Design of a Probabilistic Ontology-based Clinical Decision Support System for Classifying Temporal Patterns in the ICU: a Sepsis Case Study

Femke Ongenae, Tom Dhaene, Filip De Turck
Department of Information Technology
Ghent University - IBBT
Gaston Crommenlaan 8 bus 201, 9050 Ghent, Belgium
Email: Femke.Ongenae@intec.UGent.be

Dominique Benoit, Johan Decruyenaere
Department of Intensive Care
Ghent University Hospital
De Pintelaan 185, 9000 Ghent, Belgium
Email: Johan.Decruyenaere@UGent.be

Abstract

Medical time series contain important information about the condition of a patient. However, due to the large amount of data and the staff shortage, it is difficult for physicians to monitor these time series for trends that suggest a relevant clinical detoriation due to a complication or new pathology. This paper proposes a framework that supports physicians in detecting patterns in time series. It has three main tasks. First, the time-dependent data is gathered from heterogeneous sources and the semantics are made explicit by using an ontology. Second, Machine Learning techniques detect trends in the semantic time series data that indicate that a patient has a particular pathology. However, computerized classification techniques are not 100% accurate. Therefore, the third task consists of adding the pathology classification to the ontology with an associated probability and notifying the physician if necessary. The framework was evaluated with an ICU use case, namely detecting sepsis. Sepsis is the number one cause of death in the ICU.

1. Introduction

The rapid development of computing technologies has a major impact on healthcare, particularly in intensive care units (ICUs). This growth in computer technologies has been accompanied by an increase in complexity and number of monitoring equipment, thus generating large amounts of data that must be rapidly interpreted by the medical staff. Databases have become an essential part of ICUs to store, integrate and share data about the medical condition of a patient. Time series often appear in these databases and contain important information about the condition of a patient. For instance, it has already been shown that subtle changes in certain laboratory values reflecting vital organ function, even within the normal range, are of prognostic value for sepsis [1]. However, it is difficult for physicians to continuously monitor these time-dependent parameters for subtle or sometimes even overt changes that suggest a relevant clinical detoriation due to a complication or new pathology because of the large amount of data and staff shortage. This is particularly true for junior or senior physicians who lack clinical expertise in the field of the pathology. To solve this, this paper proposes a Clinical Decision Support System (CDSS) that gathers time series data about patients from various sources, detects trends and alerts the clinicians of possible new pathologies or complications.

A first challenge is that temporal knowledge is often not explicitly represented by the database. E.g., if a time point $09/04/2010$ 5:42:00 is associated with a White Blood Cell (WBC) count, the exact meaning of this time point is not known. It could be the time at which a blood sample was drawn or the time at which it was analyzed in the laboratory. To resolve this the framework integrates the data in an ontology [2]. Ontologies are used to represent and structure knowledge about a certain domain in a formal way. This ontology is used to annotate data with their meaning and express the relationships with other data. Moreover, it allows the integration of data coming from heterogeneous sources.

Second, computerized techniques are needed to detect trends. In the field of Machine Learning (ML) two approaches for time series classification can be identified. The first approach pre-processes the time series data by extracting and selecting features [3] such that classical statistical classification techniques can be used. The second approach uses techniques which can directly cope with high-dimensional non-linear temporal data. Both approaches were integrated into the framework. They take the medical time series as input and return a classification label as output e.g. indicating if a patient has the pathology or not.

Using ML techniques to predict the condition of a patient, is not 100% accurate or reliable. However, the CDSS is not meant to replace the medical staff. It is essentially

Nr.	Indicator	Parameter	Measure type
1	Fever	Temperature blood Temperature armpit Temperature rectal	Monitored Observed Monitored
2 3	Leukocytosis Leukopenia	White Blood Cell Count (WBC)	Laboratory
4	Plasma CRP $>$ 2SD above the normal value	CRP blood	Laboratory
5	Arterial Hypotension	Non-invasive Blood Pressure (NIBP) Systolic Arterial Pressure (SAP) Levophed	Monitored Monitored Prescription
6	Thrombocytopenia	Platelet Count	Laboratory

Table 1. Selected diagnostic criteria of sepsis

developed to assist junior or non-expert senior, and to lesser extent, expert senior physicians in diagnosing a patient's condition. It is therefore important to convey the accuracy or uncertainty of the prediction to the medical practitioner. Therefore, the developed framework explicitly models this uncertainty in the ontology and reasons with it.

The remainder of this paper is structured as follows. Section 2 introduces the sepsis use case. Section 3 details the architecture of the framework. Section 4 focuses on the modeling details of the sepsis ontology and Rules, while Section 5 describes some sepsis scenarios. Section 6 highlights the main conclusions and future work.

2. Use case: Sepsis

Sepsis [1] is a severe inflammatory response, called systemic inflammatory response syndrome (SIRS), of the body to an infection. In the US, approximately $750,000$ people are diagnosed with sepsis every year, with a mortality rate of 30%. Early detection is crucial, as appropriate therapy aimed to prevent further detoriation in organ failure reduces mortality by 15% [11]. However, sepsis can only be accurately diagnosed by the presence of a positive blood culture for a known pathogen. This infection is often discovered too late. However, there is a guideline that lists 25 possible indicators of sepsis, which allows early detection [7]. Continuously monitoring these, often time-dependent, parameters is difficult, which makes it the ideal use case for the developed framework.

To perform an initial evaluation of the framework, 6 of the 25 indicators were selected, as shown in Table 1. It was investigated if the framework could be used to detect trends in these parameters to automatically alert physicians of patients who might have sepsis. For this, data about 650 ICU patients, of whom 342 have sepsis, was collected during three consecutive years from the ICU databases of Ghent University hospital. The parameters collected which match the indicators are also shown in Table 1.

3. Architecture Description

The architecture of the CDSS is shown in Figure 1. Three layers can be discerned, as detailed below.

3.1. Data Management Layer

The modern ICU contains computerized equipment to convey the condition of a patient such as monitoring equipment and electronic patient records integrated in a *Patient Data Management System (PDMS)*, an *Infection Management System (IMS)* which tracks the infections and antibiotic treatments of a patient and a *Laboratory* database. All this data is stored in medical database systems across the hospital, as shown in the *Data Management Layer* of Figure 1. Time series often appear in these databases and contain important information about the condition of a patient.

3.2. Semantic Reasoning Layer

Medical databases have some limitations, such as the lack of interoperability and standardization [8]. Moreover, the meaning of the time points in the database is difficult to derive. To resolve this, the *Semantic Reasoning Layer* contains *ontologies* for the integration and analysis of the medical and temporal data. Existing *Medical Ontologies* are integrated and extended with *Domain Ontologies* which model the information specific to a particular hospital setting such as the measured parameter values. These *Domain Ontologies* also contain probabilistic information which express the probability that a patient has a pathology if particular conditions are met, e.g. a patient with a low WBC has 17% chance of having sepsis. A *Temporal Ontology* associates each parameter value with its meaning in time.

Due to the foundation of ontologies in *Description Logics* (*DL*), the models can be formally proofed by using a *DL Reasoner*. This *DL Reasoner* is used to detect inconsistencies in the model and infer new information from the correlation of the data. E.g., a concept *Fever* is created in the ontology, which automatically detects patients with a temperature above $38\,^{\circ}\mathrm{C}$. More complex logic is expressed by defining *Rules* [10] on top of the ontology. These *Rules* express algorithms which take advantage of the temporal information in the ontology. This way, increasing or decreasing trends in time series are automatically detected and notified to the clinicians. This notification is handled by the *Notification Module*. It allows to specify which clinicians are interested in what information about which patient.

3.3. Learning Layer

As explained in Section 2, not all time series trends can be easily expressed as Rules. ML techniques handle these difficult cases, as shown in the *Learning Layer* of Figure 1.

3.3.1 Data selection and pre-processing

First, training data is extracted from the ontology. Which parameters are used as input and which pathology needs

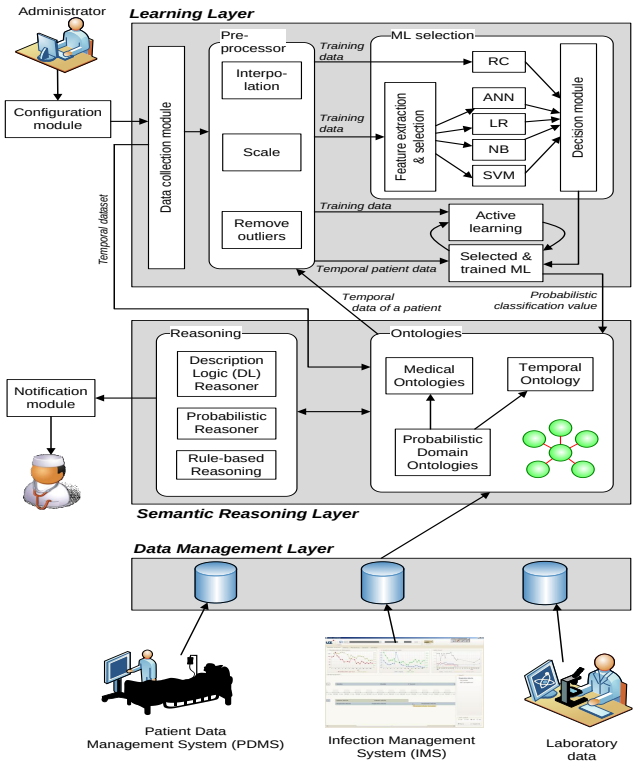

Figure 1. The probabilistic ontology-based CDSS for classifying temporal patterns

to be predicted is specified by the administrator by selecting the appropriate ontology concepts in the *Configuration Module*. The *Data Collection Module* automatically extracts all the applicable data from the ontology and provides it to the *Pre-processor*. This *Pre-processor* contains several modules to clean up the patient data. The *Remove Outliers* component removes outliers from the time series or even removes complete patients, e.g. patients for whom the input time series is too short. The *Scale* component centers the input values at zero. This is often beneficial for the learning algorithms of ML techniques. Finally, interpolation is needed. Many parameter measurements are performed by hand, so there exists some variance in the intervals between succeeding measurements. However, to use the time series as input for the ML techniques, they need to contain measurements over regular time intervals and these intervals must be the same for all the input parameters. Currently, the *Interpolation* component offers three techniques, namely outmiddling, linear and step interpolation.

3.3.2 Training the ML techniques

Two ML approaches are used to classify the pre-processed time series. The first approach extracts and selects features from the time series data such that classical classification techniques can be used. Most of these techniques have been designed with a static data model in mind and are not suitable for coping with the dynamic nature of time series. The *Feature Extraction & Selection* module first generates features from the time series such as the slope, and secondly selects the most appropriate features and thus reduces the amount of input data [3]. Next, a classifier is applied to the selected features to classify the data. Four such classifiers are integrated namely *Linear Regression (LR)*, *Artificial Neural Networks (ANN)*, *Support Vector Machines (SVM)* and *Naive Bayes (NB)* [12].

The second approach focuses on techniques which can directly cope with the temporal data such as Recurrent Neural Networks (RNN). An obstacle when using RNNs is that only a few training algorithms exist which are complex and often yield poor results [4]. However, recently an approach, called Reservoir Computing (RC) [13], was developed to simplify this training process. The key idea is to model the dynamic system producing the time series data in a reservoir consisting of a RNN. The reservoir is then read by a linear readout function. The training algorithm only affects this readout function. For training linear functions many algorithms exist such as linear regression.

Afterwards, the *Decision Module* selects the technique with the highest accuracy to be used as the classifier to detect trends for this particular pathology. This technique is plugged into the *Selected & Trained ML* component.

3.3.3 Probabilistic classification and active learning

The patient, who is classified by the trained classifier, is preprocessed in the same manner as the training data. The ML technique outputs a classification label, e.g. 1 or -1, indicating if the patient has the pathology or not. It is important to convey the uncertainty of the prediction to the physician. Therefore the classification is added to the ontology with a probabilistic value that expresses its accuracy. A *Probabilistic Reasoner* is used to combine this classification with the other probabilistic data about this pathology in the ontology. This way, a probability is obtained which expresses how trust-worthy the advise of the framework is.

Finally, the *Active Learning* component continuously trains the *Selected & Trained ML* and improves its accuracy. Newly labeled data, e.g. a physician indicates that a patient has sepsis, is periodically gathered from the ontology as training data. This data is intelligently selected such that the classifier does not become overfitted, e.g. patients that were wrongly classified by the classifier are ideal input for active learning. More information can be found in [9].

4. Modeling the sepsis ontology and Rules

Part of the ontology that models the sepsis use case is shown in Figure 2. The concepts preceded with the `galen`

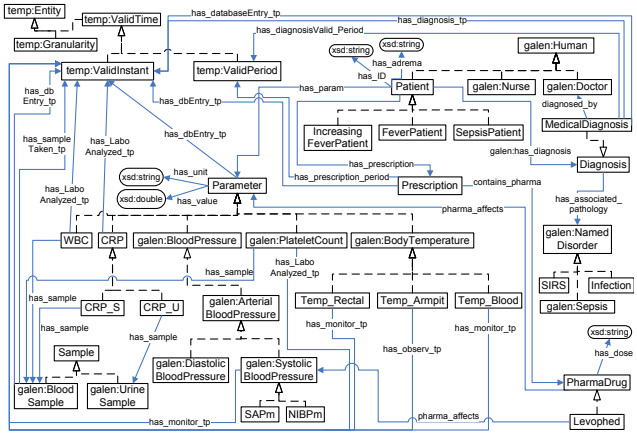

Figure 2. Fragment of the ontology modeling the time series of the sepsis parameters

keyword are imported from the Galen Ontology[1]. This shows nicely how other ontologies are integrated and reused in the *Semantic Reasoning Layer*. The *SWRLTemporalOntology* [10] models the temporal information. These concepts are preceded with the `temp` keyword. By defining meaningful relationships between the medical and temporal concepts, e.g. `has_dbEntry_tp`, the semantics of the different time points in the database can be differentiated. The Protégé editor[2] was used to develop the OWL[3] ontology.

Time series data is extracted from the medical databases and automatically mapped on the ontology by using D2R[4]. D2R specifies that the `LaboTable` in the database maps on the `Parameter` concept in the ontology. This table has a column `ParamID`. Rows with `ParamID` 1500345 are mapped on the `CRP` concept. The columns `entrytime` and `datetime` are mapped on the `has_dbEntry_tp` and `has_laboAnalyzed_tp` relations respectively. The first relation expresses the time at which the data was entered into the database, while the second indicates the time point at which the analysis of the blood sample was obtained. From the `has_sample` and `has_sampleTaken_tp` relations can be derived at which time point the blood sample was drawn. Finally, the `value` column is mapped on the `has_value` relation in the ontology. This way an entire time serie is encoded in the ontology by creating multiple `CRP` instances, each associated with a value, a unit and the different time points. This `CRP` value is then associated with a `Patient` through the `has_param` relation. Note that concepts cannot only be associated with specific time points, but also with intervals during which they are valid. E.g., the `has_prescription_period` relation.

[1] `http://www.opengalen.org/`
[2] `http://protege.stanford.edu/`
[3] `http://www.w3.org/TR/owl2-overview/`
[4] `http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/`

The Pellet Reasoner[5] is used as *DL Reasoner* to check the consistency and automatically classify the ontology. E.g., the concept `SepsisPatient` is defined as a patient who has `SIRS` and an `Infection`, as follows:

$$\texttt{SepsisPatient} \sqsubseteq (\exists \texttt{has\_diagnosis}(\exists \texttt{has\_associated\_pathology SIRS})) \sqcap \sqsubseteq (\exists \texttt{has\_diagnosis}(\exists \texttt{has\_associated\_pathology Infection}))$$

By using the SWRL Temporal Built-In Library [10], *Rules* are defined that use the temporal information in the ontology to create complex algorithms. For example,

$$\begin{aligned}&\texttt{FeverPatient}(?p) \wedge \\ &\texttt{has\_bodyTemp}(?p,?temp1) \wedge \texttt{has\_bodyTemp}(?p,?temp2) \wedge \\ &\texttt{has\_value}(?temp1,?v1) \wedge \texttt{has\_value}(?temp2,?v2) \wedge \\ &\texttt{swrlb:greaterThan}(?v2,?v1) \wedge \\ &(\texttt{has\_observ\_tp}(?temp1,?t1) \vee \texttt{has\_monitor\_tp}(?temp1,?t1)) \wedge \\ &(\texttt{has\_observ\_tp}(?temp2,?t2) \vee \texttt{has\_monitor\_tp}(?temp2,?t2)) \wedge \\ &\texttt{temp:equal}(?t1,?t2, \texttt{temp:Hours}) \wedge \texttt{temp:before}(?t1,?t2) \wedge \\ &\Rightarrow \\ &\texttt{IncreasingFeverPatient}(?p)\end{aligned}$$

is a *Rule* that detects if the body temperature of a patient, who already has a fever, is still increasing within the hour. If so, the patient is categorized as an `IncreasingFeverPatient`. This allows the notification module to alert the appropriate physician.

The Reasoner Pronto[6] is used to model and reason with the probabilistic information in the ontology. Probabilistic statements are expressed in OWL by using axiom annotations. An example is discussed in Section 5.2.

5. Detailed scenario description

5.1. Training ML to detect sepsis

First, the *Configuration Module* is configured, as shown in Figure 3(a). The `CRP_S`, `WBC`, `NIBPm`, `SAPm`, `Levophed`, `Temp_Rectal`, `Temp_Armpit`, `Temp_Blood` and `PlateletCount` concepts are used as input and the `SepsisPatient` concept as output. The relationships used to find the associated time points are also specified, e.g. the`has_observ_tp` relation for the `Temp_Armpit` concept. The `Temp_Blood` values are preferably used as the temperature inputs as they are very reliable. If there are however `Temp_Armpit` or `Temp_Rectal` values which precede or follow the `Temp_Blood` time series, then they should be added to the temperature time series. This is configured by indicating that these parameters should be combined into 1 input parameter and specifying a ranking. Similarly, the `NIBPm` and `SAPm` parameters are combined into 1 `Bloodpressure` time series, for which `NIBPm` has the highest ranking. Finally, the period of time over which the time series data should be searched is specified.

By using these configuration settings, the *Data Collection Manager* retrieves all the patients from the ontology

[5] `http://pellet.owldl.com/`
[6] `http://pellet.owldl.com/pronto`

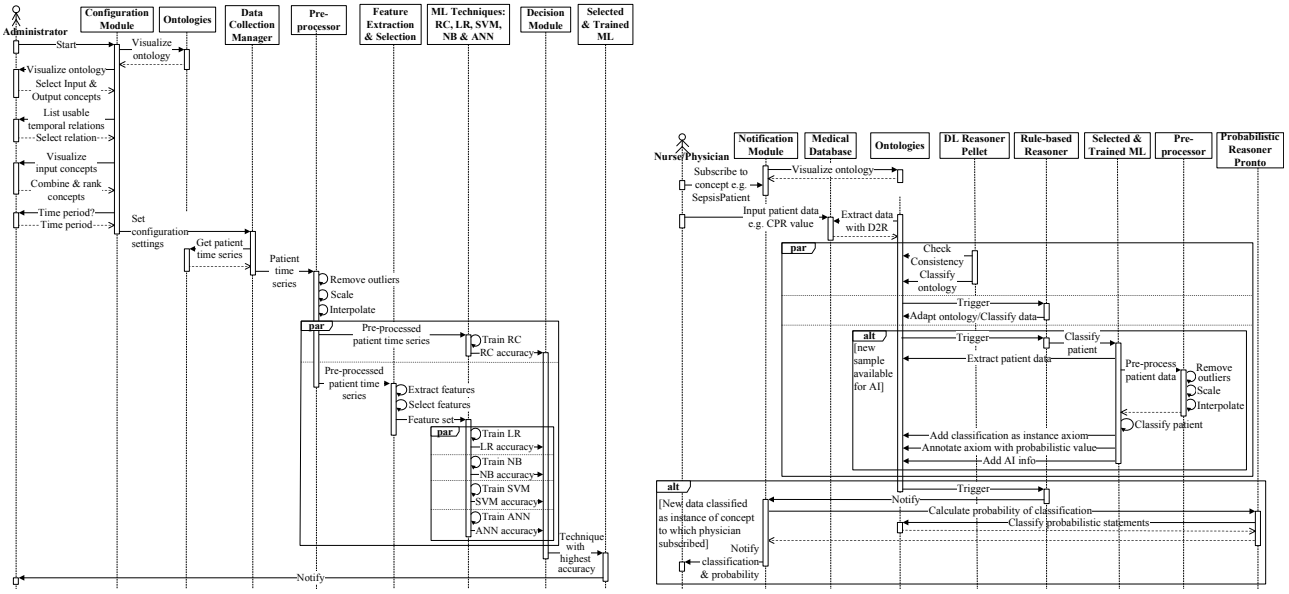

Figure 3. Sequence diagram of a) training the ML techniques, b) detecting a patient with sepsis

that have been explicitly stated as either being or not being a `SepsisPatient` by a physician by using SPARQL queries[7]. In total 650 patient with five input parameters, namely `CRP`, `WBC`, `PlateletCount`, `BloodPressure` and `Temp`, and 1 output parameter were collected.

Second, the data is pre-processed. Outliers are removed from the time series. Patients for which the time series do not have an overlapping minimal length of 40 values are also removed. This leaves 125 patients. Next, the time series are scaled to the $[-0.9, 0.9]$ interval. These bounds are chosen instead of -1 and 1 to avoid excessive weight saturation in the RC. Finally, linear interpolation is used, such that all the time series have a length of 72 time points.

Third, feature extraction and selection is performed. The following techniques are implemented in the *Feature Extraction & Selection* module to extract features from the time series: calculating the minimum, maximum, mean, median, 25^{th} and 75^{th} percentile, standard deviation, the linear regression coefficients and the area under the curve (AUC) of the time series. For the sepsis case, expert opinion reveals that the tail of the time series contains more information than the start of the series. Therefore the feature extraction was repeated for the time series each time reduced with one at the head. Finally, the slope was calculated for each two succeeding points in the time series. The slope was also calculated for each two points with a distance of 2, 3, and so further until 71. Out of all these features, useful ones are selected. Currently one feature selection technique is implemented, namely a greedy algorithm which iteratively adds the feature that improves prediction the best. This approach is similar to the one used in [5], but in each iteration the set of candidates is filtered so that it contains only features that are not collinear with the already selected set. This drastically reduces the number of candidate features in each iteration and speeds up the selection process. The classifier used in this hybrid filter-wrapper [3] is the NB.

Finally, the data is used by the ML techniques as training data. The *NB* and *LR* are custom implementations, while for the *SVM*, *ANN* and *RC*, the libSVM[8], FANN[9] and RC-Toolbox[10] libraries were integrated. All these classifiers are trained using cross-validation and their parameters, such as the reservoir size of the *RC*, are optimized using parameter sweeps. In the current implementation, the classifiers can be optimized towards two accuracy measures, namely the AUC and the maximum balanced accuracy (Max. Acc.).

Table 2 shows the AUC and Max. Acc. achieved for the sepsis case with the five input time series for each of the classifiers. The *NB* classifier with *Feature Extraction & Selection* has the highest accuracy and is used as classifier.

5.2. Detecting a patient with sepsis

When a new patient is added to the ontology who has the required time series data for the sepsis case, this patient is automatically classified by the trained *NB*, as shown in Figure 3(b). The time series data is extracted from the ontology using SPARQL and is pre-processed in the same way as the training data. The *NB* with *Feature Extraction & Selection*

[7] `http://www.w3.org/TR/rdf-sparql-query/`

[8] `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[9] `http://leenissen.dk/fann/`

[10] `http://snn.elis.ugent.be/rctoolbox`

Classification technique	AUC	Max. Acc.
SVM	0.712	0.700
NB	0.752	0.718
LR	0.573	0.604
ANN	0.569	0.621
RC	0.686	0.683

Table 2. AUC and Max. Acc. for the various ML techniques for the sepsis use case

returns a classification label, namely 1 or -1, indicating if the patient has sepsis or not.

To convey the accuracy of the prediction to the physician, the classification is added to the ontology as an instance axiom annotated with a probability. The annotated axiom

```
<owl11:Axiom>
    <rdf:subject rdf:resource="#patient1"/>
    <rdf:predicate rdf:resource="rdf:type"/>
    <rdf:object rdf:resource="#SepsisPatient"/>
    <pronto:certainty >0.752;0.752</pronto:certainty>
</owl11:Axiom>
```

expresses that a particular patient, namely `patient1`, has 0.752% chance of being a `SepsisPatient`. The AUC of the *NB* is used as probabilistic value.

Information about the ML technique is added to the ontology, by using the concepts shown in Figure 4. First, an instance of `PredictedDiagnosis` is created to indicate that this classification is a predicted diagnosis. This instance is connected to the classification class `SepsisPatient` with the `has_diagnosis` relation and associated with the time point at which the prediction was made. Next, a new instance of the `NB` concept is created to specify by which `ML_technique` the classification was predicted. This `NB` instance is attached to the classification by using the `predicted_by` relation and associated with `Configuration` and `Settings` instances. These instances contain all the parameter values of the *NB*. Finally, instances of the `BalancedAcc` and `AUC` classes are created containing the values of Table 2 of the *NB*.

The *Notification Module* allows staff members to subscribe to classes, e.g. `SepsisPatient`, defined in the ontology. Each time an instance is classified as a member of this class, the staff member is alerted. The associated trust in this prediction is also calculated by combining all the probabilistic information in the ontology.

6. Conclusions

In this paper a framework was proposed to 1) augment the time series data in medical databases with semantic information by using ontologies, 2) use Machine Learning techniques to automatically classify this semantic time series data as suggesting a relevant clinical detoriation due to a complication or new pathology or not, and 3) add this classification to the ontology, while expressing the uncertainty that is associated with this prediction. Future work will mainly focus on integrating ensembles [6].

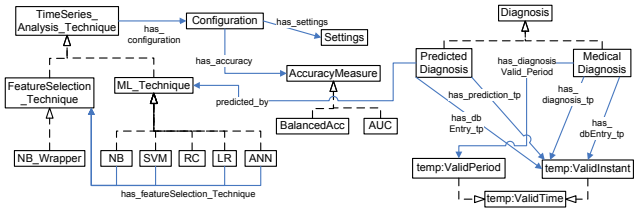

Figure 4. Fragment of the ontology modeling the time series classification methods

7. Acknowledgment

Femke Ongenae would like to thank the IWT, Institute for the promotion of Innovation through Science and Technology in Flanders, for supporting this work through her PhD grant.

References

[1] J. F. Dhainaut and et al. Dynamic evolution of coagulopathy in the first day of severe sepsis: relationship with mortality and organ failure. *Crit Care Med*, 33(7):450–452, 2005.

[2] T. Gruber. A translation approach to portable ontology specifications. *Knowl Acquis*, 5(2):199–220, 1993.

[3] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Mach Learn Research*, 3:1157–1182, 2003.

[4] H. Jaeger. A tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF, and the "echo state network". Technical Report GMD 159, German National Research Institute for Computer Science, 2002.

[5] P. Langley and S. Sage. Induction of selective bayesian classifiers. In *Proc. of UAI-94*, pages 399–406, 1994.

[6] M. C. Lee and et al. A two-step approach for feature selection and classifier ensemble construction in computer-aided diagnosis. In *Proc. of CBMS'08*, pages 548–553, 2008.

[7] M. M. Levy and et al. 2001 sccm/esicm/accp/ats/sis international sepsis definitions conference. *Intens Care Med*, 29:530–538, 2003.

[8] H. V. D. Linden and et al. Towards a generic connection of EHR and DSS. In *Proc. of MIE2005*, pages 211–216, 2005.

[9] B. Longstaff, S. Reddy, and D. Estrin. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *Proc. of PervasiveHealth*, 2010.

[10] M. J. O'Connor and A. K. Das. A lightweight model for representing and reasoning with temporal information in biomedical ontologies. In *Proc. of HEALTHINF*, 2010.

[11] E. Rivers and et al. Early goal-direct therapy in the treatment of severe sepsis and septic shock. *New Engl J Med*, 345(19):1368–1377, 2001.

[12] S. J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, 3rd edition, 2009.

[13] D. Verstraeten and et al. An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):414–423, 2007.