A Hybrid Artificial Intelligence System for Assistance in Remote Monitoring of Heart Patients

Theodor Heinze, Robert Wierschke, Alexander Schacht and Martin v. Löwis

Hasso-Plattner-Institut, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{Theodor.Heinze, Robert.Wierschke , Alexander.Schacht, Martin.vonLoewis }
@hpi.uni-potsdam.de

Abstract. Advancements in the development of medical apparatuses and in the ubiquitous availability of data networks make it possible to equip more patients with telemonitoring devices. As a consequence, interpreting the collected data becomes an increasing challenge. Medical observations traditionally have been interpreted in two competing ways: using established theories in a rule-based manner, and statistically (possibly leading to new theories). In this paper, we study a hybrid approach that allows both evaluation of a fixed set of rules as well as machine learning to coexist. We reason that this hybrid approach helps to increase the level of trust that doctors have in our system, by reducing the risk of false negatives.

1 Introduction

With the changing demographics in Germany, the number of elderly people that the health care system must support increases. Furthermore, Germany faces the problem that specialists such as cardiologists move away from rural areas requiring patients from those areas to travel long distances to see them. Therefore, even a simple check up involves significant effort. In the context of the Fontane [1] project, we plan to use mobile communication technology to develop new solutions that overcome these limitations. The main idea is to equip patients that require regular observation of their vital signs, e.g. patients with heart diseases, with sensors that collect medically relevant information. Such sensors include scales and blood pressure monitors, a simple self-evaluation of a patient's wellbeing as well as more complex sensors such as electrocardiogram recorders. The measurement data is collected by a special home broker device that transmits the data to a telemedicine center (TMC). In the telemedicine center the patient's vital sign will be analyzed by medical experts. Furthermore, the data is available to other persons involved in the patient's care, e.g. the family doctor. Thus, patients can be monitored even in rural areas without the need to travel long distances or even be hospitalized. This not only saves time and money, but also improves the health care itself and provides the patient with the comfort of their own home. The Fontane monitoring scenario is depicted in Figure 1.

Investing in **your** future!
EUROPEAN UNION
European Regional Development Fund
www.efre.brandenburg.de

The project on which this report is based was funded by the Ministry of Economy of Brandenburg and the EU under grant No. 80136574. This publication reflects the views only of the author.

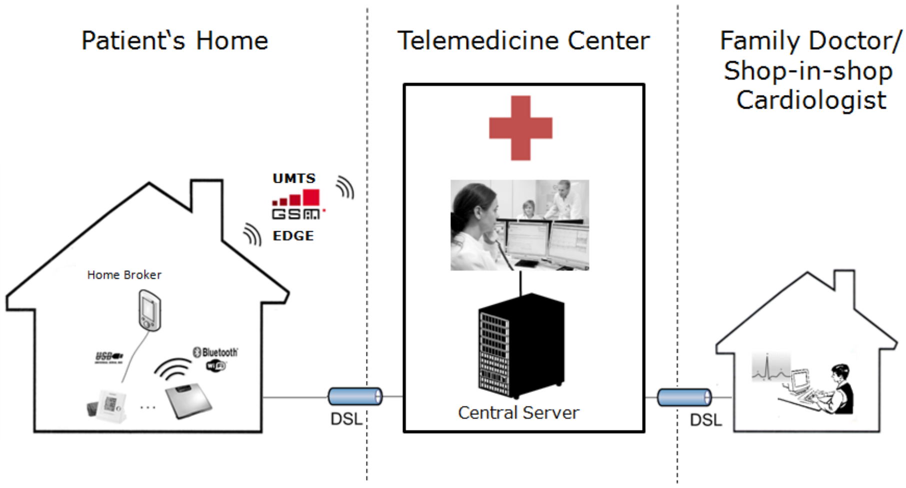

Fig. 1. Fontane Architecture.

In the context of Fontane, we expect a ratio from 5-10 doctors for about 1000-5000 patients to be monitored during three medical studies. In order to cope with the raising number of patients that require telemedical monitoring, the measurement data needs to be prioritized. We are currently developing a self-adaptive, prioritizing middleware (SaPiMa) that integrates hybrid artificial intelligence techniques [8] in order to suggest a review order of the patient data. Additionally, the middleware uses the decisions made by the doctors to improve its suggestions. Hence, more patients can be monitored while patients with critical conditions are noticed early. The main contribution of this paper is the introduction of an architecture of a hybrid artificial intelligence system for telemonitoring of heart patients.

2 Architecture

In the Fontane scenario described in Section I, the patient uses a number of different measurement devices. The types of the devices (e.g. blood pressure monitor, ECG recorder) vary depending on the therapy, which defines the measurement plan. After completion of the measurement the devices transmit the data to the home broker via Bluetooth. The home broker acts as the central hub in the patient's home. Upon receiving the measurements from the various devices, the home broker converts the device specific data formats and transmits the data to the telemedicine center using mobile networks. On the side of the telemedicine center, the incoming data will be stored in an electronic health record (EHR). In order to provide interoperability with various EHR systems and medical data standards, the measurements are processed by a J2EE-based SaPiMa Endpoint (see Figure 2). The SaPiMa Endpoint uses the Apache Camel framework [2]. Camel allows the declarative configuration of routes and processors that are used to handle messages. Due to its declarative approach the system can be easily reconfigured and extended. The SaPiMa Endpoint converts the measurement according to the required record format. Additionally, the measurements are processed by a prioritization engine. According to the calculated priority, the SaPiMa system will suggest a review order for the patients. Furthermore, SaPiMa will learn from the actions of the doctors to improve the prioritization in the future. Due to the prioritization, critical changes in the health status of a patient can be noticed early which increases the survivability of the patient.

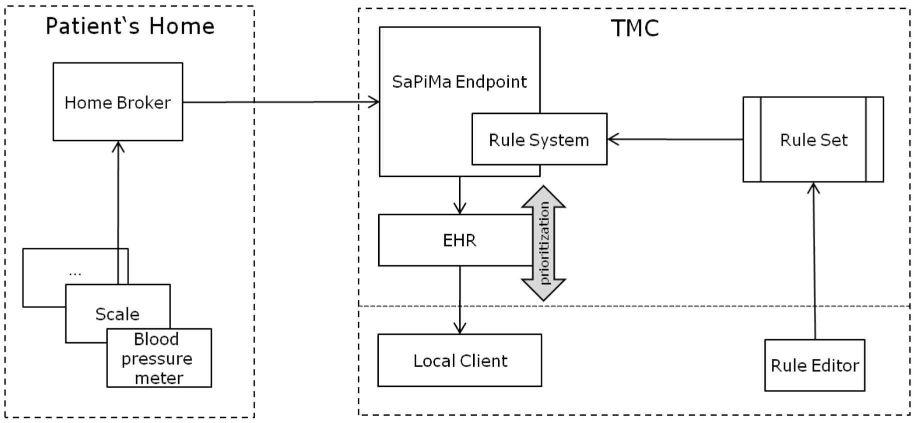

Fig. 2. SaPiMa testbed.

We have set up the system depicted in Figure 2 in a testbed for further research and added a neural network which, together with the rule system, builds a hybrid prioritizer. The following sections explain in detail the implementation of our hybrid artificial intelligence prioritization engine.

3 Concept of a Hybrid Artificial Intelligence Classifier

3.1 Challenges

In order to provide a prioritization as described above, SaPiMa makes high demands on the utilized classifier. Vital parameters, such as blood pressure and body weight are transmitted according to a measurement schedule. In the TMC, SaPiMa calculates a priority based on the measurement values. Thus, a review order for the patients is suggested in order to assist the medical experts. To increase the prioritization accuracy, not only current data but also historical and master data are taken into account. However, the data available for the prioritization differs depending on patient and situation. Hence, the prioritization algorithm must be flexible enough to deal with variable input vectors.

Furthermore, the prioritization mechanism must be able to interpret the different values correctly, e.g. distinguish a body weight from a pulse rate value. This can be achieved by using a classification system. In the context of the Fontane project we augment each measurement value with a LOINC code [3].

Another important aspect is the statistical distribution of the aberrant measurement values. According to oral reports of experienced telemedicine cardiologists, only about 1-5% of all patient data reviews result in an intervention. In the majority of cases, no further action is taken. This distribution poses a particular challenge for a statistical priority estimator because it obtains a >90% success ratio simply by guessing the same every time. The training process of a classifier like a multilayer perceptron would be, without further modifications, very susceptible to terminating the training process at that point.

At the same time, this 1-5% of the patients requires a reliable monitoring due to their state of health. The goal is to eliminate the false negative error (sick patient considered as healthy), at the cost of the false positive error (healthy patient considered as sick). While a classifier with a high false negative error rate might lead to worsening of healthcare provision, a high false positive error rate makes the system less efficient with regard to working time of the telemedicine center personnel.

Another challenge is the explanation system. Due to legal and ethical reasons the ability to explain its decisions would be a great benefit for the prioritization mechanism. For a rule-based system, decision explanation would be very easy by just listing the set of matching rules, whereas statistical classifiers give results that are difficult to explain automatically [6]. On the other hand, rule-based systems are, as a general rule, not as efficient as probabilistic ones: it is tedious to construct them manually, and they grow significantly in size if derived automatically.

In summary, neither a rule-base nor a static classifier satisfies the requirements for remote patient monitoring. For our middleware, we developed a hybrid classification system which combines both approaches in order to overcome their shortcomings.

3.2 Concept of a Hybrid Classifier

Since both approaches offer numerous advantages, we propose a classification system which combines both a rule-based and a statistical classifier thus stacking some benefits and eliminating some shortcomings. The goal is to achive the precision and flexibility of a statistical classifier as well as the controllability of a rule-based one.

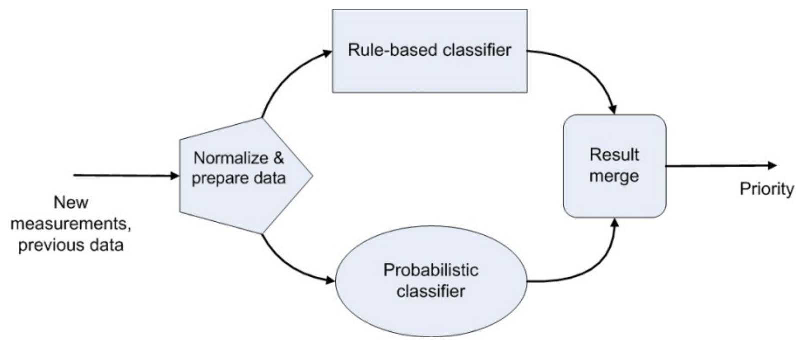

Fig. 3. Hybrid Classifier.

Figure 3 shows the conceptual design of the system. The incoming data consists of current and historical measurements sorted by LOINC-Codes. The data is passed to both the rule-based and the statistical classifier. Both then determine a prioritization value. The output of the statistical classifier is a very exact estimation of the abnormality of the given measurement data. It is a real number in the range between zero (= absolutely normal) and 1 (= totally abnormal). The statistical classifier uses machine learning techniques to learn from former classification decisions met by the doctor to learn to fully imitate his decision behavior. There is also a strong focus on avoiding false negative errors. Currently, an artificial neural network (feed-forward MLP with backpropagation) with loss matrix is used.

The output of the rule-based classifier is a Boolean value which is zero (normal) or 1 (abnormal). The rule-based classifier relies on a set of rules created and maintained by the doctors. This allows to influence the classification process and adapt the prioritization to patient-specific cases that the statistical classifier does not recognize for some reason.

The overall output of the system is computed as follows:

```
output = maximum(output_statistical, output_rule_based)
```

We hope that the statistical classifier produces reliable results in the majority of the cases, where the rule-based one deals with outliers.

4 Rule-Based Classifier

The core of a rule-based or manual classifier is a set of rules. These rules must be created and managed by domain experts (physicians or study nurses). Therefore, we developed a domain-specific language (DSL) that empowers the medical experts to specify the rule set using a familiar terminology. The language is based on predicate logic [4] and allows the specification of generic as well as patient-specific thresholds.

Additionally, we developed an editor to edit the rule set easily as possible. The process of classification starts with creating or editing the rule set with the aid of the described editor. After that, the rule set is exported into a rule interpreter. This interpreter classifies the measured values on the basis of the rule set. The language and its automatic generated editor are implemented using Xtext [5].

5 Probabilistic Classifier

As described in the challenges chapter, the prioritization may not produce false negative errors for the sake of healthcare on the one hand and should on the other hand keep the false positive error rate as low as possible to reduce the amount of time doctors spend reviewing patient measurements. To deal with asymmetries in the weighting of the errors, a loss matrix is used to adjust how the particular errors are weighed. More specifically, a doctor could adjust a loss matrix to prefer additional effort (more reviews) over false negative errors (healthcare), to prevent measurement data being classified as low priority.

In order to come up with a good classifier, proper selection of the input vector is important. Several input vectors should be considered to choose the one that leads to the best classification result. On the one hand, the optimum input vector is the one that uses all available relevant information. On the other hand, the data available may differ. Hence, the classifier should be able to deal with missing data.

As a strategy, we use a static input vector which is semantically divided into slots. A slot is a group of input neurons which receives input data from a single data entity e.g. a particular measurement and, in case of a perceptron, addresses several input neurons. For every measurement type like weight, blood pressure etc. n slots are reserved. Each slot contains information such as measurement value, measurement age in hours and, very important, availability of the particular information. Slots reserved for a particular measurement type are filled with the according last n measurements, e.g. there are some slots reserved for blood pressure, marked by "LOINC 8480-6". The rare case of missing information is dealt with by simply marking the availability input neuron. Of course, the perceptron needs to be trained accordingly to be prepared for missing information. The structure of the input vector is depicted in Figure 4.

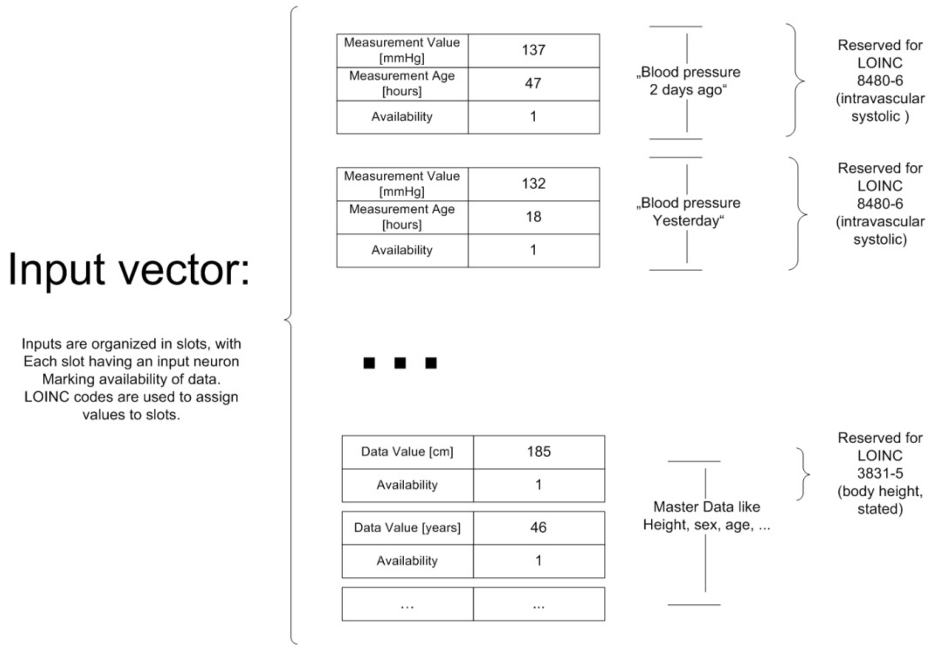

Fig. 4. Input Vector Construction.

For a given data set, a suitable topology must be determined for the classifying MLP. It is necessary to perform several training runs with varying number of hidden neurons, before committing to the most promising particular topology.

6 Experimental Results

For experimenting purposes, we have generated artificial patient measurement data. We used parts of data from the “Heart Disease Data Set“ (Courtesy of Cleveland Clinic Foundation) found in the UCI machine learning repository [7] and added some noise to create artificial timelines of patient monitoring data. Additionally, artificial events have been added by randomly increasing weight and decreasing blood pressure for a period of 11 days, peaking in a (noisy) 5% change in day 6. Then we have run simulations on how effectively these events are recognized by both statistic and rule-based classifiers alone and in combination.

The statistic classifier consists of a MLP with one hidden layer and a random nodes count, selected from 10 training runs. The input values are normalized to be in the range 0…1. Target (priority) values are 1 in case of an event, >0 if the day lies within the 11 days escalation period and 0 else.

The rule-based classifier consists of the following two rules that are arbitrarily derived from the artificial event generation process. Return value is 1 if a rule matches and 0 else.

```
IF last_weight > 110% of mean_weight
                              THEN recommend_hospitalization

IF last_blood_pressure < 90% of mean_blood_pressure
                              THEN recommend_hospitalization
```

Table 1. Results

	Neural network classifier	Rule-based classifier	Combined classifier
Av. Error %	**1.25**	18.70	2.05
false negative %	23.82	30.31	**17.98**

While the overall error ratio of the combined classifier is slightly worse than the error ratio of the neural network alone (Table 1), the ratio of patients prioritized too low is considerably lowered by the hybrid model.

7 Conclusion and Outlook

We have presented the core functionalities of the middleware architecture in the Fontane project: a network of medical telemonitoring devices, and a telemedicine center in which the measurements are classified with the objective of determining patients for whom a medical intervention is necessary. As the number of patients increases, inspecting daily measurement results become monotonous and requires more personnel. Using machine-based classification can help to reduce the risk of mistakes as well as to make doctors more productive, giving them more time for the relevant cases.

Our system is designed to be independent from and adaptable to the specific decease and therapy. To achieve this objective, both data models and classification machinery must be flexible. We have presented approaches to achieve this flexibility by supporting open data models, in-field updates to "static" rules, and machine learning. Currently, we consider the integration of a case based reasoning component as described in [9].

As an evaluation, we have presented results of feeding test data into our implementation. In the coming years, we will employ this implementation in field studies involving several hundred patients; this should give more insights into the practical operation of our hybrid approach to artificial intelligence.

References

1. Polze, A., Tröger, P., Hentschel, U., Heinze, T.: A Scalable, Self-Adaptive Architecture for Remote Patient Monitoring. In: 2010 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, isorcw, pp. 204-210 (2010)
2. Apache Camel Framework, http://camel.apache.org/
3. McDonald, C.J., Hu, S.M., Suico, J.G., Hill, G., Leavelle, D., Aller, R., Forrey, A., Mercer, K., DeMoor, G., Hook, J., Williams, W., Case, J., Maloney, P.: LOINC, a universal standard for identifying laboratory observations: a 5-year update. Clinical Chemistry 49(4), 624{633 (Apr 2003), http://www.ncbi.nlm.nih.gov/pubmed/12651816, PMID: 12651816
4. Levy, A.Y., Rousset, M.C.: CARIN: A representation language combining horn rules and description logics. In: ECAI 1996, pp. 323–327. (1996)
5. Xtext Language Development Framework, http://www.eclipse.org/Xtext/
6. Bishop, C.: Pattern Recognition and Machine Learning, Springer, New York (2006)
7. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/
8. Abraham, A., Corchado, E., Corchado, J. M.: Hybrid learning machines. Neurocomputing 72(13-15): 2729-2730 (2009)
9. Ashwin Kumar, K., Singh, Y., Sanyal, S.: Hybrid approach using case-based reasoning for domain independent clinical decision support in ICU, Expert Systems with Applicaton 36, 65-71 (2009)