Project PROPHET™ : Brief report of preliminary results

ABSTRACT

Forecasting forthcoming "health events" is an extremely challenging task for the Remote Patient Monitoring systems (RPM systems) sector, which relies in real time information and communication technologies. Remote patient monitoring is a medical service which includes following and observing patients that are not in the same location with their health care provider. In general, the patient is equipped with a “smart” monitoring device, and the recorded data (vital signs) are securely transmitted via telecommunication networks to the health care provider. Modern remote patient monitoring devices are small, discrete and easy to wear, allowing "bearers" to move freely and with comfort. In this framework, MOKAAL pc has developed the IFS_RPM service (Integrated Facilitation Services for Remote Patient Monitoring) supplying the necessary ICT infrastructure, which is necessary for the provision of the RPM services. Following the completion of IFS_RPM project, MOKAAL pc launched a research project under the code name "PROPHET™" .

PROPHET™ main objective is to investigate the possibilities of introducing a real time predicting model based on remotely collected vital signs, that would utilize time series of metric data in conjunction with the information stored in the Electronic Health Records (EHR) of the "bearer", attempting to predict in real time, the probability of a "health event" occurring in the near future.

To meet this objective, the PROPHET™ project team designed an evolutionary prototype of the "health event" forecasting model, which was developed and tested in a laboratory environment and it will be upgraded to a working prototype to be tested in real conditions, in order to be incorporated into the IFS_RPM system, after reaching its maturity state.

INTRODUCTION

Wearable Remote Patient Monitoring Devices are medical devices that are widely used to measure basic medical indexes, like ECG, HR, RR, oxygen saturation, body temperature, posture and physical exercise monitoring.

Wearable RPM devices that are supported by advanced IT systems and advanced sensors, provide constant monitoring capabilities to observe and evaluate human physiology and consequently to assist in faster response to health events and application of the correct therapeutic protocol. Usually, in a health care environment, technology support is essential in implementing clinical trials, monitoring disease progress and increasing the decision making rate

In the realization of the PROPHET™ project, MOKAAL used an RPM device manufactured by an established European brand, as the core component of the Wireless Body Area Network (WBAN) which was worn by each participant in the study. The said RPM device is a 3 lead ECG mobile monitoring system which allows clinicians to continuously record and monitor full disclosure ECG, heart rate, respiration rate and motion data, whilst allowing full patient mobility. The device

monitors for out of normal range vital signs and key cardiac arrhythmias (atrial fibrillation, tachycardia, bradycardia, ventricular fibrillation, and asystole).

The PROPHET™ predicting model was developed with the use of laboratory data collected over a period of 260 days. The LAB Database comprises a main section of raw medical data, along with a number of periodical reports depicting the assessment of a personalized risk factor for each subject at a given moment. These reports were the outcome of regular medical tests and health status assessment of all participants, throughout the duration of the study. The predicting model uses machine learning techniques, to introduce a set of algorithms that teach the parameters of the model from a set of training data for which we know the results, aiming to predict with the greatest possible accuracy, the results to be obtained from the processing of the test data.

The PROPHET™ predicting model is based on the Decision Tree algorithm principles. Decision Tree algorithm is a supervised learning algorithm, used to create a model capable to predict the class of a target variable, by implementing simple decision rules that were established during the processing of the training data subset.

THE PROPHET™ LAB DATASET

The LAB data were collected over a period of 260 days, with the participation of 10 outpatients with diagnosed cardiovascular disease, wearing a RPM device for 12 consecutive days per session, for a total of 60 days (5 sessions) throughout the duration of the study. The RPM device used is a small, lightweight, portable, non-invasive, with rechargeable battery, connected to disposable electrodes placed on the patient's body. The device monitors for out of normal range vital signs and key cardiac arrhythmias (atrial fibrillation, tachycardia, bradycardia, ventricular fibrillation, and asystole). It also features patient-activated event recording, that allows patients to press the button when symptoms are felt. This triggers diagnostic quality ECG to be recorded and transmitted to an online database (IFS_RPM DB) over a wireless connection. The system provides full configurability of the duration and frequency of monitored data. The system also records continuously ECG on the device during monitoring to provide a back up to wireless transmission if required and to permit retrospective analysis of performance of the cardiac arrhythmia detection algorithms. MOKAAL's "mhealth" viewing and analysis software, was used to import and review full disclosure data at the end of the evaluation. In conclusion, the RPM device incorporates the core functions of the traditional remote ECG monitoring devices into one single device : the full disclosure data from the Holter (7 days), the events (both auto-captured and patient activated) transmitted immediately from the Event and implantable loop recorders. ECG measurements are rendered in μV per millisecond (ideally with a steady step), while respiratory rate measurements yield breaths per minute and heart rate measurements yield pulses per minute. Upon arrival of any time series instance into the PROPHET SERVER, the "mhealth" reception module performs a consistency check on the μV values of all 3 leads which are registered in the incoming observations, dumping any instance in which any of these values doesn't match the acceptable ECG pattern

The structure of PROPHET™ lab dataset

The main pool of LAB data was comprising time series (i.e. sequence of observations taken sequentially in time), carrying raw medical data collected from each participant (“bearer”) with the use of the WBAN. These data represented biological signals of the "bearer", which were constantly collected every 2,000 millsec (approx) by the RPM device and were stored locally (in the "Full Disclosure" dataset).

The “Health Event” dataset is a subset of the "Full Disclosure" dataset, containing only those time series in which key cardiac arrhythmias or/and out of normal range vital signs were detected, i.e. atrial fibrillation, tachycardia, bradycardia, ventricular fibrillation, and asystole. Cardiac events are detected using arrhythmia detection algorithms running on the RPM device, while a patient-activated event button ensures ECG can be recorded during symptomatic periods. These algorithms are designed to detect and record Atrial Fibrillation and have been tested according to ANSI/AAMI EC57:2012. For any such event detected (both auto-captured and patient activated), The device transmits immediately (in real time) a fix format message to the IFS_RPM Server in the central premises of MOKAAL.

MAIN DATASET ("FULL DISCLOSURE") : summary technical information

TIMESTAMP : Day of session (01,02, …), time (hh:mm:ss)

- ECG : 3 Leads, Frequency response: 0.5Hz - 40Hz, Sample Rate: 360 samples per second, Resolution: 12 bit
- EVENT DETECTION : Bradyarrhythmia, Tachycardia, Ventricular Fibrillation, Atrial Fibrillation, Supraventricular tachycardia, Asystole, Patient activated event recording
- Heart Rate
- Respiration Rate
- Impedance Pneumography, Sample rate: 120 samples per second, Resolution: 12 bit
- ACTIVITY : 3-axis accelerometer, Scale: ±2G, Sample rate: 100 samples per second, Resolution: 12 bits on each axis

“HEALTH EVENT” subset : In addition to the information contained in the Main Dataset, this subset includes codes corresponding to the detected “health event” at that particular moment.

Arrhythmias Monitored by RPM Device

The Heart Rate averaging computation is as follows : The average of the last 12 seconds R-to-R intervals (up to 16 intervals) for rates greater than or equal to 60BPM and average of last 8 R-to-R intervals for rates below 60BPM, the update rate of the Heart Rate on the display is once per update interval period.

Asystole is a state of no cardiac electrical activity. The RPM device will trigger this arrhythmia when it is present for 10 consecutive seconds. Pause episodes are short term events (less than 10 seconds) which are not detected by the RPM device.

Atrial Fibrillation is a cardiac arrhythmia (abnormal heart rhythm) that involves the two upper chambers (atria) of the heart. The RPM device will trigger this arrhythmia

when it is present for at least 15 consecutive seconds. Episodes shorter than this are not recorded.

Bradycardia is defined as a slow resting heart rate, the RPM device will trigger this arrhythmia when the average heart rate is continuously lower than a user configured threshold (default 50 BPM) for a 30 seconds confirmation time. Episodes shorter than this are not recorded.

Tachycardia is a fast heart rhythm, the RPM device will trigger this arrhythmia when the average heart rate is continuously higher than a user configured threshold (default 150BPM) for a 30 second confirmation period. Episodes shorter than this are not recorded.

Ventricular Fibrillation is a condition in which there is uncoordinated contraction of the cardiac muscle of the ventricles in the heart, making them quiver rather than contract properly and resulting in a random and chaotic fluctuation in the ECG signal of the patient. The RPM device will trigger this arrhythmia when it is present for at least 20 consecutive seconds of clean signal or at least 29 consecutive seconds of signal with high noise levels. Episodes shorter than this are not recorded.

For implementing the predicting modelling machine learning algorithm (predictive algorithm), the PROPHET™ laboratory datasets were divided into 3 subsets : The TRAINING data subset that was used to adjust the parameters of the model so the machine be able to recognize patterns in the data set, the VALIDATION data subset that was used to improve the generalization capacity of the model as well as to improve the efficiency and the accuracy of the algorithm used to train the machine and the TEST data subset which is used to evaluate the ability of the machine in predicting upcoming "health events", based on its training. The aim was to develop a tool that will search the 1st pool of data (the “full disclosure” dataset) to locate one or more repeating patterns of time series by processing the time coherent changes occurring in the 3 types of biosigns (ECG, RR, HR) collected by the RPM device, throughout the session. Then to locate the "health events” detected by the RPM device within the same session (stored in the "health events” dataset) and apply an algorithm to assess the correlation of the appearance of each pattern with the advent of detected health events.

THE PREDICTIVE PATTERN

In general terms, a pattern is a description of a state occurring over and over again within a given dataset. Patterns are used in solving problems, by overshooting personalized solutions. In the framework of the project, we focused in applying pattern mining in the supervised setting where we have marked a specific target variable and we want to identify patterns in the “Full Disclosure” TRAINING subset, capable in locating this variable in the “Health Event” dataset. The objective was to develop an algorithm for detecting predictive patterns that will eventually lead in predicting the target variable in the “Full Disclosure” TEST subset.

The key challenge in building classification models for PROPHET dataset, was to define a group of representative features that would be able to express accurately

the time-sensitive aspect of the “Full Disclosure” dataset that is important for the prediction of the target variable.

Both “Full Disclosure” and “Health Event” datasets are typical examples of temporal databases as they store data relating to time instances. The stored information is related to past and they are considered to be uni-temporal databases as they have one axis of time, namely the instances’ timestamps. Each data instance in both datasets, is associated with a single class label which means that all temporal observations are equally useful for classification. By definition, a classification model attempts to draw some conclusion from observed values. Given one or more inputs, a classification model will try to predict the value of one or more outcomes. Our aim was to develop a pattern mining technique that would take into account the local nature of decisions for monitoring and event detection problems. We decided to experiment with the Recent Temporal Pattern (RTP) mining framework [1], which mines frequent temporal patterns backward in time, starting from patterns related to the most recent observations. Our objective was to present a classification model that can accurately detect adverse health events and apply it in future RPM sessions for the same patient or for other patients with similar health conditions (i.e. suffering from some kind of cardiovascular disease).

The application in PROPHET time series

Assuming : (1) that $P_{FD} = \{t_i,x_i\}$, i=1,n is the TRAINING dataset where $x_i \in$ ”Full Disclosure” and x_i is a temporal instance of a multivariate dataset in the time interval between t_1 and t_n and (2) that $P_{HE} = \{t_j,y_j\}$, j=1,n is the TRAINING dataset where $y_j \in$ “Health Event” and y_j is a target variable associated with x_j at time t_j in the time interval between t_1 and t_n, then our objective was to build a function $f : X \rightarrow Y$ that can detect and mark health events among unmarked instances in the TEST dataset.

In PROPHET environment, every data observation x_i is a record that contains 5 discrete biosigns of a specific patient (a “bearer” of a RPM device), collected the given moment in time (t_i) and the class label y_j denotes whether a “health event” was diagnosed by the RPM device at the specific t_i. Taken in account that classification is the process of predicting the target in a dataset of given temporal observations strings and that classification predictive modelling is the task of approximating a function {F} using input variables (x) to discrete output variables (y), our aim was to learn a classifier – which is a function that assigns a target to a temporal observation string – that could predict a forthcoming “health event”, with the lowest possible percentage of false negative results.

The processing of PROPHET datasets of biosigns

For learning the classifier, we applied a space transformation $[T] : x_i \rightarrow (x_i)'$ that mapped each observation x in the “Full Disclosure” dataset, to a numerical representation x' (a feature vector), preserving in parallel the predictive temporal features of x as much as possible. After applying [T], we employed Decision Tree to learn function {F}.

More specifically, we learn [T] from the biosign strings (i.e. the laboratory values) in the "Full Disclosure" dataset, with the use of temporal pattern mining, by applying the following steps :

1. Define "cohorts" : A cohort is a group of *r* successive instances registered in the "Full Disclosure" dataset. In the LAB environment *r* was originally set to 10. In the context of the temporal pattern mining proceedings, a "cohort" could be considered a standardized unit of temporal patterns. In this context, a temporal pattern in the PROPHET dataset is made of one or more successive cohorts. Cohorts do not overlap, therefore cohort c_1 includes the observations collected within the time interval t_1-t_{10}, cohort c_2 contains those collected within the time interval t_{10}-t_{19} and so forth.

2. Define a temporal abstraction[1] pattern : All temporal instances were organized in cohorts and all cohorts were classified into 5 distinct abstract states as described in §6 of the "The LAB implementation" chapter. The "interval state" is defined as a state occurring during a distinct time interval. The interval state is denoted by (B,S,t_s,t_e) where B is an observation in a temporal variable, S is a member of the abstractions' values set (e.g. H=high, L=low) and t_s,t_e the start time and end time (respectively) of the interval state.

3. Define Multivariate State Sequence : After awarding each cohort a unique abstract state, every cohort C_m (m=1, n/r) is represented in a dataset C as a Multivariate State Sequence (MSS). Each record of dataset C displays 2 timestamps (start – end of cohort's instance). There is only one temporal relation between two successive cohorts (state intervals) in the PROPHET dataset, i.e. the "Finish-to-Start" relation.

4. Temporal Patterns : *A temporal pattern is defined as* $P = (\langle S_1, \ldots, S_k \rangle, R)$, *where* S_i *is the* i^{th} *state of the pattern and R is an upper triangular matrix that defines the temporal relations between each state and all of its following states :*

$i \in \{1, \ldots, k-1\} \wedge j \in \{i+1, \ldots, k\} : R_{i,j} \in \{b, c\}$ *specifies the relation between* S_i *and* S_j *(definition from "Mining Recent Temporal Patterns for Event Detection in Multivariate Time Series Data", I. Batal, …)*.

In the context of "PROPHET" project, R parameter is redundant as there is only one temporal relation between each state and all of its following states. A temporal pattern (TP) in the PROPHET dataset, is defined as a timeordered sequence of *n* successive valid cohorts C_i (i=1,n, n≥1) and its size is defined by the number of states it contains. So if a pattern contains k states, is defined as a k-pattern. Therefore, for a Multivariate State Sequence to contain a k-pattern, requires matching all k states of this pattern.

5. Recent Temporal Patterns (RTP) : Generally, we accept that recent observations of biosigns (variables x_i) are considered to be more predictive comparing to distant observations, in respect of predicting a forthcoming "health event". Nevertheless, this is not always true. Therefore, in order for a temporal pattern (TP) to be considered a

[1] An abstraction is a general concept or idea, rather than something concrete or tangible. The goal of "abstracting" data is to reduce complexity by removing unnecessary information [1]

“Recent Temporal Pattern” (RTP) in regard of a specific target variable, we narrowed the aforementioned assumption in a limited number of temporal patterns in the TRAINING dataset, that meet the following condition : The time frame between the start time of the given TP and the detection time of the specific target variable (as stored in “Health Event” dataset) is less than 36 hours.

Based on the above, the objective of identifying a predictive pattern that could be effectively integrated into the "health events" predicting model, seemed to be feasible. As we stated earlier in this report, the PROPHET™ predicting model is based on the Decision Tree algorithm principles and in particular it uses the “apriori” algorithm

DECISION TREE ALGORITHM

As stated above, machine learning classification is a three-step process : training step, validation step and prediction step. In the training step, the predicting model is developed based on given training data. In the validation step certain techniques are applied to improve the generalization capacity of the model as well as the efficiency and the accuracy of the algorithm used to train the machine. In the prediction step, the model is used to predict the response for given data. Decision Tree algorithm belongs to the family of supervised learning algorithms and is a typical example of a low bias algorithm. The reason for using a Decision Tree is to train a predicting model, capable to predict the class or value of a target variable by assimilating a set of simple decision rules, derived from already processed raw data. PROPHET predicting model is based on the “categorical variable” decision tree type, which includes categorical target variables that are divided into categories. This means that every stage of the decision process falls into one of the categories, and there are no in-betweens.

Identifying Predictive Patterns – The concept

Project’s objective was to predict a forthcoming health event (target event) within the timeframe of an RPM session. In responding the health event prediction challenge, an algorithm belonging to the Decision Tree (DT) family was employed, to search for predictive patterns in the “Full Disclosure” dataset. The goal was for the system to learn a set of rules of the form : pattern⇨target event, in order to be considered as classifier. The task of predicting future health events is associated with low accuracy, therefore the effectiveness evaluation of the system was based on recall and precision. Recall measured the percentage of target events that were successfully predicted (i.e. the reverse of “false negatives”) while precision reflects the percentage of predictions that were correct.

A target variable is a time series instance that includes a "major health event" notification generated by the RPM device detection algorithms (i.e. “AFIB”, “TACHY”, “BRADY”, “VF”, “SVT”, “ASYS” or “ABF). Given that a cohort C_i is a timeordered sequence of *m* successive instances (*m* ranges between 3-15 in LAB proceedings), the Target Variable Cohort (TVC) is defined as a set of *m* successive valid instances that includes at least one target variable instance. Note that if the same cohort

displays medically incompatible values for HR or RR biosigns or includes contradictory "major health events", is characterized as "unreliable" and omitted.
The way of solving the "target variable" prediction problem, was to learn a prediction procedure that could correctly predict the occurrence of possible "target variables" in the “near future”, given a sequence of Recent Temporal Patterns (RTP). The prediction procedure involves matching a set of learned RTPs over the “Full Disclosure” dataset and predicting the occurrence of a target variable if the match succeeds. Critical factors of this task were definitely (a) the “warning time” which defined as the “lead time” necessary for a prediction to be useful and (b) the monitoring time which determines how far into the future the prediction extends. In our approach we set a ceiling on the value of monitoring time, beyond which the prediction is considered of low importance. Solving the event prediction problem involves two steps : In the first step, a DT algorithm is used to search the space of RTPs, in order to identify sets of “equivalent” RTPs that, individually, could effectively predict a subset of the target variables and collectively predict most of the target variables. The second step classified the RTPs from best to worst, by justifying the precision of their predictions and, consequently, prunes redundant patterns (i.e., those patterns that didn’t predict any target variables not already predicted by some “better” pattern). Based on this outcome, we produced a pool of prediction strategies, by adding one "promotable" RTP at a time and we gradually built a chart of acceptable strategies which were incorporated in the PROPHET concept.
This chart, along with the relative cost of "false negatives" (missed target values) to "false positives" (false predictions), was used to determine the optimal prediction strategy.

The task of mining RECENT TEMPORAL PATTERNS (RTP)

Definition of “support” : Given a database DBRTP = {RTP_1, RTP_2…….RTP_n} of n RTPs, and given a certain $RTP_m(r)$ where $1 \leq m \leq n$ and r is the R-grade of $RTP_m(r)$, the support of $RTP_m(r)$ in the database DBRTP is defined as the total number of sequences in DBRTP that contain $RTP_m(r)$. The lowest acceptable value for the “support” of given RTP is called “minsup”.
A key data mining problem is to detect sets of itemsets (i.e patterns), with occurrences that exceed a predefined minsup. A second and significant challenge, is the approximation of the level of a pattern’s “confidence” which is expressed as a conditional probability.
Generic level-wise search algorithm
RTP searching philosophy is based on the following principle : The support for any given temporal pattern TP can be no greater than the support for any of its subsets. This means that if one subset of TP is not frequent, then TP cannot be frequent. Therefore, finding a temporal pattern with low support, relieves us of the obligation to consider any pattern that is a superset of TP. (antimonotonicity property). This property allowed us to narrow the search space significantly, by pruning all patterns based on the current level.

EVALUATION OF THE "PROPHET" PROTOTYPE

In this section we present the results of the preliminary research towards the development of a real time predicting model, based on remotely collected vital signs with the use of a mobile, wearable WBAN.

The LAB dataset(s)

The PROPHET™ laboratory multivariate time series dataset (referred as "Full Disclosure" which is a superset of "Health Event" dataset) was divided into 3 subsets: The TRAINING, the VALIDATION and the TEST. Each instance of the dataset (observation row) contains the following variables :

Timestamp *(in dd:hh:mm:ss:ms format)*, Lead 1, Lead 2, Lead 3 *(electrical activity in μV of V1 (RA/R) , V2 (LA/L) and V6 (LL/F) leads respectively)*, Heart Rate (*beats per minute, integer)*, Respiratory Rate *(respirations per minute, integer)*, "Health Event" encoded *(displaying one of the following codes : "AFIB", "TACHY", "BRADY", "VF", "SVT", "ASYS", "ABF", "NOEVT", or it was left blank if no event occurred)*[2]

The identity of the study

- Number of participating outpatients : 10 (2 groups of 5 persons)
- Overall study duration : 260 days (FEB 2019 – OCT 2019)
- Total number of completed sessions : 50 (5 sessions per participant)
- Each session was scheduled to last 12 consecutive days. During each session, the participants ("bearers") were wearing a Wireless Body Area Network (WBAN) pack, comprising of a the RPM device and a common MiFi router
- Each session produced approximately 370,000 instances per outpatient, resulting in a total of approximately 18.5 million instances throughout the study .
- For each patient, an idle period of 48 days was allowed between 2 consecutive sessions ("null" period)
- Immediately after the end of each session, the doctors of the research team were performing an overall assessment of the health status of each participant

Close to 70% of total instances population, was allocated for the use of the TRAINING and the VALIDATION subsets.

[2] Tachycardia (TACHY), Bradycardia (BRADY), Ventricular Fibrillation (VF), Atrial Fibrillation (AFIB), Supraventricular tachycardia (SVT), Asystole (ASYS), Bradyarrhythmia (ABF atrial brady-fibrillation), Patient activated event recording (NOEVT)

The LAB implementation

DEFINITIONS EXTENSION

COHORT : A cohort is already defined as a group of *r* successive instances registered in the "Full Disclosure" dataset. Observations included in a cohort component instances are used for calculating its R-grade value, in direct and exclusive dependence on the corresponding observations of the cohorts that "surround" a particular target variable, among those detected in the "Full Disclosure" dataset (as described in the "R-grade calculation model" paragraph). Any cohort that contains one or more instances with HR-class HIGH or LOW is labelled as HR-H or HR-L respectively. If a certain cohort contains both HIGH and LOW HR-classes, the system ignores it and excludes it from participating in any tasks of "predictive patterns detection" process. These cohorts are labelled as “unreliable”. Any cohort containing one instance with RR-class HIGH or LOW is labelled as RR-H or RR-L respectively. If a certain cohort contains more than one instances with RR-class ≠ ☒, is labelled as “unreliable”.

PILOT TVC (Pilot Target Variable Cohort) :. A TVC used for the training of the predictive algorithm, is referred as “pilot TVC”

TVAC : A “Target Variable Adjacent Cohorts” pool (TVAC) is defines as a set of *k* successive cohorts that include at least one "major health event" (i.e. “AFIB”, “TACHY”, “BRADY”, “VF”, “SVT”, “ASYS”, “ABF”, or “NOEVT”). Assuming that the 1st (if only) "major health event" is detected in instance S_i, then the 1st cohort of this particular TVAC starts with instance $S_{(i-(k/2)*m)}$ and the last (k^{th}) cohort ends with $S_{(i+(k/2)*m)}$. The value of *k* varied, depending on the “pattern dimension” (i.e. the number of cohorts included in a temporal pattern) as this has been defined, for each distinct lab trial

RTP TAG RENDERING : Assuming that in the “Full Disclosure” TRAINING subset we detected a target variable (t_var_i) in the instance registered in row r_i, By definition there is a number of cohorts equal to the quotient{(i-k)/k}, that preceded the 1st cohort of the TVAC pool of this particular target variable. Let also assume that all preceding cohorts are valid (i.e. not “unreliable”). We set a new dataset $[C(t\_var_i)] \subseteq C$, containing only those cohorts that their abstract state (class) relative to t_var_i, was better than “L” (low). Let *n* be the number of cohorts in $C(t\_var_i)$. The prerequisite for a temporal pattern to be considered as a recent temporal pattern in regard to any specific TVC, is for the time frame between the start time of the given temporal pattern and the detection time of the particular TVC to be less than 36 hours.

R-GRADE : The R-grade of a cohort represents the morbidity relevance of this particular cohort against each pilot TVC in the TRAINING and VALIDATION subsets or against each TVC set in the TEST subset, to be used as reference TVC for any distinct type of registered major health event. For consistency reasons in model's calculations, each TVC in the "TRAINING" and "VALIDATION" subsets, was given an "R-grade" = 4

“CONTROLLER” : The “controller” is a tool of the PROPHET prototype that its main task is to supervise the operation of the predicting model, for detecting signs of possible impending problems and either taking corrective actions, whenever these are dictated by unequivocally defined and non-misinterpretative instructions, or disclaims responsibility, simply notifying the system admin. The “controller” deals – among others – with a number of quality issues related to the model’s operation, such as monitoring the frequency of unreliable/invalid temporal instances, early warning for possible extreme deviations of the interim sampling results against the respective reference values, warning of upward trends of high variability cohorts population which could be detrimental to RTP reliability, etc

LAB implementation workflow

Tasks carried out during the implementation of PROPHET prototype in the LAB environment, by applying the predicting model’s concept, as presented in the previous chapters of the present report :

1. Choosing an outpatient P_{λ} with a known cardiovascular disease, to whom a specific risk factor has been attributed, as evidenced by the medical examinations held a few days before the start of the 1st session (distinct types of attributable risk factor : "low –", "low", "low +", "intermediate", "high –", "high", "high +", "critical")
2. Invoke the "Full Disclosure" dataset of session #1 for the outpatient P_{λ} (referring as $D(P_{\lambda},1)$)
3. In the TRAINING dataset we located those instances that carry identical target variables, i.e. instances displaying identical “major health events”. We then marked each health event group (HEg) of identical target variables, with a unique indicator (AFIB group, SVT group, etc). In cooperation with the MDs research team, we processed the TVCs of all members in every HEg, to produce one “combo” TVC for each HEg. Each “combo” TVC was used as the reference TVC for every distinct type of health event (i.e. the TVC of the pilot target variable)
4. Next, a pilot target variable ($tvar_p$) was selected from $D(P_{\lambda},1)$ (i.e. an instance in $D(P_{\lambda},1)$ containing a “major health event” code ≠ <blank>). Definition of a typical tvar differs for each major health event, based on the medical assessment of the associated observation values. Taking “AFIB” as example, the MD team processed all the appearances of this particular major health event in the TRAINING subset along with their associated TVCs, in order to suggest a representative pilot TVC which was used as the reference “target variable” ($tvar_p$) for training the algorithm of the predicting model
5. Searching $D(P_{\lambda},1)$ to locate those cohorts that include the selected $tvar_p$ in its first appearance ($tvar_{p.1}$) in $D(P_{\lambda},1)$ ($C_m(tvar_{p.1})$, m=1-n, n=number of cohorts containing $tvar_{p.1}$

5.1.1. Setting up a table *m* rows X 8 cols containing all *m* consecutive instances which include the instance of the selected target variable (t_var). The column labels are : TIMESTAMP / LEAD#1 /LEAD#2 /LEAD#3 / HR / RR / EVENT. This table is referred as TVC table (Target Variable Cohort table).

NOTICE : If any cohort in a pool of Target Variable Adjacent Cohorts proved to be unreliable, the particular TVC is also labelled as "unreliable" and the system is instructed to ignore this particular target variable, moving on searching for the next reliable TVC

5.1.2. Calculating the R-grade of all cohorts in $D(P_{\lambda},1)$ based on the particular TVAC that includes the pilot TVC

R-grade calculation model

After selecting a target variable in the “Full Disclosure” dataset and defining the TVC for the specific target variable, we generate a table [*m* rowsX8 cols] in which all values of TVC’s constituent instances are transferred. Any “reliable” (valid) TVC is labelled as follows :

- If it contains at least one instance in which HIGH or LOW HR is detected, the TVC is labelled as HR-H or HR-L respectively, otherwise is labelled as HR-N (for normal)
- If it contains one instance in which HIGH or LOW RR is detected, the TVC is labelled as RR-H or RR-L respectively, otherwise is labelled as RR-N (for normal). Note that a valid TVC cannot contain more than one instance carrying a RR observation, as the time between the start and the end of a TVC is less than one minute

Then we start with the 1st cohort of the “Full Disclosure” dataset (1st examinee cohort) which contains the *m* first instances registered in the dataset. The following table depicts a visualized layout of any cohort (for *m*=10) :

Session day	Time	LEAD1	LEAD2	LEAD3	EVENT	HR class	RR class
1	00:00:03	-0,12695	-0,22461	-0,09766	NIL	N	N
1	00:00:06	-0,12207	-0,21973	-0,09766	NIL	N	N
1	00:00:08	-0,11719	-0,20508	-0,08789	NIL	N	N
1	00:00:11	-0,10742	-0,20508	-0,09766	NIL	N	N
1	00:00:14	-0,08789	-0,19043	-0,10254	NIL	N	N
1	00:00:17	-0,08301	-0,18555	-0,10254	NIL	N	N
1	00:00:19	-0,10254	-0,18066	-0,07813	NIL	N	N
1	00:00:22	-0,10742	-0,17578	-0,06836	NIL	N	N
1	00:00:25	-0,12207	-0,19043	-0,06836	NIL	N	N
1	00:00:28	-0,12695	-0,20508	-0,07813	NIL	N	N

Where : LEAD1, 2, 3 display the electrical activity of each lead in mV, EVENT displays the code of a major “health event” if present, otherwise displays “NIL” and HR class / RR class display N for heart rate / respiratory rate within normal limits, while “H” stands for HIGH and “L” for LOW rates respectively.

For the experimental calculation of an examinee cohort's R-grade, the following data are used :

a. The arithmetic means of the absolute values of Pearson's correlation coefficient formula, that is applied on the observations of each of the 3 LEADs of the given examinee cohort, against the corresponding observations of each of the 3 LEADs of each of the *m* instances of the pilot TVC.
b. The relevance degree between HR and RR measurements contained in the examinee cohort and the corresponding measurements in the pilot TVC, which is depicted in the following half matrices :

	HR-N	**HR-L**	**HR-H**
HR-N	0,50	0,25	0,25
HR-L		0,5	0
HR-H			0,5

	RR-N	**RR-L**	**RR-H**
RR-N	0,50	0,25	0,25
RR-L		0,5	0
RR-H			0,5

The R-grade value is calculated by the following formula :

R-grade = $\sum(x_i * w_i)$, i=1,5, where x_i are the variables described above and w_i are their relative weights of importance ($w_i \leq 1$). The PROPHET prototype introduces a "health event" predicting model that operates using personalized patient-specific parameters. In this extent, the aforementioned weights of importance could be varied to reflect the current health status of each RPM service user

By the end of this process, each valid cohort in the “Full Disclosure” dataset, had been awarded an individual R-grade, which was used for classifying all cohorts according to their ”morbidity level” in respect to the selected (pilot) TVC.

6. Define a temporal abstraction pattern

Assuming, for the sake of simplicity, that in the R-grade formula, all $w_i = 1$, $\forall i$, then the range of R-grade values ranges between 0 and 4 (the theoretical value of 4 is achieved when $x_1 = x_2 = x_3 = 1$ and $x_4 = x_5 = 0,5$). Using the R-grade value as a driver, we can assign an abstract state (class) to each examined cohort as follows (given that for any TVC R-grade=4) :

Cohorts with R-grade < 0,50 are considered to display a "very loose" morbidity relevance (VL) to the pilot TVC, cohorts which their R-grade ranges between 0,50 and 1,10, display a “loose” relevance (L), between 1,10 and 1,70 the class is marked as "average" (AV), while between 1,70 and 2,40 is considered as "high" (H) and between 2,40 and 3,00 is considered as "high+" (H+). Finally, an R-grade > 3,00 characterizes the class as excellent (E)

Therefore, a state is an abstraction of any given cohort. Given that a temporal pattern TP in the PROPHET dataset, is made of one or more valid successive cohorts C_i and each cohort has been assigned with exactly one abstract state (class), a temporal abstraction pattern is defined as $TP_t = (ASc_1, \ldots, ASc_k)$,

where ASc_i is the abstract state (class) of the i^{th} cohort included in temporal pattern TP_t.

7. Define dataset C : We represent every cohort C_i (i=1,n/r, where n=number of instances in the sample and r=cohort dimension) in a dataset C as a Multivariate State Sequence (MSS) without including any attributes that link any cohort with any particular target variable (i.e "R-grade" and "Abstract State") Each record in dataset C displays a cohort's row indicator along with 2 timestamps (start – end of cohort's instance). Define temporal patterns (i.e. group of cohorts) possibly associated with the pilot target variable.
   In implementing the PROPHET LAB test plan, we experimented with different time frames for the duration of a temporal pattern. In a reverse approach and assuming that we set the duration of a trial temporal pattern to *n* minutes, we conclude that a typical temporal pattern should comprise $(\lambda * n)-1$ consecutive cohorts, where λ is an integer between 1–7, depending on the dimension *r* of the choice cohort (r=3-15 instances of 2,000 msec each). Based on this assumption and taking in account the definition of "recent temporal pattern", we built a new dataset [$RTP(t\_var_i)$] containing only those temporal patterns derived from all possible combinations of the cohorts in dataset $C(t\_var_i)$, that, at the same time, were matching the definition of "recent temporal pattern". Next, a "combined abstract state" (combined class) was assigned to each entry in $RTP(t\_var_i)$, for reflecting the relevance of each recent temporal pattern to TVAC. To assign a unique "combined class" for any recent temporal pattern in $RTP(t\_var_i)$, we used the following procedure : Let R_j be an entry in $RTP(t\_var_i)$ dataset, then R_j contains $\lambda * n$ consecutive cohorts (C_j-$C_{(j+2*n)}$), each of which is labelled with a unique ASc_i where $ASc_i \in$ {VL, L, AV, H, H+, E}[3], Each ASc_i derives from the R-grade that has been assigned to the corresponding C_i of any particular R_j, where $R_j \in RTP(t\_var_i)$, as defined in chapter "LAB implementation workflow" (§6)

8. In searching for frequent temporal patterns, we chose the "apriori" algorithm, which is a typical example of level-wise algorithm capable of discovering any type of pattern that satisfies antimonotonicity. The implementation of the algorithm was comprising the following steps,,in a generalized approach : (1) The initialization step in which frequent temporal patterns were located, using the values assigned to the minsup and *n* (minutes) parameters (2) The 2nd step where the system produced candidate temporal patterns (TP) by using the frequent TPs *o*f the previous step. (3) Then, the TRAINING subset of the {X} DATASET was scanned for entities of candidate temporal patterns (4) Attempting to identify recent temporal patterns (RTP) associated with a certain target variable, by "winnowing" the entities of candidate TPs and retaining only those that met the requirements set out in the RTP definition (5) The 5th and

[3] VL=very low, L=low, AV=average, H=high, H+=high plus and E=excellent

final step, where the algorithm conducted a quest in an attempt to extend frequent RTPs – related to a specific target value – backward in time.

9. By setting *n*=10 and minsup=6, we built C' – a subset of C which contains only those FTPs matching the above values. Steps 1 thru 5 of §8 were followed, in building a new set R' ⊆ RTP(t_var_i). For this particular examinee case, RTP(t_var_i) ≡ RTP(t_var_p).

10. Now, R' looks like containing a number of frequent, recent temporal patterns, sharing the following common attributes :
    - Original value of *n*=10. NOTE : The *n* value characterizes any R' set (R'_{10})
    - All instances of the cohorts that set up the patterns included in R', had occurred 36 hours at earliest, prior to the pilot target variable recording
    - All patterns have a minsup ≥ 6
    - Each pattern is rated with one "combined class" in regard to the pilot target variable, displaying one of the following grades : VL (very low), L (low), AV (average), H (high), H+ (high plus) and E (excellent)

10.a. Additional R'_n sets were created, each one with a different *n* size (n=number of cohorts per pattern), n ranging between 20 and 40.

A record in any R'_n comprises the following data fields : Pattern ID (Patient ID, Session number, pattern s/n), number of cohorts in this pattern, pattern's class, number of appearances, timestamps [session day/time] of 1st, 2nd...., last appearance, timestamp of pilot target variable appearance.

THE PREDICTING MODEL

The PROPHET concept is to introduce a reliable model for predicting upcoming "health events" with the use of an RPM device which continuously monitors the vital signs of its "bearer", detecting out-of-normal-range HR and/or RR, as well as key cardiac arrhythmias. In achieving this goal, a more or less, traditional linear regression model was used.

The evolution of time intervals that mediate between the successive appearances of a given RTP – member of a specific R'_n, till the occurrence of its associated target variable, proved to be a critical factor for assessing the contribution of the given RTP in the reliability and accuracy of the predictive model. The PROPHET predicting model considers that only the RTPs with a class ≥ AV are eligible to participate in decision making. These RTPs are referred to as "actionable" RTPs.

Each R'_n hosts 4 distinct subsets of actionable RTPs, each containing only the RTPs of a particular class. So any actionable RTP ∈ R'_n, can only belong to one of the following "class groups" : AV, H, H+ or E.

<u>The LAB trials</u>

A scoring algorithm was developed to rank any RTP in any R_n', according to the proximity of its morbidity relevance to the pilot TVAC, based both on the proximity of the "morbidity relevance" of each RTP's component cohorts to the pilot TVC, as well as on the variability of this latter attribute.

Based on this, each RTP of class ≥ AV, was assigned an R-grade equal to the value of R = $[(\sum x_i)/n\ (i=1\text{-}n)] / [(\sum |x - \bar{x}|)/n]$. This value is then referred to as RTP_X "pattern R-grade", for any given x.

Using the results of task 10.a above, we experimented with the linear regression analysis (LRA) model on a variety of *n* values (i.e. the dimension of R'_n), against the selected pilot target variable ($tvar_P$). For each distinct "class group" in any "n-dimension" R'_n, we applied the LRA model to calculate the values of the critical statistical parameters required to evaluate the predictive potential of each specific "class group", i.e. the R^2, the Significance F and P-values, the coefficients and the trend of the regression equation representing the temporal evolution of the appearance of all its members, on the 36 hours interval prior the detection of the pilot target variable. In this approach, the explanatory (independent) variables are the integer hours h_1, h_2,..... h_{36}, of the 36 hours trial time series and the predictor (dependent) variable corresponds to the number of times an RTP of a particular class appears each hour within the trial period (i.e. density of the occurrences of the specific RTP class, along the evolution of the independent variable). By extending the test process in the VALIDATION subset, it proved to be more constructive for the evaluation of the model, to quantify the "class density" by awarding a grade to each "trial hour" resulting from the formula : C_{AV}*1,50+C_H*2,50+C_{H+}*4,00+C_E*6,00 where C_i is the number of RTPs registered during this particular "trial hour" for i= AV/H/H+/E. The introduction of an equivalent feature for quantifying the evolution of the "class density" across time series, allowed the use of numerical thresholds in evaluating the contribution of the "class density" parameter in the sensitivity of the predictive model. Nevertheless, during LAB trials, the "pattern R-grade" feature proved to be more consistent compared to that of the "class density", as it uses the ambiguity level in scoring any given RTP, by introducing the measurement of variability as an increasing/reducing factor of the overall pattern's score. In this approach we used the quarters of the hour as independent variables of the LRA model (thus x=1-144 for the 36 hours trial), while the "pattern R-grade" of each RTP registered in the trial period, was used as predictor (dependent) variable, regardless of the value of its corresponding "class density" parameter.

Conducting numerous trials of the LRA model, using all 4 "class groups" of RTPs in the TRAINING subset, led to the narrowing of options regarding the RTP dimension that serves best the optimal feasible solution for maximizing the results of the predictive algorithm

The key experimental parameters that were differentiated in the process of evaluating the effectiveness of the predictive model, were : (1) The time span preceding the appearance of the pilot target variable in the TRAINING subset (the trial duration) (2) The dimension n of an RTP (n=number of component cohorts) (3) The cohort's *m* dimension (number of component instances).

In the RTP population of any given 36-hour trial time series, each actionable RTP class is represented by a number of members, with each occupying exactly one of the 144 time slots that define the trial's lifetime. Let a class AS (AV / H / H+ / E)

numbers k members in the RTP population of a trial set in the TRAINING subset for a certain pilot tvar and its relevant pilot TVC. Let also tj (j=1,k) the time slots that host the RTPs of the particular class and t_0, t_{143} the starting times of the 1st and the last RTP respectively.

By experimenting with the dimension *n* of the RTPs $\in R'_n$, we arrived to the conclusion that the optimal feasible value for n=30, based on the values of the statistical parameters, as analyzed above. Given this and the fact that the duration of a cohort is ~ 30 secs, is concluded that the duration of a typical RTP is 15 mins and therefore, the 36 hours trial comprises 144 RTPs

Regarding the *m* parameter, the results of LRA model application, proved to be less accurate as the dimension *m* was decreased (trials were performed for m=3, 7, 10, 12, 15). The LRA statistics were clearly improved by increasing the *m* dimension. Furthermore, experimenting with the time span preceding the appearance of the pilot target variable, proved that the increase of preceding time span to 48 hours (i.e. expanding the population of the RTP set, to cover a 48 hours period instead of 36), was directly contributing to the increase of the variability measurements, which is equivalent to reduced reliability of the "pattern R-grade", calculated for each RTP "older" than 36 hours.

A summarized report on the results of evaluating the trial's alternative building blocks, is given under APPENDIX A

<u>The validation process : Validate the algorithm with VALIDATION subset</u>

Up to this point we came up with a consistent approach to designing the core entities of the PROPHET prototype, namely the modelling of the predictive algorithm, as well as the functional framework and the key attributes for the choice temporal pattern. These core entities were built around a single “health event” (i.e. the pilot target variable), detected in the TRAINING subset of the "Full Disclosure" dataset. Next we had to validate the functionality of PROPHET model over other health events detected in the same TRAINING dataset and in the same or subsequent sessions, provided that the patient's health status remains relatively unchanged during the examinee session in regard to the piloting session. In this context, the following steps had been taken to validate the effectiveness of the predictive algorithm :

- Define a set $\{V\} \subset$ VALIDATION dataset, comprising 36 continuing hours time series, to be used as trial period for $tvar_2$, after making sure that there is at least one major health event registered in {V}
- Check validity / reliability of all instances in the said trial period, dumping any unreliable / invalid entries
- Assembly cohorts after setting *m* dimension to 15
- Select a tvar ($tvar_2$) in {V} that belongs to the same HEg with that of the pilot tvar ($tvar_p$), after ensuring that $RTP(tvar_p) \cap RTP(tvar_2) = \emptyset$.
- Create respective TVC and TVAC entities for $tvar_2$
- Define and identify TPs registered in {V}, build $RTP(tvar_2)$ subset in {V} and run the procedure described in §9 of “LAB implementation workflow” chapter.

- Compare the TVC of $tvar_p$ to the TVC of $tvar_2$, expressing their differences as percentage
- Calculate the R-grades of all RTP($tvar_2$) members
- Apply the “pattern R-grade” algorithm on {V}
- Evaluate the results of LRA model
- Compare the LRA results for $tvar_2$ with those of $tvar_p$, taking in account the figure representing their differences (%)
- Draw inferences

The validation process was repeated for all types of major health events in {V}, displaying the following results :

Health event		AFIB	TACHY	BRADY	VF	SVT	ASYS	ABF
Number of runs		4	1	1	2	2	2	3
Average percentage of differences from respective pilot tvar ($tvar_p$)_[a]		18,30%	5,50%	6,20%	22,00%	23,40%	20,50%	19,20%
convergence factor ($tvar_r$ vs $tvar_p$)_[b]	RUN 1	90,00%	88,72%	88,79%	90,37%	90,51%	90,22%	90,09%
	RUN 2	65,00%			66,30%	66,79%	65,77%	65,32%
	RUN 3	78,00%						78,20%
	RUN 4	75,00%						

[a] After applying the model for both $tvar_p$ and $tvar_r$ ($\forall$ r), we analysed the differences found in the respective LRA results related to key statistical figures. These differences were consolidated under a single value (%), which was used for adjusting the result of the convergence factor assessment.

[b] The convergence factor gives the percentage at which the results of the application of LRA model for $tvar_p$, converge to those of the respective health event $tvar_r$ (r : “AFIB”, “TACHY”, “BRADY”, “VF”, “SVT”, “ASYS”, “ABF”), separately for every run.

Following the validation process, a reverse procedure was developed to improve the agility and the cohesiveness of the choice algorithm, based on a heuristic approach that emerged as a result of the validation process iterations, on the way to maximize the convergence factor values

The predictive algorithm

The evaluation process of the trials' results, provided the necessary information for selecting the optimal, functional predicting model as well as the values of its critical parameters. In this context, the process revealed that the “pattern R-grade” algorithm displays stronger predictive potentiality comparing to the respective performance of its competitor that implements the "density class" approach. Furthermore, for every $Ri \in RTP(tvar_p)$, i=1,n, n≤144, the threshold for the confidence level of their variability factor was set to 0,09. This assumption led in excluding from the model those patterns that displayed a variability factor higher than the set threshold, thus leading

to a subset of the RTP($tvar_p$), comprising patterns bearing an acceptable level of variability, which were then used in the validation repetitive process. Due to the somewhat low values of the convergence factor recorded sporadically during validation phase, the reverse process was invoked to adjust the model parameters and consequently eliminate the convergence deviations. On this route, we experimented with the RTP subset by disembarking a number of patterns that didn't qualified as a result of the aforementioned corrective process. After elaborating the results of the "reverse process", we came up with a "temporarily final" version of the "pattern R-grade" algorithm, that was then implemented in the TEST subset environment, to assessing its ability in predicting upcoming "health events", based on its training. The parameters of the "temporary final" version of the algorithm, were configured as follows :

- Cohort dimension (m) : 15 instances
- RTP dimension : 30 cohorts
- Minsup : 6
- Number of actionable temporal patterns in the 36-hour timeframe : ≤ 144
- Polynomial for calculating the "actionable RTP R-grade", $R = [(\sum x_i)/n]/\sum|x-\bar{x}|)/n$ i=1,n, n=15 (cohort dimension)
- LRA function $y=f(x)$
  y = actionable RTP "R-grade", x is a natural number representing the temporal order (order of appearance) of any entry in RTP() and $a_1...a_\Lambda$ coefficients determined and fine-tuned during the application of the model on the TRAINING and the VALIDATION subsets of the "Full Disclosure" dataset
- Area Under the Curve (AUC) $=\int_1^n f(x)dx$, n≤144, the core figure used in predicting the time of possible appearance of a major health event

REFERENCE MAJOR HEALTH EVENT

Atrial fibrillation (AFIB) was selected as the "reference major health event" to be used with the simulator. Target variable features that were set for a single type of "health event" (AFIB), to assessing the efficiency of the model with the VALIDATION subset :

- Number of instances displaying AFIB health event (target instances) : 60
- Number of non overlapping cohorts (TVC by definition) comprising at least one target instance : 9
- Number of TVAC pools, that the intersection per pair is the blank set : 4
- Weighted average of TVACs' R-grade (using the number of target instances included in each TVAC pool as weight factor) : 3,8
- AUC target value of the reference LRA function y=f(x) : 250 (Calculated using the trapezoidal rule), where x = 1 to n, and n being the temporal order of the RTP entry, within which the AFIB event was detected (n≤144)

<u>Simulating the "real world" environment</u>

The real time operation of the PROPHET prototype, was simulated with the use of the RPM data time series, registered in the TEST subset of the "Full Disclosure" dataset

Preparatory actions : The following steps were taken when preparing the simulator of an RPM session for a specific patient P :

Set "driver" temporal pattern : Assuming an ongoing real time RPM session, we define as the "driver actionable temporal pattern", the 1st set of k consecutive m-dimension cohorts (currently m=15, k=30), matching all relevant criteria set in the previous chapters and displaying an R-grade value ≥ $(R\text{-grade})_{min}$, where $(R\text{-grade})_{min}$ was determined during the training phase of the model (for PROPHET prototype trials $(R\text{-grade})_{min}$ set to 1,40).

Set "control cluster" : Next, we introduce the "control cluster" entity (CC_i, i=1-(n/l)), as a set of l successive actionable TPs, each entity starting with TPi, ending with TP(i+l) and not overlapping with the next CC that starts with TP(i+l+1) and ends with TP(i+2l+1), where l is the CC dimension, set to 4 during simulator.

Set the alarm threshold : The model's training process, dictated a "health event alarm" threshold, linked to the AUC of f(x), calculated for x=1 and x=144, assuming that in the TRAINING subset, the target variable (i.e. the "health event") is detected in the RTP corresponding to x=144

False alarms : By definition, an alarm message is labelled as FALSE POSITIVE when the alarm procedure is unnecessarily activated, warning for a "health event" that will not going to occur within the set "tolerance time limits". On the other hand, PROPHET "controller" registers a FALSE NEGATIVE incident, when a "health event" is detected which the model failed to predict within the set "tolerance time limits"

Set the tolerance time limits : Let x_a being the actual time slot in the active predictive period, within which the actual "major health event" occurred and x_p the time slot in which the alarm procedure was activated, in warning of an impending "reference health event". Then, the figure $d = [(x_a - x_p)*15]$, represents the average temporal distance (in minutes) between the actual event and the corresponding alarm. A d<0 means that the alarm was triggered earlier than the actual event. In this case the tolerance time limit was set to (-240) mins, so if d is less than (-240) the alarm is labelled as FALSE POSITIVE. A d>0 means that the alarm was set off after the actual event. For this case the tolerance time limit was set to 15 mins, therefore, if d is greater than 15 the alarm is labelled as FALSE NEGATIVE.

Set the predictive algorithm environment : Build a custom TVC (i.e. a TVC adapted exclusively for patient P), with *m* = 30, for being consistent to the optimal choice, for every major health event HE_t (t : "AFIB", "TACHY", "BRADY", "VF", "SVT", "ASYS", "ABF", or "NOEVT") that could arise during the RPM session of patient P. The values of the custom TVC component instances' parameters, could either derived from a prior exploratory RPM session of this particular patient (if any) or set by the MDs' team, based on other patients' RPM data, displaying similar EHR[4] image with that of P. The TVAC pool associated to this particular custom TVC would be created using a similar – as above – approach

Set the "entry timestamp"

[4] EHR : Electronic Health Record

Timestamp definition : a unique value, marking the moment in time that WBAN recorded a specific set of vital signs, concurrently measured

Basic element in understanding the workflow of any out-of-the-LAB application of the predicting model, is the fact that the temporal pattern (TP) been registered in the 1st time slot (i.e. x_1) of a real time session, could not treated proportionally with the TP registered in time slot x_1 of the TRAINING model. At the starting point, the system locates the time slot (i.e. the independent variable x), optimally corresponding to the R-grade value of the driver temporal pattern, using the nearest R-grade value of the training R-grade pool, to that of the driver. We define this time slot as the “entry timestamp”.

<u>The predicting model workflow</u>

REFERENCE SESSION : The LRA defines a and b coefficients for y=a*x+b (y=R-grade), where x the temporal order (order of appearance) of actionable RTPs only. During classifier learning, the "TRAINING" session was set to start 36 hours prior the target health event (AFIB) detection. This means that the upper limit of the integral used for calculating the reference AUC value, was equal the number of actionable RTPs counted in the "TRAINING" session, till the occurrence of TVC (or TVAC for homogeneity reasons). In reference session we know that the AFIB event occurred 36 hours since starting time slot. In this extent, the upper limit of the integral for the reference session will always be ≤ 144 (36h=36*60mins=144 quarters, using 15-minute time slots as independent variables of the LRA model). When AFIB showed up, x=j (j≤144), y_j=R-grade of “target” RTP, i.e. the RTP carrying the AFIB health event, therefore $AUC=\int(a*x+b)dx$, x=1,j. The AUC value for x=j is defined as the “target AUC” value. Given the “target AUC”, we proceeded in setting the sensitivity limits prior starting the real time RPM session (the actual session).

ADJUSTMENT FACTOR (R-factor) : This is a real number <1, which is used in adjusting the actual R-grade values in order to maximize the predicting accuracy of the model, regarding the proximity of actual AUC value (i.e. the AUC value of f(x) the moment of the target health event detection in the actual session) versus the target AUC value, registered during the reference session. Steps taken in calculating the R-factor : For each actionable RTP we define RTP/HR as the average of the heart rate (HR) measurements included in the instances that “belong” to this particular RTP. Respectively, we define as RTP/RR the average of the respiratory rates (RR) measurements included in the instances that “belong” to this particular RTP. For any given RTP, if both RTP/HR and RTP/RR are greater than the respective upper limits as set by the MD or both are lower than the respective lower limits, its R-factor is set to λ. If only one of these values was out of limits, the RTPs R-factor was set to λ’. Should both values ranged within valid limits, the R-factor set to zero. We considered that the R-factor of Reference session equals 1. Then, the R-factor for any examinee RPM session is resulting as the price of the polynomial $R=[\sum(RTP/R)\lambda_i-\sum((RTP/actual)\lambda_i+\sum(RTP/R)\lambda'_i-\sum((RTP/actual)\lambda'_i]/1000$. The above procedure was a result of experimenting with different sessions of the VALIDATION subset, in conjunction with the reference session of the TRAINING subset. When it comes for

actual RPM sessions, the model uses the R-grade values of the already registered *n* RTPs (let *n*=10) for projecting the R-grades values of future RTPs that are expected to be recorded during actual session. Subsequently, the system applies the aforementioned adjustment procedure on the projected R-grades. The actual AUC value is then calculated using the adjusted values of the projected R-grades.

Although R-grade and R-factor figures are neither related nor interdependent as core constituents of the predicting model, their combination yields medically critical information, on which the MD team was based for setting the role of R-factor in adjusting the parameters of the predicting model in real time operation.

Assuming that the matching independent variable x = n (n≤144), the PROPHET prototype begins applying the model for the entry timestamp onwards, building the relative RTPs in real time and assigning the corresponding adjusted R-grades to the actionable ones only. In parallel, the "Controller" is in charge of calculating and checking in real time, the ever evolving R-factor of the actual session. A high R-factor denotes unstable session environment, which could result in discriminating the reliability of R-grade regarding its predictive role. Should this factor is constantly increasing to reach a predefined alarm ceiling,, the "Controller" issues a warning message, leaving the decision for possible corrective actions to the PROPHET admin team. Another option in using the R-factor, is for the controller to put the model operation on hold, when the factor's value exceeds a warning threshold and finally discontinue the process, should the value crosses upwards a critically high limit.

The setting of the driver RTP R-grade in an actual session, yields the corresponding x value, let this be equal n. Then for every newcomer CCn, the system calculates in real time the AUC value of y=f(x) for x=1 and x=n, then for x=1 and x=n+1, and so forth. The alert mechanism is activating when AUC value of f(x) reaches the alarm threshold, which for PROPHET prototype trials was set to 250 (with a sensitivity factor ranging between -5% and 5%).

Simulator results

The aforementioned procedure was applied in the TEST data subset which was used for evaluating the machine's ability in predicting upcoming "health events", based on its training. The TEST subset comprised about 80 hours of continuous recording of vital signs, with the use of a WBAN worn by a single patient throughout the session.

The use of simulator was confined in evaluating the model against the reference major health event (AFIB), only

Results in numbers	
Simulator duration (hours)	80
Number of time slots	323
Number of complete RTPs	320
Number of control clusters (r=5)	64
entry timestamp (Driver RTP)	day 5/00:00:01
Number of actual reference health events	21 (AFIB)
ALARM HISTORY	

Number of total alarms	46
Successful hits [(-240)<d<15]	14
False positives [d<(-240)]	29
False negatives [d>15] or “no alarm”	7
predicting success (%)	66,67%

Sensitivity analysis

Differentiations in test outcomes, resulting from selective changes in core parameter values :

SENSITIVITY ANALYSIS RESULTS			
PARAMETER	Sensitivity factor	Tolerance time limits (d)	"time slot"
CHANGE	Increase range limits to ±10%	Increase range limits of “d” to -300<d<20	Using "control cluster" instead (downgrading the timing detail by 1:5)
ALARM HISTORY			
Number of total alarms	80	66	124
Successful hits	16	15	17
False positives	62	48	105
False negatives or “no alarm”	5	6	4
predicting success (%)	76,19%	71,43%	80,95%

COMBINATION OF SENSITIVITY PARAMETERS	combining sensitivity factor & tolerance time limits	combining sensitivity factor & "control cluster"	combining tolerance time limits & "control cluster"	combining all 3 sensitivity parameters
ALARM HISTORY				
Number of actual reference health events	21	21	21	21
Number of total alarms	92	138	124	161
Successful hits	17	18	18	19
False positives	73	119	105	141
False negatives or “no alarm”	4	3	3	2
predicting success (%)	80,95%	85,71%	85,71%	90,48%

	R^2				Significance F			
	n=10	n=20	n=30	n=40	n=10	n=20	n=30	n=40
m=3	0,075203	0,100271	0,200542	0,22561	1,63E-06	1,7E-06	1,61E-06	1,59E-06
m=7	0,325881	0,401084	0,45122	0,436179	1,74E-06	1,77E-06	1,66E-06	1,53E-06
m=10	0,501356	0,436179	0,461247	0,481301	1,36E-06	1,33E-06	1,23E-06	1,31E-06
m=12	0,526423	0,551491	0,486315	0,461247	1,29E-06	1,32E-06	1,18E-06	1,25E-06
m=15	0,626694	0,511383	0,651762	0,526423	1,27E-06	1,33E-06	1,16E-06	1,43E-06

	P-value (γ)				standard deviation of absolute differences (%) between residuals and actual values			
	n=10	n=20	n=30	n=40	n=10	n=20	n=30	n=40
m=3	0,048823	0,050858	0,04801	0,047603	32,20%	33,60%	39,20%	40,60%
m=7	0,052078	0,052892	0,049637	0,045569	46,20%	50,40%	53,20%	52,36%
m=10	0,040686	0,039872	0,036618	0,039059	78,40%	52,36%	53,76%	54,88%
m=12	0,038652	0,039466	0,035397	0,037431	57,40%	58,80%	55,16%	53,76%
m=15	0,037838	0,039872	0,034583	0,042721	63,00%	56,56%	28,00%	57,40%

CONCLUSIONS

The results of testing the PROPHET prototype using real time RPM session data and simulating the dataflow of the session, yelded the following conclusions :

- The model success rate in predicting upcoming health events, ranges between 66% and 90% approximately, depending on the values assigned to the core parameters of the model.

The trials revealed that the success rate of the model is directly proportional to the total number of alarms generated by the system. Therefore, adapting the model's parameters to achieve the highest possible success rate, eventually leads in increasing the number of false alarms. In turn, FALSE POSITIVES congestion, triggers a reasonless mobilization and consequently causing work overload to the doctors in charge operating the RPM system, thus, leading the predictive process in becoming toilsome and ultimately counterproductive.

- For containing the oversized pool of false alarms, we needed to compromise with a lower success rate of the model. The results could be still considered moderately satisfactory with an average success rate of ~75%. This approach has resulted in a reduction of false alarms to about 80, throughout the simulator's 80-hour trial period, which is considered a marginally tolerable size.

In summary, the present report introduces the PROPHET predicting model which was built on the predictive algorithm described here and incorporated into the PROPHET prototype, with the aim of evolving into a real time “health event” prediction tool, based on outpatients' vital signs, remotely collected with the use of a mobile, wearable WBAN

Although the trials presented here, deal with only one type of major health event, namely atrial fibrillation, we reasonably believe that the procedures described, could be applied identically to any other type of major health event, without need modifying the logic or the workflow of the PROPHET model

APPENDIX A

LRA sensitivity results for all combinations of *n, m*, where *n* represents the dimension (number of cohorts) of the RTPs $\in R'_n$ and *m* the number of component instances of a cohort. Figure [1] depicts the evolution of the AUC values, within a selected period of the simulator session.

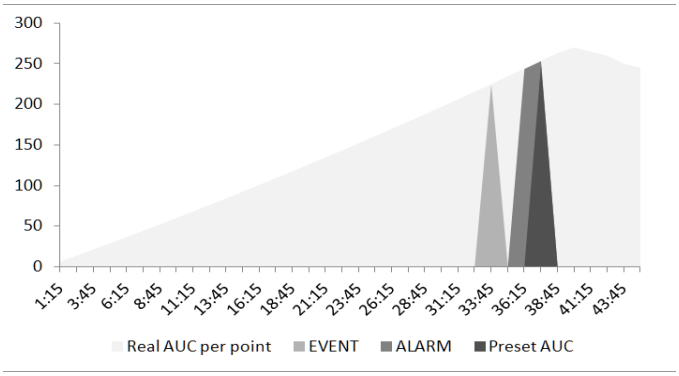

Figure 1

In terms of numbers : The preset AUC value was 249.13, while the real value when the AFIB event was detected was equal to 224.87, which corresponds to a deviation at the level of -9.74%. The model activated the alarm procedure when AUC was equal to 243.91, which would be considered a success hit should the sensitivity factor had been set to ±10%, but it would be registered as FALSE NEGATIVE in case the sensitivity factor was set to ±5%.

FUNDING
Project “PROPHET” is partially funded by the Sectoral Operational Programme “Competitiveness, Entrepreneurship and Innovation” in the framework of the Single Action “RESEARCH-CREATE-INNOVATE”

PATIENT CONSENT
We are using the guidelines of the General Data Protection Regulation (GDPR) which is a legal framework that sets guidelines for the collection and processing of personal information from individuals who live in the European Union (EU). Under GDPR rules, patients participating in the research, were notified of data collected from them and explicitly consent to that information by signing the appropriate document

CONFLICT OF INTERESTS
Author explicitly states that there is no conflict of interest to declare

REFERENCES
[1] Batal, I., Fradkin, D., Harrison, J., Moerchen, F., & Hauskrecht, M. *(Mining Recent Temporal Patterns for Event Detection in Multivariate Time Series Data. KDD : proceedings. International Conference on Knowledge Discovery & Data Mining, 2012)*