defines the likelihood of the different abstractions for each $a_i \in A$. In our example, the probability of a_1 to be the current observation "high tide" is $p_1 = 0.3$. A causal strength $0 < c_{ij} < 1$ defines how frequently a_i causes o_j. Therefore o_2, with $c_{12} = 0.7$ is more likely a reason for a "high tide" abstraction than o_1 with $c_{11} = 0.3$. This allows to derive a formula to calculate the possibilities for certain abstractions under the current observations:

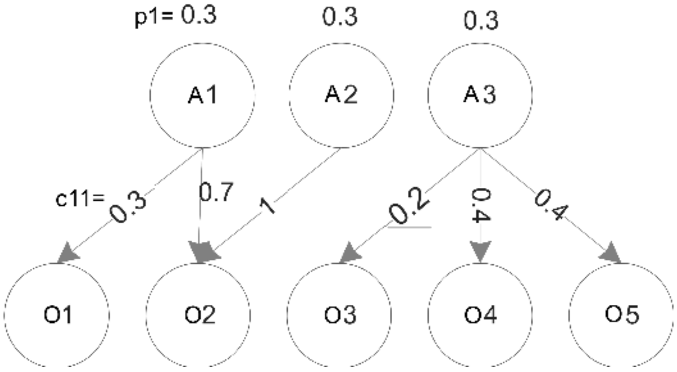


Fig. 7: An Example of an extended PCT with probabilities

$$L(A_I, O^+) = L_1(A_I, O^+) * L_2(A_I, O^+) * L_3(A_I) * L_4(O^+) \tag{1}$$

The first product, shown in equation (2), represents the likelihood to cause the presence of the abstractions in the given

[2] http://en.wikipedia.org/wiki/Occam%27s_razor

O^+. In other words, how likely is it that o_1 and o_2 are reasons for a_1

$$L_1(A_I, O^+) = \prod_{o_j \in O^*} (1 - \prod_{a_i \in A_i} (1 - c_{ij})) \tag{2}$$

The second product, equation (3) calculates the weights based on expected observations in the knowledge base within A_i but not observed. If our current set of observations only consists of o_1 it is less likely to be a ”high tide” abstraction as o_2 is not present.

$$L_2(A_I, O^+) = \prod_{a_i \in A_I} \prod_{o_i \in effects(a_i)} (1 - c_{ij}) \tag{3}$$

The third product, equation (4), represents probabilities related to p_i. In our example, the chance that the current state is ”high tide” will be 0.3 as p_1 is defined as 0.3.

$$L_3(A_I, O^+) = \prod_{a_i \in A_I} \frac{p_i}{(1 - p_i)} \tag{4}$$

And the fourth product, equation (5), takes the distance of the discretised pattern representation into account. The distance between the observed discretised values defines the likelihood of relevancy between a pattern and an abstraction. This allows us to take small variances in the discretised data into account. (e.g. ”ABCD” and ”ABCE” are closer to each other than ”BBBB” and therefore ”ABCD” could also be an observation for the ”High Tide” abstraction). The distance function is defined in the original SAX algorithm [3].

$$L_4(A_I, O^+) = distance(O^+, O) \tag{5}$$

To find the most relevant explanations of a set of observations, the abstraction with the highest probability is chosen in equation (6).

$$Y = a^* \ where \ a^* \in A_I : max(L(A_I, O^+)) \tag{6}$$

However, PCT was never designed to model temporal observations as they occur in processing continuous real world data. Sensors make observations over time and therefore can change the inferred explanation each time a new observation is made. PCT however is only a static model and only represents the variables during a time slice. Therefore, a time-dependent model is also needed to infer explanations from observations during a period of time. In the next section, we describe the extension of the parsimonious covering theory with a temporal component.

C. Temporal Reasoning

The output of the abductive model can vary over time. Especially in sensor networks the state of an observation is constantly changing and most likely following some patterns. To perceive a concept or phenomena both the current and past states are required. To model and include this time-dependent aspect of higher-level concept creation, we combine the previous static model with a Hidden Markov Model (HMM). The HMM enables to use the abstractions obtained from the static abductive model to acquire events and processes that occur over a certain amount of time. The temperature change during a day from cold over warm to cold that represents a ”regular” temperature pattern can be modelled as a new hidden state that is dependent on several abstractions inferred during the day. Derivations from this pattern can lead to newly observed hidden states that eventually can be perceived as outliers.

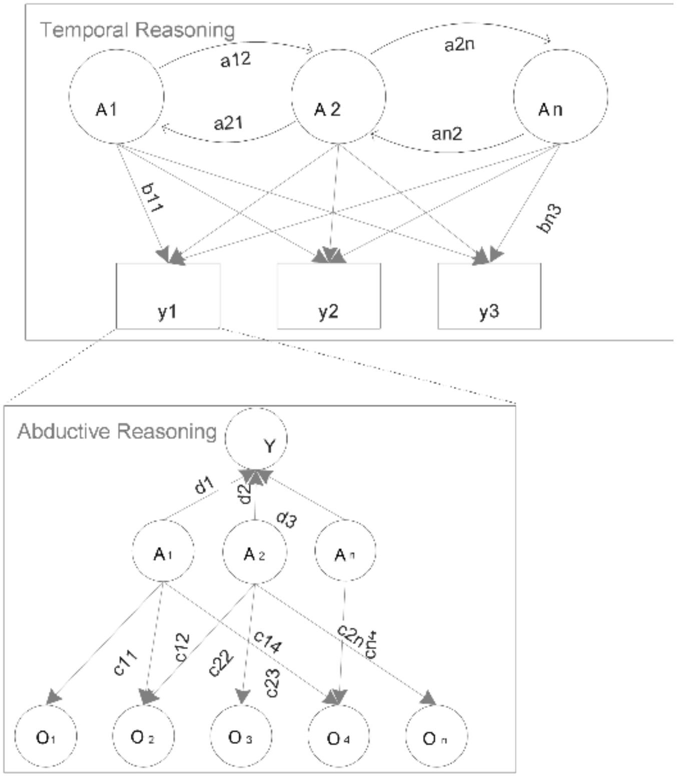


Fig. 8: Complete abstraction model

The output $Y(A_I, O^+)$, the inferred abstraction, under a given set of possible abstractions and current observations serves as the input for the dynamic model as shown in Figure 8. The overall observation process is divided into several time windows. The window size enfolds a fixed amount of observations, made by a sensor and can vary depending on the sample rate of the different sensors. To model temporal relations we use a hidden markov model, consisting of the following 4-tuple: $HMM = \langle X, A, Y, B \rangle$ where X is the set of time-dependent abstractions, A is the transition probabilities between the different abstractions, Y is the emission parameter - output of our previous abductive model, B represents the probabilities b_{ij} to make a time-dependent abstraction x_i based on the output of y_j. Based on the frequency and type of occurring events, the HMM model changes the probabilities and provides feedback for the static model. For instance in our example from the previous section, is it more likely that the current observation will be ”high tide” after the detection of a ”low tide” pattern.

This model transforms observation and measurement data (originated from sensory devices) and relations into higher-level abstractions to formalise concepts and knowledge from the underlying raw data. Figure 8 depicts the relations in temporal and original abductive reasoning models.

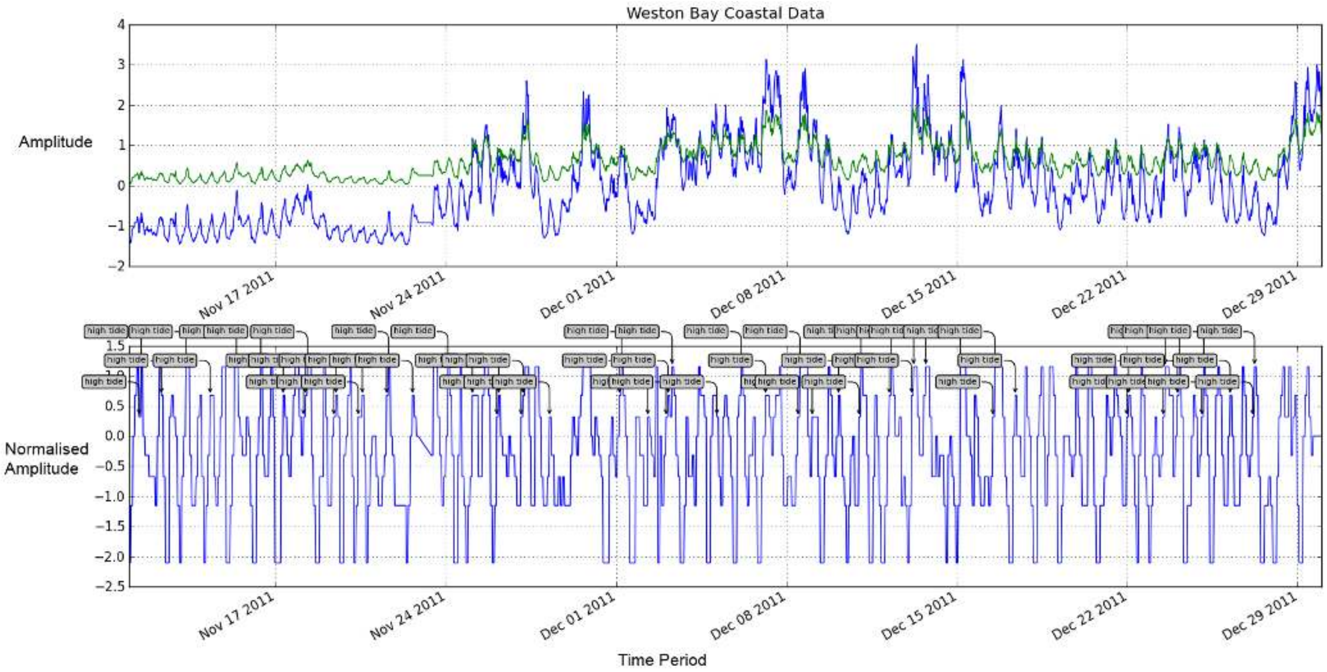


Fig. 9: The original and abstracted data

V. Evaluation

To prove the feasibility of the proposed approach, we measure accuracy, data size reduction and latency of the process to create the abstractions. We use the UK coastal observation data provided by the Strategic Regional Coastal Monitoring Program for the evaluation. A plug-in is implemented that uses the Channel Coastal Observatory API[3] and reads the datastreams into our gateway. The gateway collects the data from several stations and provides the abstracted information.

We evaluate the accuracy of the method by comparing the constructed abstractions with a tidal time table calculated with the help of a tide and current prediction program[4]. We use tidal time table data to show that the model is able to transform sensor readings to abstractions which occur in the real world. We measure the data size reduction (i.e. we only measure the data size and other communication protocol overheads are not included) and calculate the average correlation between the original and reconstructed data. The results show that the data has a correlation coefficient of 0.89 with a positive direction which means that the reconstructed data is very similar to the original data. The execution time is also measured for the construction of a set of abstractions over a data collection period.

A. Accuracy

The precision and recall metrics have been used to evaluate the reconstructed data. Precision and recall are predominantly used in information retrieval to evaluate search algorithms [28]. In this work recall is used to measure the completeness in terms of retrieved and relevant abstractions by the algorithm compare with the real events as they are reported. Relevant abstractions are abstractions which could be mapped to an event in the real world. Recall is defined as follows:

$$recall = \frac{relevantAbstractions \cap retrievedAbstractions}{totalRelevantEvents}$$

Precision, in the current work, measures the accuracy of the result by comparing relevant abstractions with the total number of retrieved abstractions. Defined as:

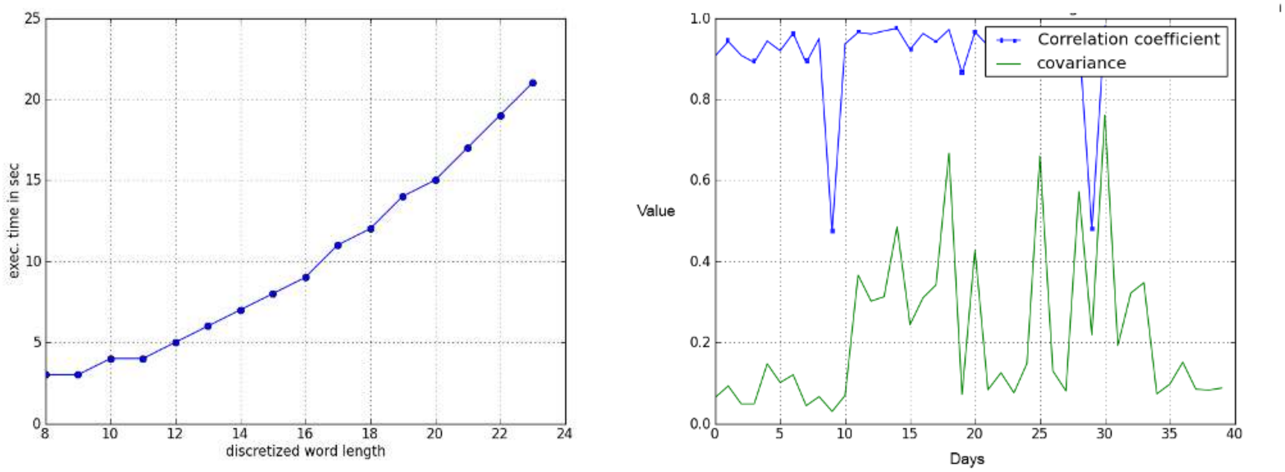
$$precision = \frac{relevantAbstractions \cap retrievedAbstractions}{retrievedAbstraction}$$

Figure 9 shows a set of coastal data collected over a period of time between November and December 2011[5]. The top diagram in Figure 9 shows the raw sensory data shown as a time series data. The blue line represents the raw sensor data, the height of waves measured in the Weston Bay coastal observation station. The green line represents the z-normalised data that is used to discrete and reduce the dimension of the data by applying the SensorSAX algorithm. The second diagram in Figure 9 shows the constructed abstractions from the data. The abstraction process is evaluated using three different probability sets depicted in Table II (a,b). The tables show the ”training” data. In particular they show which discretised pattern is represented with what abstraction and with what probability. To show the feasibility of our approach we choose small training sets. The probability values for the training data shown in Table II incrementally use more observations. Each table shows relation between discrete value and the ”high tide” abstraction (shown in Figure 9) and

[3] http://api.channelcoast.org/

[4] http://www.wxtide32.com/

[5] Due to the space limitations, we cannot show a detailed version of the diagram. A higher resolution version of Figure 9 is available at: http://tinyurl.com/c3hfoll



(a) Latency to create abstracted data

(b) Correlation between raw and reconstructed data

Fig. 10: Evaluation results of abstraction creation

High Tide Abstraction	Discretised String length =12	
	Discretised Value aabbdeggghfe	Probability cij 1

High Tide Abstraction	Discretised String length =12	
	Discretised Value aabbdeggghfe bbbcccbcfghh	Probability cij 0.5 0.5

(a) First and Second Abductive Probabilities

High Tide Abstraction	Discretised String length =12	
	Discretised Value aabbdeggghfe bbbcccbcfghh hgeeecbcbbcd	Probability cij 0.3 0.3 0.3

(b) Third Abductive Probabilities

HMM Probabilities	Transition Probabilities	Emission Probabilities
High Tide	Low Tide=0.9, High Tide=0.1	$Y = a^*\ where\ a^* \in A_I : max(L(A_I, O^+))$
Low Tide	Low Tide=0.1, High Tide=0.9	$Y = a^*\ where\ a^* \in A_I : max(L(A_I, O^+))$

(c) HMM (Temporal Probabilities)

TABLE II: Initial values for the abductive and temporal reasoning model

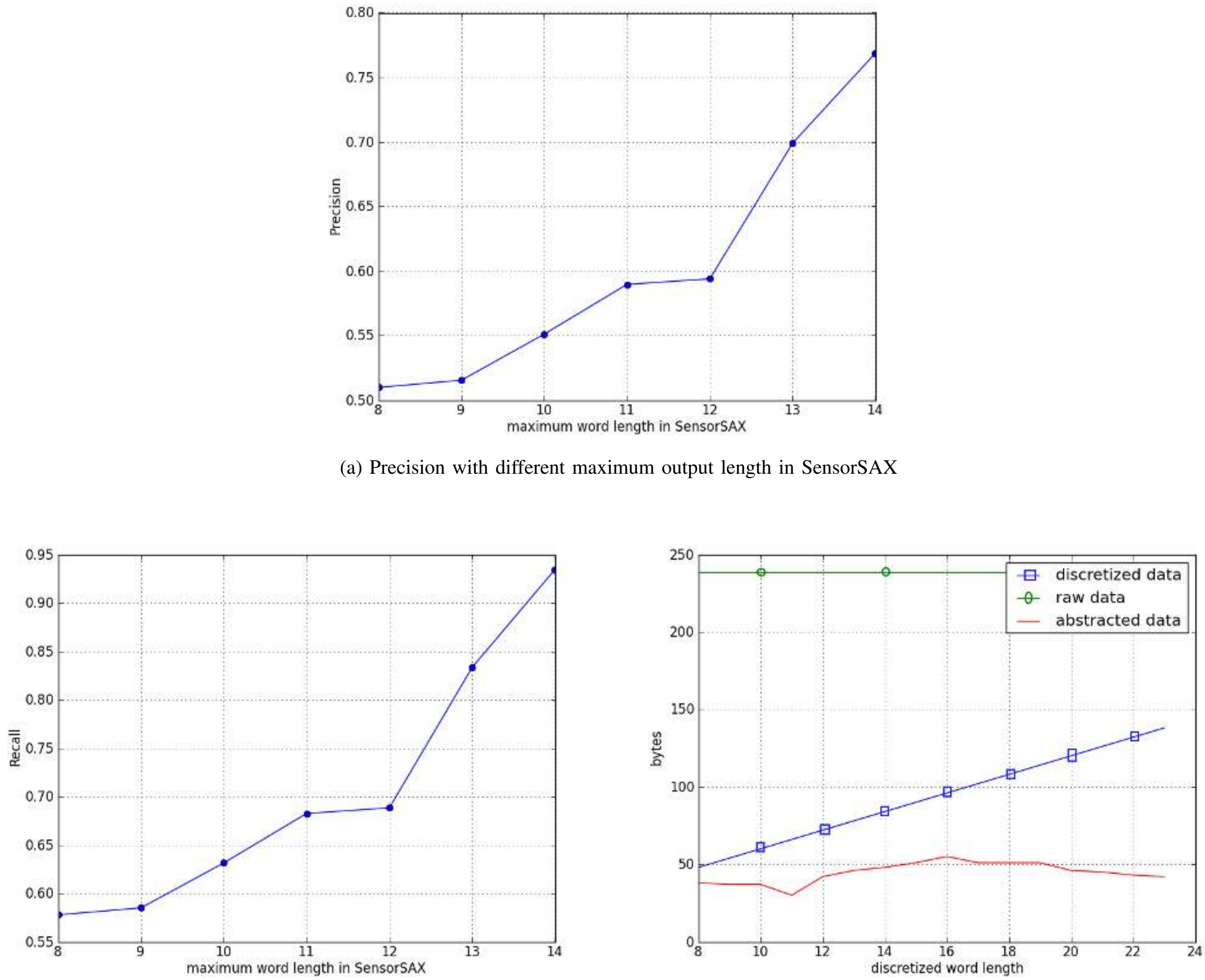
a set of possible abstractions. It is clear that including more data to describe an abstraction will result in a more precise abstraction. To find the temporal relationship, the output of the abductive reasoner is also used to identify the irregular patterns in the temporal model if they do not follow a previously known pattern. This irregular pattern can represent important events that may have occurred. In the current system, the unknown patterns are labelled as ”outliers” and a user can be informed or an action can be defined on the gateway for these outliers. Once an outlier is defined (and labelled) then it can be also included in the temporal model. The initial transition probabilities for the temporal process in our sample data set are shown in Table II (c). This considers temporal relevance of the abstractions and their occurrences (for example in our dataset a high tide abstraction always appear after a low tide abstraction). The HMM reasoning model is able to alter the transition probabilities as more observations are made. We use a training length of 10 days (out of 48 days) to improve the initial manually entered probabilities based on our model (i.e. the initial probabilities are defined manually; however as the system observes more patterns it re-adjusts the probability weights according to the real occurrences).

First we evaluate the correlation and covariance of the discretised and reconstructed data using the SensorSAX algorithm. Figure 10 (b) shows the correlation of the sample data and the reconstructed data over 40 days. As can be seen, the reconstructed data shows high correlation with the original data.

We compare the detected abstractions with the observation data provided by the tidal prediction program. Let’s note that the Channel Costal Observatory collects and communicates this data to compute the observations and in our proposed method the abstractions (which are reduced in size) will be sent from the gateways instead of the raw-data streams. Figure 11 (a) and (b) demonstrate the precision and recall over the coastal data with different maximum word length in the SensorSAX algorithm. A maximum recall result of 0.93 is achieved with a maxium word length of 14. In other words, 93% of the real occurring ”‘high tide”’ events could be found by our model while the data size is reduced by 80%. The precision result is 77% for a maximum word length of 14. There is also always access to the raw data on the gateway, in case more evidence is required.

B. Data Size Reduction

To reduce the communication traffic, gateways can process the data and send the abstractions instead of the raw data. The ”‘size reduction”’ of the abstraction mechanism relies on the dimensionality reduction of the underlying discretisation algorithm. As described in Section IV.A, we use the SensorSAX



(a) Precision with different maximum output length in SensorSAX

(b) Recall with different maximum output length in SensorSAX

(c) Data size reduction between original Data, discretised data and abstracted data

Fig. 11: Evaluation results of abstraction creation

algorithm which transforms continuous sensor readings into string representations. The sample data includes measurements over 48 days with a total of 2317 data points. In Figure 10 (c) the size of data that needs to be communicated using different methods is depicted. The raw data consists of around 246 Kbytes and is unchanged for different maximum word length (i.e. the maxOut parameter in Figure 4) used to discretise the data using the SensorSAX algorithm. The size of discretised data for the same data rises with an average of around 100 Kbytes, if a longer maximum word length is chosen. The size of the abstracted data stays steady with an average of around 49 Kbytes that needs to be communicated from the gateway to other destinations. This leads to a reduction of data by nearly 80% compared with the raw data.

C. Latency

It is important that the abstractions can be inferred in a feasible time. To evaluate the latency of creating the abstractions, the execution time for different solutions is measured. The execution time is determined by the length of the pattern representation defined in the SensorSAX algorithm. The longer pattern representation create more precise abstractions results; however the long patterns in SensorSAX will also imply more data that needs to be processed by the abductive reasoner. We evaluate the latency over the same dataset that is used in the previous section. For smaller pattern representations (i.e. word length < 12) the abstraction creation process takes under 5 seconds (we used a Pentium Quad Core i5 with 2 GHz and 4Gbyte RAM memory). For longer pattern representations, the execution time raises linearly up to 22 seconds for the pattern representation with a length of 23 as shown in Figure 10 (a). In a use case scenario according to latency and accuracy requirement a trade-off needs to be made between the maximum size of patterns and the latency. The current approach is realised as an online algorithm and is executed each time data is requested; however in caching mechanisms of the abstractions need to be also developed to enhance the response time.

D. Discussion

The current work allows to abstract from raw sensor data to interpretable concepts. Software components run on sensor nodes and aggregate and discretise the raw data to low-level SensorSax abstractions. Those abstractions are transmitted to the nearest gateway to abstract it to higher-level abstractions. This requires that the used sensor hardware platform is able to run modified code or adapted implementations. In this work TMote Sky nodes [11] have been utilised. In case that sensor nodes are not open source or proprietary, and our SensorSax module cannot be run on the node itself, the data discretising process can be applied on the gateway level. However this would lead to loss of transmission saving on the south bound direction, as still raw data has to be submitted to the gateway.

Our evaluation results show that we detect 93% of the real occurring events (represented as concept high tide) with a saving of 80% on data communication on the north bound side of the gateway. 23% of the detected abstractions found, were classified as false positives, therefore wrong inferred concepts that did not occur in reality. 7% were classified as false negatives that occurred in reality but where not mapped as an abstraction. The amount of false positives and false negatives is too high that the system can be used for real-time critical scenarios such as flood detection or health observation, but in non-critical scenarios such as Smart Home, Office and other pervasive computing scenarios, and with the huge saving in transmission, the approach is applicable. Especially in the case of false positives, a domain expert is able to examine the raw data that is still stored on gateway/node level, but not transmitted to save traffic, to rule out wrong abstractions.

VI. Conclusions

In this paper, we propose a solution to reduce the communication traffic between sensory devices and applications and services in the Internet of Things and Cyber-Physical systems. We transform the data to higher-level abstractions and transmit these data abstractions instead of raw data. Data abstractions are created using a probabilistic approach where raw data is analysed and based on abductive and temporal reasoning, the abstractions are generated. The abductive and temporal reasoning methods are evaluated over a real world dataset accessed from the UK Channel Coastal Observatory. We show that the occurring events with an accuracy of 93% can be transformed to reduced sized abstractions that are smaller in size by 80% compared to the raw data. The current approach is applied only to one local domain. For global domains and multiple WSNs in different domains, we believe the model still needs to be localised. What can be identified as an outlier in one domain can be normal in a different domain. However, sharing trained local models between different networks can help enhancing the model by learning from other domains or bootstrapping a model for new WSNs. The future work will focus on the parameter learning for the probabilistic model by applying expectation maximisation (EM) algorithms to different local models to increase the accuracy of abstractions.

Acknowledgment

This paper describes work undertaken in the context the EU ICT Iot.est project (www.Ict-Iotest.eu) contract number: 288385. The authors would like to thank Prof. Amit Sheth and Cory Henson for their contributions to the initial PCT-based model.

References

[1] C. IBM. (2012, Aug.) Bringing big data to enterprises. [Online]. Available: http://www-01.ibm.com/software/data/bigdata/

[2] J. C. A. Cisco IBSG, C. H. M. C. Steve Leibson, U. of Michigan, and Fraunhofer. (2012, Aug.) The internet of things infographic. [Online]. Available: http://allthingsd.com/20110714/cisco-reminds-us-once-again-how-big-the-internet-is-and-how-big-its-getting/

[3] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, ser. DMKD '03. New York, NY, USA: ACM, 2003, pp. 2–11. [Online]. Available: http://doi.acm.org/10.1145/882082.882086

[4] E. Keogh, J. Lin, and A. Fu, "Hot sax: efficiently finding the most unusual time series subsequence," in *Data Mining, Fifth IEEE International Conference on*, nov. 2005, p. 8 pp.

[5] Q. Yan, S. Xia, and Y. Shi, "An anomaly detection approach based on symbolic similarity," in *Control and Decision Conference (CCDC), 2010 Chinese*, may 2010, pp. 3003 –3008.

[6] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, pp. 107–144, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10618-007-0064-z

[7] D. Minnen, C. Isbell, M. Essa, and T. Starner, "Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, oct. 2007, pp. 601 –606.

[8] J. A. Reggia and Y. Peng, "Modeling diagnostic reasoning: a summary of parsimonious covering theory," *Computer Methods and Programs in Biomedicine*, vol. 25, no. 2, pp. 125 – 134, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0169260787900484

[9] V. Tsiatsis, A. Gluhak, T. Bauge, F. Montagut, J. Bernat, M. Bauer, C. Villalonga, P. Barnaghi, and S. Krco, "The sensei real world internet architecture," 2010.

[10] F. Ganz, P. Barnaghi, C. Francois, and K. Moessner, "Context-aware management for sensor networks," in *the Fifth International Conference on COMmunication System softWAre and middlewaRE (COMSWARE11), ACM*, 2011.

[11] "Tmote Sky Low Power Wireless Sensor Module," http://sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf, Oct. 2011.

[12] P. Barnaghi, F. Ganz, C. Henson, and A. Sheth, "Computing perception from sensor data," in *Sensors*. IEEE, 2012, pp. 1–4.

[13] Y. Chen, J. Shu, S. Zhang, L. Liu, and L. Sun, "Data fusion in wireless sensor networks," in *Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on*, vol. 2, may 2009, pp. 504 –509.

[14] P. Jesus, C. Baquero, and P. S. Almeida, "A survey of distributed data aggregation algorithms," *The Computing Research Repository ACM*, vol. abs/1110.0725, 2011. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1110.html

[15] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, vol. 2, april 2005, pp. 8 – 13 Vol. 2.

[16] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *INFOCOM, 2012 Proceedings IEEE*, march 2012, pp. 603 –611.

[17] M. Stocker, M. Rönkkö, and M. Kolehmainen, "Making sense of sensor data using ontology: A discussion for residential building monitoring," in *Artificial Intelligence Applications and Innovations*. Springer, 2012, pp. 341–350.

[18] M. Yun, D. Bragg, A. Arora, and H.-A. Choi, "Battle event detection using sensor networks and distributed query processing," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, april 2011, pp. 750 –755.

[19] G. Mulligan, “The 6LoWPAN architecture,” in *Proceedings of the 4th workshop on Embedded networked sensors*, 2007, p. 7882.

[20] ZigBee, “ZigBee specifications,” 2010. [Online]. Available: http://www.zigbee.org/Specifications.aspx

[21] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, “TinyOS: An Operating System for Sensor Networks,” in *Ambient Intelligence*, W. Weber, J. Rabaey, and E. Aarts, Eds. Berlin/Heidelberg: Springer Berlin Heidelberg, 2005, ch. 7, pp. 115–148. [Online]. Available: http://dx.doi.org/10.1007/3-540-27139-2_7

[22] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, 2004, pp. 455–462.

[23] R. B. Smith, “Spotworld and the sun spot,” in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*. IEEE, 2007, pp. 565–566.

[24] F. Ganz, P. Barnaghi, F. Carrez, and K. Moessner, “A mediated gossiping mechanism for large-scale sensor networks,” in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, 2011, pp. 405–409.

[25] I. Syu and S. Lang, “Adapting a diagnostic problem-solving model to information retrieval,” *Information Processing Management*, vol. 36, no. 2, pp. 313 – 330, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457399000370

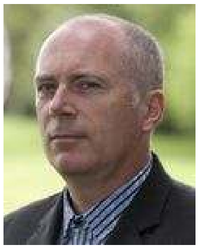
[26] C. Henson, A. Sheth, and K. Thirunarayan, “Semantic perception: Converting sensory observations to abstractions,” *IEEE Internet Computing*, vol. 16, pp. 26–34, 2012.

[27] Y. Peng and J. A. Reggia, “Plausibility of Diagnostic Hypotheses: The Nature of Simplicity,” in *Proceedings of the 5th National Conference on AI (AAAI-86)*, 1986, pp. 140–145. [Online]. Available: http://www.aaai.org/Library/AAAI/1986/aaai86-022.php

[28] N. Jardine and C. van Rijsbergen, “The use of hierarchic clustering in information retrieval,” *Information Storage and Retrieval*, vol. 7, no. 5, pp. 217 – 240, 1971. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0020027171900519

Frieder Ganz is a PhD Student at the Centre for Communication Systems Research at the University of Surrey. His research is focused on information abstraction and extracting machine-interpretable knowledge from large volumes of sensory data using stream processing and machine learning techniques.

Payam Barnaghi is a Lecturer (Assistant Professor) in the Centre for Communication Systems Research (CCSR) at the University of Surrey. His research interests include machine learning, Internet of Things, semantic Web, Web services, information centric networks, and information search and retrieval. He is a senior member of IEEE.



Francois Carrez received a PhD in Theoretical Computer Science from the University of Nancy France in 1991. For 18 years he worked for the Alcatel Research centre in Paris in areas such as Security, Distributed Artificial Intelligence, AdHoc Networking and Semantics. He joined the University of Surrey in 2006 and is currently a Senior Research Fellow at the Centre for Communication Systems Research (CCSR). His research interests include: Semantic Web, Internet of Things, Activity Theory and Social and Behavioural Sciences.