HYDRA: a HYbrid Diagnosis and monitoRing Architecture for diabetes

Özgür Kafalı
Department of Computer Science
Royal Holloway, University of London
Egham, TW20 0EX, UK
Email: ozgur.kafali@rhul.ac.uk

Ulrich Schaechtle
Department of Computer Science
Royal Holloway, University of London
Egham, TW20 0EX, UK
Email: ulrich.schaechtle@cs.rhul.ac.uk

Kostas Stathis
Department of Computer Science
Royal Holloway, University of London
Egham, TW20 0EX, UK
Email: kostas.stathis@rhul.ac.uk

***Abstract*—We present HYDRA: a multi-agent hybrid diagnosis and monitoring architecture that is aimed at helping diabetic patients manage their illness. It makes use of model-based diagnosis techniques, where the model can be developed by two different approaches combined in a novel way. In the first approach, we build the model based on the medical guidelines provided for diabetes. A computational logic agent monitors the patient and provides feedback based on the model whenever the current observations regarding the patient are sufficient to draw a conclusion. In the second approach, we assume a function for the model, and learn its parameters through data. The model is consistently updated via incoming observations about the patients, and allows prediction of possible future values. We describe the components of such an architecture, and how it can integrated into the existing COMMODITY$_{12}$ personal health system. We implement a prototype of HYDRA, and present its workings on a case study on hypoglycemia monitoring. We report our prediction results for this scenario.**

I. INTRODUCTION

Diabetes is defined as a group of metabolic diseases in which a patient has high blood-sugar, either due to the pancreas failing to produce enough insulin, or because cells do not respond to insulin as expected [1]. Two of the most important short-term complications of diabetes are hypoglycemia and hyperglycemia (very low and high blood glucose; respectively), both of which are life-threatening. The management of diabetes is becoming an increasingly important problem worldwide with recent efforts aiming at controlling the aforementioned short-term complications [2], [3], [4]. The normalisation of blood glucose is one of the parameters that must be monitored by a personal health system according to a formal model of the disease (i.e., medical guidelines). This is especially important for *Type 1* diabetic patients, who need to have their glucose levels monitored continuously.

In this work, we are motivated by our participation in the COMMODITY$_{12}$ personal health system, which focuses on giving diabetic patients medical advice to maintain their illness through the use of sensors and artificial intelligence agents [5]. Currently, we measure glucose levels through the use of two different sensors: (i) the finger prick method via the GlucoTel sensor[1] suitable for everyday use, and (ii) the continuous glucose monitor CGM[2] to be implanted on the side of the abdomen and worn for 3 to 5 days continuously. In the near future, continuous monitoring of glucose will also be relevant for *Type 2* diabetic patients as new technology becomes available to effortlessly measure blood glucose (e.g., smart contact lenses).

Currently, glucose monitoring in COMMODITY$_{12}$ is performed based on expert systems like logic-based rules, that represent recent medical expertise in diabetes management. The measured glucose value is fed into the system, and if any of the rules fire (e.g., a hypoglycemia situation where the glucose value is less than 3 $mmol/l$), then an alert is created. While this is very helpful for the patient, a personal health system should be able to assist in situations where the rules cannot make a conclusive decision. For example, if the measured value is 3.1, the rules would assume everything is under control. However, due to some noise in the sensor, the value might be misread and the patient's situation might get worse if we wait for another measurement. Our motivation in this paper is to deal with such situations, and try to predict whether a measured value of 3.1 may lead to a hypoglycemia.

Accordingly, we propose a multi-agent hybrid diagnosis and monitoring architecture for diabetes called HYDRA. We build upon existing work on COMMODITY$_{12}$, and extend its current diagnosis architecture with machine learning based prediction. We extend the guidelines to allow more customised and flexible management of patient's glucose values by giving advice in situations which are not considered critical. We present a novel way of combining the two model-based approaches, in which the model is described in a different manner. In the current version of COMMODITY$_{12}$, the model is represented by a logic based system that reflects the medical guidelines for diabetes. In the proposed extension, the parameters of the model are rather learnt through data and consistently updated (customised) with data regarding each patient. We present the details of such a hybrid architecture. Moreover, we implement a prototype of HYDRA using the GOLEM agent platform [6], and present a case study on hypoglycemia monitoring. We show that our system achieves promising prediction results on synthetic data.

The rest of the paper is structured as follows. Section II reviews the COMMODITY$_{12}$ architecture and related work on blood glucose prediction. Then, Section III describes our proposal to extend the current diagnosis system with a prediction component. Section IV presents a case study using continuous glucose data and shows our results. Finally, Section V concludes the paper and presents possible future directions.

[1] `http://www.bodytel.com`

[2] `http://www.medtronicdiabetes.com/products/guardiancgm`

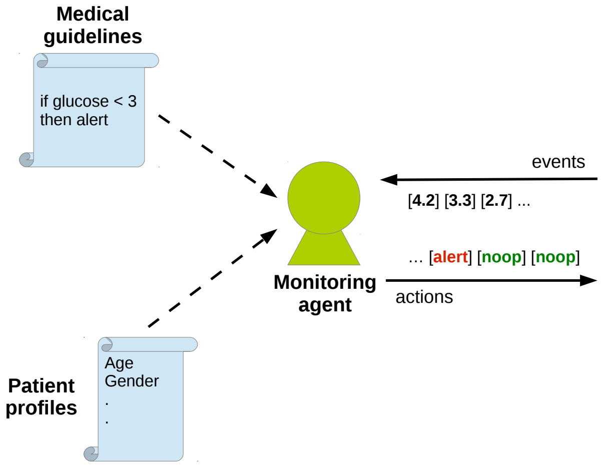

Fig. 1. Current architecture of the COMMODITY$_{12}$ PHS. The monitoring agent makes a decision based on the medical guidelines and the current situation of the patient.

II. Background & Related Work

A. *COMMODITY$_{12}$ Personal Health System*

Figure 1 describes the current diagnosis architecture of COMMODITY$_{12}$. In this paper, we focus on the workings of the monitoring agent, rather than describing every component of the system (e.g., sensors).

```
select(monitor_glucose, alert(Patient, hypoglycemia), T):-
       holds_at(glucose(Patient)=Value, T),
       Value < 3.
```

Listing 1. Rule for sending a hypoglycemia alert. The rule is given in Prolog syntax and can be read as follows: while trying to monitor glucose, select the action to alert the patient of a hypoglycemic attack if the patient's current glucose value is less than 3. The predicate holds_at/2 is implemented in the Event Calculus [7], a logic-based formalism that is used to handle temporal conditions in the body of the rules.

The monitoring agent uses computational logic to decide an action that reflects the current situation of the patient, e.g., send an alert if recent glucose levels are low (hypoglycemia). It uses medical guidelines represented as logic rules and any relevant profile information about the patient's medical background. Listing 1 shows the rule for selecting a hypoglycemia alert. Glucose measurements are received by the agent in the form of observation events. When a new observation event comes, the agent checks the corresponding glucose value and produces an alert if the value is below the certain threshold.

B. *Glucose Prediction*

For prediction of continuous signals such as blood glucose, we combine ideas from fields such as data-driven model-based diagnosis and time series analysis. Flesch and Lucas map the process of model-based diagnosis to the formalisation of Bayesian networks [8]. This work is related to the proposal presented in this paper as one can read (switching) Kalman filters as dynamic Bayesian networks. They describe static boolean models only, whereas we allow also time series and hybrid models of continuous and discrete valued data. There is a number of studies that deploy data-driven models and machine learning for prediction, diagnosis, clinical decision making and other health-care related topics. Recently, diverse uni-variate and multi-variate models that are related to Kalman filters have been used in continuous glucose monitoring and hypoglycemia prediction [9], [10], [3]. They differ from our proposal since they consider fixed parameters of variables (i.e., chosen by an expert investigator).

Knobbe and Buckingham apply the extended Kalman filters for continuous glucose monitoring [2]. Here, the standard Kalman filters equations are extended with a non-linear process function to capture non-linear dynamics in the data. This can lead to higher performance, but the structure of this non-linear function has to be determined via trial and error and fine-tuning by the researcher. An online-learning method was proposed by Eren *et al.* [11]. They deploy a recursive least square method to learn an extended autoregressive moving average model, taking into account exogenous input within a multi-variate manner. This exogenous input is unknown until new variables become available. Taking additional variables for their dynamic system shows similarity with our work. However, it is hard to establish a connection to model-based diagnosis in this context, because there is no latent root in the topology of the underlying model that can be utilised as diagnostic input variable. Turksoy *et al.* have used Kalman filters and SavitzkyGolay filter for denoising and trained an autoregressive moving average model with some background information for prediction, which leads to early alarm of hypoglycemia [4].

III. Hydra Architecture

A. *Multi-agent Architecture*

Figure 2 describes how we propose to extend the current diagnosis architecture of COMMODITY$_{12}$. First, we focus on the right part of the figure and introduce our prediction agent. The prediction agent relies on a prediction model, where the parameters of the model are learnt from the training data stored in the patient database. Once the model is trained, the agent is ready to make predictions about future glucose values.

Now, we come back to our monitoring agent and describe how we have extended its rule-base to enable the prediction capability. The rule given in Listing 1 is still valid, and any glucose measurement that is considered *critical* is corresponded with an alert. In addition, the monitoring agent decides whether to request a prediction from the prediction agent. Listing 2 shows the rule for deciding this prediction request. When this request is received, the prediction agent responds with the next glucose value that is predicted based on the learnt model.

```
select(monitor_glucose,ask(Ag,glucose(Patient)=NextVal),T):-
       predictor(Ag),
       holds_at(glucose(Patient)=Value,T),
       Value >= 3,
       Value < 4.
```

Listing 2. Rule for requesting a prediction.

Note that the monitoring agent sends every received observation to the prediction agent together with any relevant profile information (e.g. the patient makes excessive use of

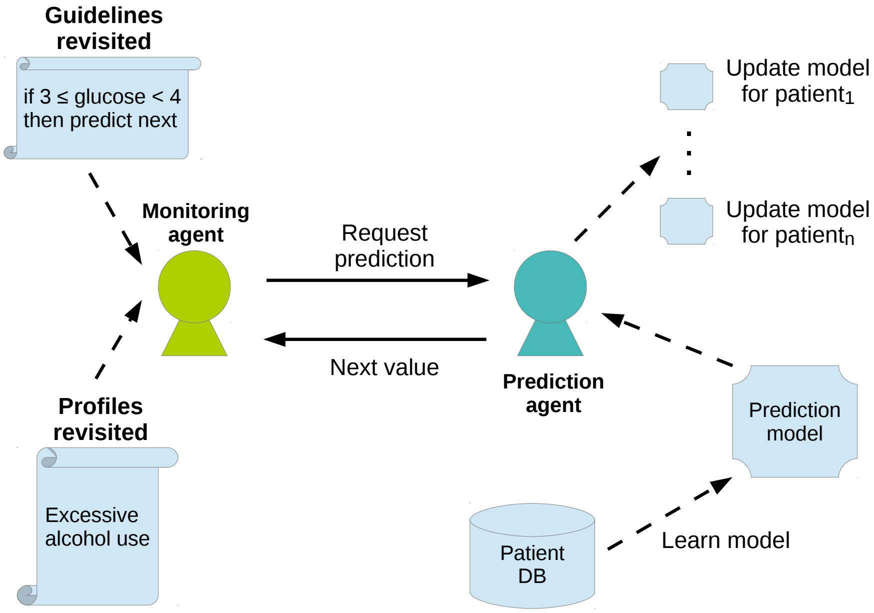

Fig. 2. Proposed extension with the addition of a prediction agent. The prediction agent learns the model parameters from a training dataset. Every incoming value from the monitoring agent about a patient is used to update the model specific to that patient.

alcohol or works evening shifts). These are used in turn to update the prediction model and customise it for that specific patient in order to increase prediction performance.

B. Prediction Component

For the prediction component, we deploy a framework for reasoning about observations over time by building upon the existing state-of-the-art in time-series analysis and machine learning techniques for prediction. Note that prediction should be read in a temporal sense: we want to predict future values of glucose. The component is motivated by ideas of the classical model-based diagnosis literature. The classical definition of model-based diagnosis entails the identification of causes for a set of symptoms [12], given a description (e.g. a logical theory) of a system comprised of components and some observations [13]. The complexity of the human organism prohibits us from providing such a description of every single process underlying a symptom or an observation, that is a measured glucose value. We need to model some of the underlying physical processes and the affected components without any knowledge about them. In machine learning, such components are called hidden or latent [14]. The method should also be able to take input from a logical theory. If the rule-based model used by the monitoring agent diagnoses a condition which affects the observed glucose values, then the prediction model should take this information into account.

We can thus summarise three requirements for our prediction model. It should support:

- prediction of future glucose values;
- hidden components;
- a hierarchical structure, with a variable on top that is either Boolean or categorical.

These requirements define our hierarchical hidden structure. This hidden structure has a categorical (discrete-valued) hidden variable at the top of the hierarchy. It models the nature and cause of a series of observations, which is what we want to detect with model-based diagnosis. ‘Hidden’ does not mean we are inhibited from diagnosing a cause by a doctor or with the logic based component - we simply assume it to be hidden during the training of the model and add the information on the hidden state for prediction if necessary. We can interpret the entire hidden structure as the underlying process which cannot be observed but which is generating measurable and therefore observable outcomes.

To reflect the state-of-the-art in glucose management, we choose a switching Kalman filter (SKF) [15], because it subsumes models used in the current machine learning-based glucose management approaches (for example see [10], [3], [16]). Furthermore, its model structure provides the desirable hidden properties described in the requirements above (see Figure 3). It interfaces with model-based diagnosis in the sense that we replace the logical theory representing the domain by a machine learning approach that relies on a statistical theory applied in the same domain. For example, imagine two patients both suffering from Diabetes type 2. One is diagnosed with alcoholism and one is not. Here, we can assume a two-state switch for our Kalman filter model. ($alcoholism = \{true, false\}$). This is our statistical theory. We can also imagine more than one variable involved leading to multiple states possible instead of two.

We deploy the SKF in two stages. In an offline training stage, we learn the parameters of the model that define its

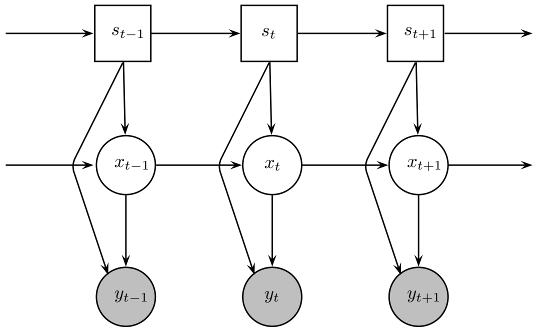

Fig. 3. Graphical representation of a switching Kalman filter. Hidden variables are depicted with no filling, observed ones are grey. We sketch discrete-valued variables as square nodes and continuous-valued ones as circular nodes. The origin of the data generating model is a discrete state variable $\mathbf{s}$. For example, if the state variable $\mathbf{s}$ is some co-morbidity of diabetes at time t, we would expect the observations to differ depending on whether or not a patient is suffering from this co-morbidity.

linear transformations and error distributions given a set of training data. Then, in an online stage, we update the hidden variables using the process model with learned parameters. A simple linear transformation provides us with a Gaussian mixture probability distribution of next observation. With a technique called moment matching we then compute the next observation with maximum likelihood. The details of the above stages we omit, but we refer the interested reader to the work presented by Murphy in [15].

C. Implementation

HYDRA requires that a monitoring agent and a predictor agent communicate with each other and the user using a request-reply protocol. The monitoring agent must be developed using declarative logic-based tools and techniques such as those characteristic of Prolog development environments. On the other hand, the prediction agent requires mathematical problem solving and number crunching to implement the techniques required behind SKF, which can be suitably supported with either Java or Matlab technologies.

To support the HYDRA requirements above, we have implemented our prototype using the GOLEM agent platform [6]. GOLEM is a logic-based agent platform developed to support multi-agent system environments that evolve over time. It provides a middleware that agent developers can use to build multi-agent systems using either Java and Prolog, which meets our requirements. The platform also allows the development of objects, that can wrap different types of entities, from databases to external functionalities provided by external components like Matlab.

Using GOLEM, the reasoning component of the monitoring agent is programmed using Prolog tools and techniques, while all other (low-level) functionality throughout the system (e.g., communication between agents, processing of measurement data) is implemented in Java. The actual prediction component is implemented in Matlab and wrapped as a Java object that is used by the prediction agent. The prediction component keeps a separate copy of the prediction model for each patient, and acts as a service to respond queries from the monitoring agent. GOLEM allows the monitoring and prediction agents to be run on different physical machines to improve performance and allow scalability. For the sake of experimentation, the prediction component can serve one request at a time (i.e., a single thread) in the current prototype.

IV. CASE STUDY: HYPOGLYCEMIA MONITORING

In this case study, we present the workings of the HYDRA architecture on continuous glucose monitoring. More specifically, we are looking for cases of a medical emergency regarding the blood glucose level of the patient known as *hypoglycemia*. Hypoglycemia is formally described by the *International Classification of Diseases* as the patient's glucose level being below a certain threshold value. It can produce a variety of symptoms and effects but the principal problems arise from an inadequate supply of glucose to the brain, resulting in impairment of function. Effects can range from mild dysphoria to more serious issues such as seizures, unconsciousness, and (rarely) permanent brain damage or death. According to the severity level of hypoglycemia, a series of actions may need to be taken immediately, including alerting the doctor of the patient.

Conventional glucose monitoring usually relies on guidelines that provide specific thresholds and constraints for hypoglycemia. If we only implemented the rules that represent the guidelines literally, then if a patient's glucose measurement is slightly greater than 3 $mmol/l$, then no alert should be sent to the user. Still, for some doctors this would still be an alarming measurement. We therefore extend the conventional monitoring of hypoglycemia with the addition of our prediction component. In particular, we aim at providing the patient and medical personnel alike with additional information, even though the current glucose level of the patient is not at a critical level. Our system continuously reads glucose data as they become available, and takes actions based on the patient's current measurement as well as updating the prediction model specific to that patient.

To test our prediction agent and the HYDRA architecture, we work with synthetic data generated with auto-regressive models. The justification behind this choice is two-fold: (i) we are not aware of any open data set providing continuous glucose data suitable for our system, (ii) we are currently performing clinical trials on patients in COMMODITY$_{12}$, which will lead to data from a continuous glucose measurement device. Whenever the measuring device is activated, we will be able to read glucose measurements every five minutes from diabetic patients. However, this data is not yet available in the COMMODITY$_{12}$ project.

Listing 3 shows sample output of HYDRA for each possible outcome of monitoring. Note that the glucose values shown are not from consecutive measurements. When the glucose measurement is above 4 $mmol/l$, the system takes no action as we are only dealing with cases of hypoglycemia here. When the glucose level drops below 4 $mmol/l$ (but still not at a critical level), the system makes a prediction for the next possible value based on the customised prediction model for the patient. Finally, if the glucose level is critical (i.e., $<$ 3 $mmol/l$), the system sends out an alert.

Figure 4 illustrates a period of time in the data where our approach becomes useful. We start with an actual glucose

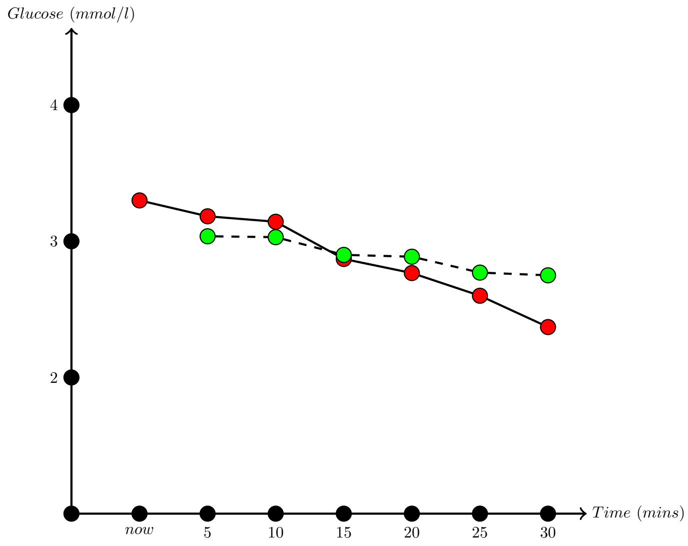

Fig. 4. Glucose level prediction for the next 30 min based on the currently measured value. Straight line shows the actual measured values (red circles), whereas the dashed line shows the predicted values (green circles).

measurement of 3.3 $mmol/l$, and predict the glucose trend of the patient for the next half an hour (i.e., a total of six values for every five minutes). Note that the straight line shows the actual trend of the patient's glucose level, which is going towards a hypoglycemia situation in the next 15 minutes. The dashed line shows our prediction for the next 30 minutes, and it successfully catches the patient's glucose trend for the given period. Moreover, the first point in time where the actual hypoglycemia occurs is captured in our prediction (the 15 minute mark). Therefore, we can use this information to alert the patient 15 minutes before the actual hypoglycemia occurs.

```
MA receives observation: obs(glucose,4.28)
   MA sends request for model update
   <<< NO ACTION >>>
   PA updates model

MA receives observation: obs(glucose,3.30)
   MA sends request for model update
   MA calls predictor
   PA sends prediction
   PA updates model
   <<< PREDICTION 3.03 >>>

MA receives observation: obs(glucose,2.78)
   MA sends request for model update
   <<< HYPO ALERT >>>
   PA updates model
```

Listing 3. Sample output of HYDRA. MA and PA stand for the monitoring and prediction agents; respectively.

Prediction Performance

We test the predictive component by assuming three different states and corresponding time series. We create each of these time series using auto-regressive models. Auto-regressive models are indeed a good choice, since some instances of this model class have been declared suitable to model glucose curves [16]. For illustrating and testing the inference and learning capabilities of the switching Kalman filters algorithm, we implement three second-order auto-regressive models:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t \qquad (1)$$

where y_t are the observations we make at time t and e_t is a zero-mean Gaussian noise/error term:

$$e_t \sim \mathcal{N}(0, \sigma). \qquad (2)$$

The auto-regressive models differ with regards to the choice of the parameters $\theta = \{\phi_1, \phi_2, \sigma\}$ (where we assume for each $\mathbf{s}_i$ a different set θ). Here, we assume that each different time series was "created" by a different underlying state s. We form one combined time series by connecting these three separate time series. The combined time series has a length of 600 points in time (200 time points per auto-regressive model). We try to initialise these parameters in a way that they resemble glucose values as closely as possible.

To provide an initial performance evaluation, we assume that the monitoring agent provides some profile information on the current state of the patient that produces the observations. Then, we create the next observation at $T+1 = 601$ given the true generative system that creates our data. We compare this actual value with the value predicted by our model.

We run the above procedure 80 times resulting in a root means square error of 0.02 $mmol/l$ for predicting the next observation. This would correspond to five minutes ahead of time with real continuous glucose data. Note that this very promising result needs to be considered in the following context. Firstly, a five minute prediction range is not very large compared to the standard glucose management literature. However, given the way the component is integrated in the agent architecture this is not of great importance. Secondly, the result is based on synthetic values and needs to be verified with real world values as well.

V. Discussion

We have presented HYDRA: an agent-based hybrid diagnosis and monitoring architecture for diabetes. The monitoring agent uses computational logic to draw conclusions based on patient's current situation. When the observed data is not adequate to make a decision (using the conventional guidelines), the prediction agent is consulted to come up with anticipated possible future observations. Such predicted observations make use of state-of-the-art machine learning techniques using switching Kalman filters.

We have developed a prototype of the system in the GOLEM agent platform, making use of the advantages provided by different programming paradigms supported: the low-level functionality of the system is implemented procedurally (in Java) to support efficiency and easy integration with a distributed environment, the monitoring agent is implemented declaratively (in Prolog) to support explanation to the user, and the prediction agent is implemented in Matlab to provide ease of the mathematical processing expected. We have tested the system on continuous glucose data, and presented our prediction results.

We have also shown how our system can be used to support decisions on observations (measurements) that are borderline. In such situations, we rely on the use of the predictor agent to report on any trends of future measurements that may have undesirable effects with the patient's monitoring. Such a feature can complement the conclusions from guidelines that may be deemed to be too "black and white", as for example we saw with the situation where the glucose level is $3.3\ mmol/l$. Such a predictive capability can be very useful to doctors who might want to complement their decisions on these borderline observations with evidence from existing patient data.

There are several diagnosis / prediction architectures in the literature that deal with medical data. Recently, Lu *et al.* implemented a system consisting of wavelet-transform and a hidden Markov model for diagnosing heart diseases [17]. They apply state of the art de-noising in context of a dynamical system. However, they do not connect their work with classical model-based diagnosis approaches as we do here.

The problem of medication adherence in type 2 diabetes patients has been tackled with data-driven methods [18]. Here, evidence-based decision support systems were enhanced by machine learning techniques. The authors cluster the data into a mixture of Gaussians. Clusters were related to patient's demographic and treatment characteristics. Then, they learn logistic regression functions and artificial neural networks to classify adherence and non-adherence behaviour. In contrast, we combine the two steps in one, since switching Kalman filters can be seen as a moving Gaussians over time.

Oberkampf and colleagues include different sources of knowledge into a diagnostic method for ranking the likelihood of given diseases [19]. Similar to the work presented here, the ranking framework starts from a symbolic logic point of view motivated by logical knowledge openly available such as SNOMED. However, their probabilistic component is defined in terms of a weighted logic and do not include state-of-the-art machine learning methods as we do here.

We plan to extend work presented in this paper in the following directions. We will evaluate the prediction component with real world data in the COMMODITY$_{12}$ project and we will extend the monitoring to other continuous signals such as heart rate. Rather than using predefined threshold values for initiating a prediction (e.g., between 3 and 4), we will learn these values which might vary from patient to patient. We believe that this will lead to better efficiency for the overall system. We will also develop a web-based GUI for HYDRA and allow users to train models with their own data. We will further revise the prototype, in particular, the prediction component as a multi-threaded service, to make it scalable.

ACKNOWLEDGEMENTS

This work is supported by the EU project COMMODITY$_{12}$ (www.commodity12.eu). We thank Alfonso Romero for helping with the implementation.

REFERENCES

[1] K. Alberti and P. Zimmet, "Definition, diagnosis and classification of diabetes mellitus and its complications. Part 1: diagnosis and classification of diabetes mellitus. Provisional report of a WHO Consultation," *Diabet. Med.*, vol. 15, no. 7, pp. 539–553, 1998.

[2] E. J. Knobbe and B. Buckingham, "The extended kalman filter for continuous glucose monitoring," *Diabetes technology & therapeutics*, vol. 7, no. 1, pp. 15–27, 2005.

[3] C. C. Palerm and B. W. Bequette, "Hypoglycemia detection and prediction using continuous glucose monitoringa study on hypoglycemic clamp data," *J Diabetes Sci Technol*, vol. 1, no. 5, pp. 624–629, 2007.

[4] K. Turksoy, E. S. Bayrak, L. Quinn, E. Littlejohn, D. K. Rollins, and A. Cinar, "Hypoglycemia early alarm systems based on multivariable models," *Industrial & Engineering Chemistry Research*, 2013.

[5] Ö. Kafalı, S. Bromuri, M. Sindlar, T. van der Weide, E. Aguilar-Pelaez, U. Schaechtle, B. Alves, D. Zufferey, E. Rodrguez-Villegas, M. I. Schumacher, and K. Stathis, "COMMODITY$_{12}$: A smart e-health environment for diabetes management," *Journal of Ambient Intelligence and Smart Environments, IOS Press*, vol. 5, no. 5, pp. 479–502, 2013.

[6] S. Bromuri and K. Stathis, "Situating cognitive agents in golem," *Engineering environment-mediated multi-agent systems*, pp. 115–134, 2008.

[7] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Generation Computing*, vol. 4, no. 1, pp. 67–95, 1986.

[8] I. Flesch and P. Lucas, "The probabilistic interpretation of model-based diagnosis," *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 204–215, 2009.

[9] A. Facchinetti, G. Sparacino, E. Trifoglio, and C. Cobelli, "A new index to optimally design and compare continuous glucose monitoring glucose prediction algorithms," *Diabetes technology & therapeutics*, vol. 13, no. 2, pp. 111–119, 2011.

[10] B. W. Bequette, "Optimal estimation applications to continuous glucose monitoring," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 1. IEEE, 2004, pp. 958–962.

[11] M. Eren-Oruklu, A. Cinar, D. K. Rollins, and L. Quinn, "Adaptive system identification for estimating future glucose concentrations and hypoglycemia alarms," *Automatica*, vol. 48, no. 8, pp. 1892–1897, 2012.

[12] M. R. Genesereth, "The use of design descriptions in automated diagnosis," *Artificial Intelligence*, vol. 24, no. 1-3, pp. 411–436, 1984.

[13] R. Reiter, "A theory of diagnosis from first principles," *Artificial intelligence*, vol. 32, no. 1, pp. 57–95, 1987.

[14] C. M. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006, vol. 4.

[15] K. P. Murphy, "Switching kalman filters," Citeseer, Tech. Rep., 1998.

[16] G. Sparacino, F. Zanderigo, S. Corazza, A. Maran, A. Facchinetti, and C. Cobelli, "Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-series," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 5, pp. 931–937, 2007.

[17] X. Lu, Y. Lin, and J. Fan, "Electrocardiography recognition based on wavelet filter and em algorithms," in *International Workshop on Artificial Intelligence and NetMedicine*, 2013, p. 31.

[18] R. Bahati, F. Gwadry-Sridhar, and F. Gwadry, "Towards evidence-based decision support systems," in *International Workshop on Artificial Intelligence and NetMedicine*, 2013, p. 51.

[19] H. Oberkampf, S. Zillner, B. Bauer, and M. Hammon, "Towards a ranking of likely diseases in terms of precision and recall," in *International Workshop on Artificial Intelligence and NetMedicine*, 2012, p. 11.