A Decision-Making Model Based on Spiking Neural Network (SNN) for Remote Patient Monitoring

Sebastien Cohen*, Florence Leve, Harold Trannois, Wafa Badreddine, and Florian Legendre

***Abstract*—Nowadays, the medical sector faces several challenges due to different factors including the increase in the number of patients to be taken care of, the economic crisis and the saturation of hospitals. Hence, hospital administrations aim to develop new strategies to handle these issues as remote patient monitoring. In this context, we propose a decision-making Spiking Neural Network (SNN) model regarding patient health conditions to integrate to patient monitoring systems. Our model offers, based on the measurements of the physiological parameters of the patient, a feedback of the patient's health condition and a raising of the alert if necessary. To do so, we construct an SNN model that represents the rules provided by a group of doctors and that allow this model to be representative of one patient. The results obtained by our model as well as those of a rule-based model validated by physicians have an error rate of less than 10%. Our goal is to reduce this error rate associating the two models and not to put the two models in competition.**

***Index Terms*—Decision-making models, remote patient monitoring, Spiking Neural Network (SNN)**

I. INTRODUCTION

The healthcare system is facing major challenges that won't be solved with our current healthcare model [1]. Indeed, several indicators attest it, for example with the ageing population that will see the number of over 60s triple by 2050.

One solution found by hospital administrations is to increase outpatient surgeries and to monitor the patient at home. This is where connected healthcare comes in, that makes technology a vital part of delivering healthcare to patients. In particular, recent medical reports predict that the number of people using home health technologies is estimated to be 12 million people [2]. In the other hand, the global IoT sensors in healthcare market was valued at C2,007.1 million in 2017, and the patient monitoring segment is expected to register an annual growth rate of 13.5% during the 2018–2026 period [3].

Thus, researchers, industry and the medical community work closely together to provide patients with efficient and reliable remote supervision systems. In this context, this work is part of Smart Angel project which is a collaboration between academics, industry and medical professionals, and that proposes a full system for remote patient monitoring. Smart Angel system aim goal is to make an accurate diagnosis of patients' health condition and, ultimately, to reduce their unjustified return to the hospital. Patients are equipped with conventional sensors and have a Personal Digital Assistant (PDA) to their disposal. The PDA will not only allow to collect the sensed physiological data but also to enable patients to answer personalized questions related to their health conditions. All these data (the collected and answered information) are transmitted to the concerned entities for additional diagnosis.

In this article, we propose a decision-making model based on a Spiking Neural Network (SNN). This choice was motivated by the suitability of the SNNs to handle our input data which are natively signal data. Indeed, SNN encodes information no longer in real values, but in the synchronization of signals called the action potential or spikes, hence collected data from sensors can be considered as spike occurring over time. In addition, SNNs are characterized with a quick adaption to specific situations and conditions and can respond based on the last. Then, our model offers a personalized decision-making regarding each patient health condition.

Our SNN model is based on the formalization of rules instructed by medical experts and reproduces the relationships between them (OR and AND logical operations). We compare our model to a rules-based model developed by Evolucare Technologies and used in Smart Angel during a first clinical study. Over 1000 patients, results show that the two models present 96% of similarity.

This paper is organized as follows: We first describe some relevant related works and generally introduce SNN in Section II. In Section III-A, we first present Smart Angel's data collection protocol per patient, and how to format the collected data. Then we present the two models for decision-making: the rules-based model in Section III-D1, and in Section III-D2 the SNN-based solution built according to the Integrate and Fire with Adaptation model. We discuss the results in Section V and finally, we conclude the paper in Section VI.

II. RELATED WORKS

In recent years, remote patient monitoring received a lot of attention from researchers and industries through new technologies and models. Hence, in this section, we present some relevant remote patient monitoring systems. We especially focus on decision-making algorithms for alert raising. Then we introduce Spiking Neural Networks (SNN) and describe their functioning.

Patient monitoring requires patient vital signs fastidious supervision. Remote patient monitoring must be as reliable as in the hospital in order to avoid needless patient returns to hospital [4]. Generally, the patient vital signs are collected through several sensors placed on, in or around human body

Manuscript received January 19, 2022; revised April 1, 2022; accepted July 22, 2022.

Sebastien Cohen, Florence Leve, Harold Trannois and Wafa Badreddine are with the MIS Laboratory - UPJV, 80000, Amiens, France.

Florian Legendre is with Evolucare Technologies, 80800, Villers-Bretonneux, France.

*Correspondence: sebastien.cohen@u-picardie.fr(S.C.)

[5]. Conventional observed vital signs are the oxygen saturation in the blood (SPO_2), the heart rate (HR), the blood pressure (systolic and diastolic), the respiratory rate, and the temperature. However, in certain cases, more parameters are needed for a decision-making about the patient's health condition as blood glucose, electroencephalogram (EEG), electromyogram (EMG) [6], urine output [7]. Moreover, other indicators as pain, nausea and vomiting or even subjective nursing appreciation have been considered as essential to detect early signs of patient sickness [8]. Besides, some additional parameters can be collected, as movement detection [9] or contextual information as support after discharge [10].

The measured data are then redirected to a specific node commonly called Personal Digital Assistant (PDA) and which can be a phone, a watch, a tablet, etc.

Mobile Health (mHealth) is becoming an important field of research [11]. Indeed, mobile applications for patient monitoring are used in many surgical specialities [12], in particular, to monitor patients at home in postoperative contexts [13]. Besides collecting the data, those systems ensure the communication between the patient and the medical entities. For example in [14], authors present Philips electronic transition to ambulatory care system (eTrAC), that offer secure vision calls, the production of a summary sheet concerning patient vital signs and remote postoperative evaluations to detect the first signs of complications. Other monitoring systems include a diagnosis component. To do so, several techniques can be used to define patient health condition, such as decision trees [15, 16], classifiers [17], fuzzy logic [18], clustering algorithms, machine learning [19].

We are interested by monitoring systems based on neural networks [10] and in particular by Spiking Neural Networks (SNN) as a decision-making model.

Originally, neural networks were inspired by the biological operation of the brain [20] and have become the benchmark for machine learning in image recognition, video analysis or even voice recognition. SNN is a recent type of neural network, which is increasingly used to solve neural computing problems and is assimilated to be 3rd Generation Neural Networks [21]. Compared to classical neural networks, SNNs encode information no longer in real values, but in the synchronization of signals called the action potential or spikes, hence collected data from sensors can be considered as spike occurring over time [22]. Mostly, SNNs have been tested on classical neural networks applications as MNIST problem [23]. However, Other applications show that it is possible to use neuron spiking to generate a mental exploration [24] (find the best possible path between two points) and then use the solution in a real robotic application.

SNNs have also been shown more secure against external attacks [25], which is valuable since we are processing medical data. In addition, the number of applications will undoubtedly explode as the industry has seized on this model, with the creation of dedicated chips. For example, in Loihi [26], the test chip designed by Intel Labs, SNNs implement parallel computations that can greatly improve computational efficiency.

There exist different SNN models, differing in the way they represent mathematically the neurons processing [27]: Izhikevich, integrate and fire, Hodgkin-Huxley, etc. The choice of the neuron model depends on whether it needs to be biologically realistic or computationally efficient.

In order to evaluate SNN models, we can perform experimentations [28] or through simulations [29]. The most common simulators are Neuron's HOC [30], NEST[31] and Brian simulator[32]. In [33], authors show that using different simulators (Neuron's HOC and NEST) give different results due to a different mathematical representation of the neurons processing. Hence, they propose PyNN, a python package, that gives a common model specification independently from the simulator. In this work, we choose Brian simulator because it gives a variety of neural model representations that we used to model our SNN.

III. Methodology

The outpatient follow-up differs from the hospital follow-up by the absence of periodic statements of the state of health, the visit of the nursing staff being less frequent with little or no evolution. For this reason, it is necessary to retrieve a sufficient amount of data regarding the patient's health condition to allow an adequate decision-making for raising alerts. In addition, a meticulous work has to be done to format the collected data to fit the chosen decision model.

A. Data Retrieval

After having a surgery, patients follow a quick training on Smart Angel system features at the hospital before going back home. Patients are invited to answer a questionnaire as well as to take physiological measurements when they are at home. The measurements are planned to be done periodically, at least every seven hours apart, but they can be done more often when it is needed (special cases of additional measurements will be described in Section III-D1). The data retrieved is taken by various sensors such as an oximeter (Heart Rate and Oxygen Saturation) or a blood pressure monitor. Smart Angel's patent [34] explain the system for the personalized monitoring of the health status. A clinical study [35] has been performed from May 2018 to September 2018 to validate the recovery of the data, in particular the transfer of data across the networks and the validity of the sensors. This clinical study was realized on 30 patients of Nîmes's University Hospital Center.

B. Data Description

The types and norm range values of the data retrieved and used by the system are described in Table I. Patients are followed up 20 days before their operation to allow medical staff to enter their average constants as well as their personal data. Only physiological and factual data is stored on the Smart Angel health data host to ensure patient anonymization on the server. The personal data of the patients are only accessible by the medical staff in the hospital database. The connection between the two databases to know the identity of the patients is not allowed by Smart Angel system. For this article, we could not use patient data from the clinical study, as this data is internal to the clinical study and is sensitive and un-sharable medical data without patient consent. We will therefore use a simulated data set that represents data possibly collected during the clinical study.

TABLE I: DESCRIPTION OF PARAMETERS IN THE DATABASE

	Name	Type	Example	Range
Physiological Data	Heart Rate (HR) (bmp)	Int	90 bmp	[40 – 200]
	Systolic pressure (SP) (mmHg)	Int	110 mmHg	[70 – 200]
	Diastolic pressure (DP) (mmHg)	Int	70 mmHg	[40 – 160]
	Mean Arterial blood Pressure (MAP) (mmHg)	Int	90 mmHg	[55 – 180]
	Oxygen Saturation (SPO2) (%)	Int	98%	[70 – 100]
Factual Data	Nausea	Boolean	True	{True – False}
	Vomiting	Boolean	False	{True – False}
	Basic treatment	Boolean	True	{True – False}
	Recourse treatment	Boolean	False	{True – False}
	Anti-emitic treatment	Boolean	False	{True – False}
	Pain score	Int	2	[0 – 9]

C. Data Formatting

We now describe how the data are formatted to detect the patient's state of health. The two models we describe in this paper have a different approach to classify the data values as "healthy" or "unhealthy", which implies that they are formatted differently.

1) Rules-based model formatting

The values domain of each data is divided into three areas: a black area, a grey area and a white area (see Fig. 1), corresponding respectively to a critical condition, an average condition and a correct condition of the patient. Those areas can be parametrized in the hospital before the surgery to fit with a particular patient, i.e. the frontiers of the area can be different for distinct patients. In Fig. 1, the values of min, l2, l1, r1, r2 and max are either configured by the medical staff during an observation of the patient 20 days before the surgery to find its average constants or calculated automatically by Smart Angel model compared to an average data concerning a representative group of the patient (depending on age, sex, ...). For example, here, l1 is 15% less than the normal and l2 is 30% less than the normal.

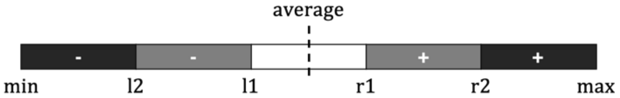

Fig. 1. Parameters areas.

2) SNN model formatting

Due to the data collection methodology (measurement is taken every 7 hours), the data used in our model is discrete. We need to format the data to handle it with SNN, that intrinsically take continuous data. The sequence of all the data retrieved by a sensor at the successive moments of measurement and then formatted and normalized is called a spike train.

We tested a shaping model which consists of shifting in time the peaks of our spike trains according to their importance compared to the nature of the input data. An input data represents a patient parameter (Heart Rate, Systolic pressure, ...) and its value indicates a risk or not for the patient. The further this value deviates from the patient's normal value, the more serious it should be considered. Our neural model is made up of several neurons linked precisely to reproduce an output similar to the one given by medical rules. Each of these neurons has one or more spike trains as input. These trains of spikes must be processed by neurons to raise or not an alert. Our formatting model is then really important to facilitate the understanding by the network of important information contained in the data. In our formatting model, data follows a succession of operations: separation, normalization and application of delaying.

a) Separation

Each data retrieved (HR, SPO_2, ...) will build two sets of input spikes. The first will take into account the measured values lower than the patient's average on this parameter, the second one will take into account the values greater than this average. That way, we can differentiate very precisely the data values above and below the patient's average. Without this separation, the data would be mixed and taken into account in the same way, but it is possible that on a given patient, a value below the average is very critical while a higher value would be more acceptable.

For example, in Fig. 2, the figure on the left represents the raw input data. Figures on the right represent the separated data. The top one takes only the data above the patient's average (here 50), for which three measurements times are concerned: t0 (60), t1 (70) and t3 (90). The bottom one takes only the data under the patient's average, for which two measurement times are concerned: t2 (40) and t4 (30).

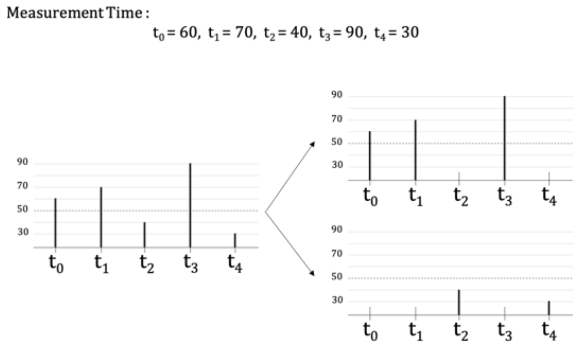

Fig. 2. Before and after data separation.

b) Normalization

The second step is the normalization of the separated data. Each parameter data is between a minimum value V_{min} and a maximum value V_{max} assigned for each patient. Let V_{avg} be the average value for a given parameter (See Fig. 1). Those three values are given accordingly to the global observation of the patient that takes place before the surgery, or calculated from the data. For example, V_{avg} is given using an average of the patient's rest values during his observation.

Each value V is normalized by the following formula according to its value compared to the value of V_{avg}:

$$V_{normalized} = \begin{cases} \left| \frac{V - V_{avg}}{V_{max} - V_{avg}} \right| \text{if } V < V_{avg} \\ \left| \frac{V - V_{min}}{V_{avg} - V_{min}} \right| \text{ otherwise} \end{cases} \quad (1)$$

In this way, each data is now between 0 and 1.

c) Applying a Delay

To correspond to the standard SNN implementation, each input value will result in a value equal to one in intensity. To relate the importance of deviation of a measured data compared to the normal value, we propose to delay by a time Δt [36] the corresponding spikes relative to the data value [37].

We chose to send a measurement peak very close to the input time (so Δt is small) if the value measured deviates very strongly from the average of the patient, indicating a high level of severity of the data, in order to raise an alert rapidly. On the contrary, a measurement peak is sent with a more important delay Δt after the input time if the value measured is very close to the patient's average. This delay Δt must be framed within 0 and a maximum offset limit Δt_{max}. It can be calculated as:

$$\Delta t = \frac{V_{normalized}}{\Delta t_{max}} \tag{2}$$

In Fig. 3, we have, for example, three different measurements that have already been separated and normalized. Suppose that this spike train corresponds to the values above the patient's normal. The first measure at T1 after normalization is near 0 (it is a measure really close to patient's normal), we then delay this spike with a big Δt. At T2, the measure corresponds practically to a limit of the patient, we delay this measurement spike with a small Δt. At T3, the measurement corresponds to a value between the patient's normal and a patient's limit. In a. we observe three measurements (normalized between 0 and 1) of some patient's parameter. In b. we observe the response of the delaying. A large delay can be seen on the T1 measurement due to the proximity to the patient's normal. Conversely, a small delay is observed on the T2 measurement due to the great distance from the patient's normal. Finally in T3, the delay is average according to the medium aspect of the measure's value.

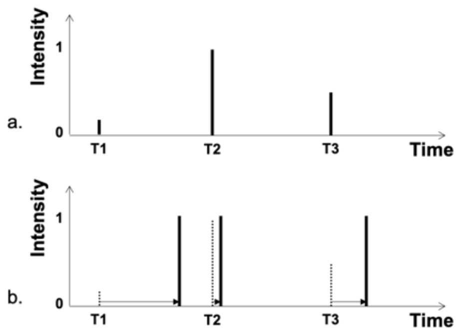

Fig. 3. Example of three delaying.

D. Models

1) Rules-based Model

a) Description

To decide if the patient's health requires a return in hospital, Evolucare Technologies has included in Smart Angel a rules-based expert system which, from the data retrieved during home caring, decides if it is needed to raise an alert or not. The rules have been validated by medical staff and followed many strict protocols, (in particular, Smart Angel use the exact same protocols used in anaesthesia departments where the study of patient parameters before, during and after an operation is critical). All the rules implemented in the system have been qualified and validated by the team of Pr. Cuvillon and Dr. Boisson of the university hospital centers of Nîmes.

As mentioned above, if an alert is raised, Smart Angel can ask the patient to retake measurements of its constants. Each alert is sent to the medical staff, but some also lead to demand a repeated measure within the following hour.

For example, let us consider a patient who had surgery on Day 0 and consider Day 1 the first day of his home observation (Fig. 4). During this Day 1, three digital appointments are scheduled on the touchpad (at 7 a.m., 2 p.m. and 9 p.m.) to check the evolution of the patient's health. At 7 a.m., Smart Angel does not detect any particular anomaly, either in the answers to the questions or in the measurements made with the sensors. No verification appointment is required. At 2 p.m., Smart Angel raises an alert following a positive response to the question "Do you have nausea?". An appointment is made an hour later to check on progress. An hour later, during this verification appointment, the patient declares having vomited. Smart Angel asks the patient if he has followed his antiemetic treatment and a new appointment is made an hour later. This time, the patient's health seems to improve. Smart Angel does not make an appointment before the next originally scheduled appointment, at 9 p.m. At 9 p.m., Smart Angel does not detect any particular anomaly. The patient can continue home monitoring for at least one more day.

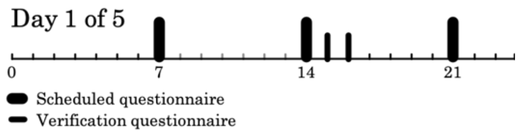

Fig. 4. Example of observation day.

b) Topology

Each rule considers one or more input data. A rule consists of checking if those data have values included in areas that have been defined as (slightly or severely) critical. Note that a given rule can be based only on one specific (critical) value area. The following figures show two simple examples: one rule taken from Smart Angel checking if the patient's heart rate is too low or too high [35] in Fig. 5, and a generic rule on two parameters in Fig. 6. In Fig. 6, if both parameters have a value included in a problematic area (grey or black areas), then an alert is sent.

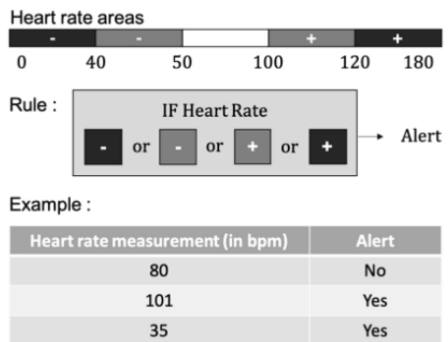

Fig. 5. Example of a rule used in Smart Angel checking the patient's heart rate.

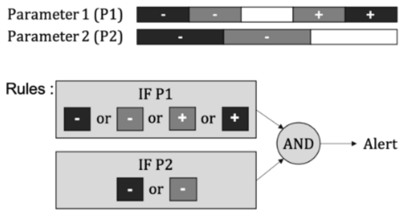

Fig. 6. Example of a generic rule on two parameters.

2) SNN Model

a) Neuron Model

Our model needs to be efficient, to raise alerts as quickly as possible and could move away from biological reality. Our choice turned then to the "Integrate and fire" model [27]. This model offers computational efficiency, has been implemented on several platforms, especially Brian Simulator [32], [38], which we used to perform our experiments and has been widely studied [39, 40]. In this model, neurons can be schematized as in Fig. 7. The global input of the neuron is equal to the sum of all weighted input data at each instant:

$$I_{total} = \sum_{i=0}^{n} I_i \times w_i \tag{3}$$

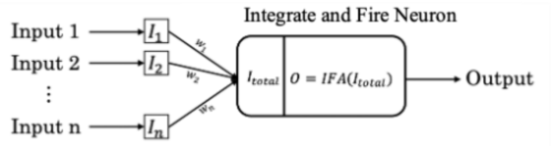

Fig. 7. Integrate and fire neuron modelisation.

This input is then processed by the Integrate and model constituted of one differential equation:

$$\frac{dv}{dt} = \frac{E_e + E_l - v}{\tau_m} \tag{4}$$

where the parameters are defined as follows:

v represents the membrane potential of the neuron. v is also controlled by a condition, representing the after-spike resetting:

$$\text{if } v \geq -23mV, \text{then } v \leftarrow -60mV \tag{5}$$

E_l represents the time scale of the recovery variable v and E_e represents the reset value of v after the spike

In order to smooth the alerts raised by the model, we define a threshold th corresponding to the optimal membrane potential of the neuron (in mV) allowing to capture if the importance of a signal is significant relatively to the number of problematic data given in entry to a neuron. Let v_{max} be the maximum conductance reached by an input peak, that can be calculated from the above equations. Intuitively, assume that the measurement of two parameters at a given time (for example the heart rate and the level of oxygen in the blood) produces two signals reaching critical values, inducing two input peaks on a neuron. Suppose also that the weights of these two signals are identical and neutral (the weights of the two input synapses are $w = 1$). To allow the model to raise an alert only if the two measurements correspond to critical values, that is produce two input peaks on the neuron, it is sufficient to place the alert lifting threshold slightly above v_{max}, for example, $th = v_{max} + \varepsilon$ (Fig. 8). Generally, to allow alerts when a number of N input signals are critical, we must define the threshold as $th = N \times v_{max} + \varepsilon$.

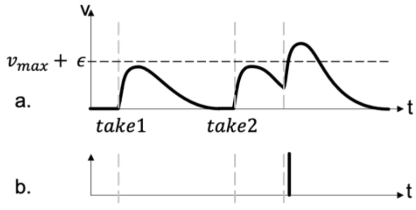

Fig. 8. Choice of the threshold $h = v_{max} + \epsilon$.

In Fig. 8, a. is the membrane potential induced by the input peaks, and b. is the corresponding output of the neuron. At take 1, in a. the membrane potential induced by the input peak reaches the maximum v_{max} , which is lower than the threshold. No response of the neuron is observed in b. On the second take, two input peaks are observed, the induced membrane potential exceeds threshold th and (at least) one response is observed.

Two other parameters influence the alert raising: The rate of re-stabilization of our conductance peaks τ_{down} and The maximum lag time of input peaks obtained during data formatting Δt_{max}.

The first τ_{down}factor (depending on τ_m in (4)) has to be minimized to avoid the superposition of conductance signals from different measurement taps causing unwarranted alerts to be raised (Fig. 9b). But at the same time, this factor must be maximized to allow a superposition of conductance signals from the same measurement (Fig. 9a). This factor can be calculated using the differential equation representing the membrane potential of the neuron.

The second-factor Δt_{max} is strongly correlated with τ_{down}. It also makes it possible to avoid the superposition of conductance signals from two different measurement taps. Let us admit that with a τ_{down} factor and two input peaks in one of our neurons carried out respectively at times take1 and take2, our resulting conductance signal becomes restabilized in a time τ_{down}, we must have at most an offset of our input peaks of $\Delta t_{max} = (take2 - take1) - \tau_{down}$. This factor makes it possible to obtain an alert more or less quickly. The lower the Δtmax, the faster the alert will be raised. On the other hand, if Δt_{max} is too low, the sum of the input signals from the neuron will necessarily cause the threshold for raising the alert to be exceeded. As the input times are offset by 7 hours (see section III), not counting the resumption of measurement, we have chosen to use a low Δt_{max} of 10ms to raise the alerts very quickly.

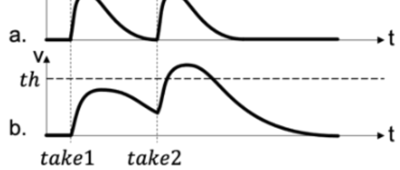

Fig. 9. Illustration of factors τ_{down}.

In Fig. 10, the solid curve represents the membrane potential obtained with an input peak of the neuron representing a measurement of a patient parameter equal to a limit value (the normalization of the data then considers this data to be 0). The dotted curve represents the membrane potential obtained with an input peak of the neuron representing a measurement of a patient parameter equal to an average value (the normalization of the data then considers this data as 1) the shift is, therefore, the greatest, namely Δt_{max}. It is observed that with a particular choice of Δt_{max}, it is possible that the membrane potential has not returned to normal before the next measurement.

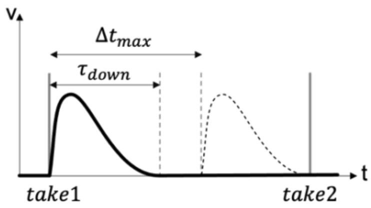

Fig. 10. Illustration of factor Δt_{max}.

b) Network Model

In our problem, the entries of the neurons are the data retrieved by Smart Angel or the output of another neuron. As done in Smart Angel rules-based model, we consider six of the physiological and physical input signals (Heart Rate, Systolic pressure, Oxygen saturation, Nausea, Vomiting, Pain scale). Those entries are represented by spike trains that are processed beforehand, as explained in Section III-C. To be able to reproduce all rules, we handle separately the values above and below the patient's average, so each of the input signals produces two input spike trains in our model.

To reproduce the decision tree behavior of an expert system with our SNN, we use a succession of neurons, each of them modelling a precise rule. Each neuron of our model uses the same parameters and proceeds as follows: Transformation of the input peaks into a conductance curve, weighted sum of conductance curves, transformation into output peaks using the Integrate and Fire with adaptation model.

To reproduce the behavior of a logical AND and of a logical OR between two input spike trains, we propose the following architecture, illustrated in Fig. 11: one or more input spike trains enter neuron N1, the output of N1 is processed along with input times by neuron N2 to suppress the noise of the network (namely to suppress the peaks corresponding to the correct values deviating from the time of the measurement). Input times' spikes trains are needed as an input of N1 in the OR model to allow a unique spike to produce an output spike.

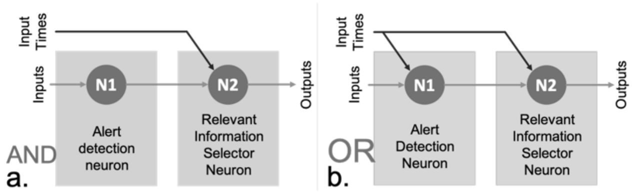

Fig. 11a: AND architecture. b: OR architecture.

AND architecture: We want to produce an output spike when two measurement input spikes represent a value deviating too much from the patient's average. For that, the data concerned by the AND are used as input to a neuron. If both values are critical, the sum of their conductance curves will exceed the firing threshold and will produce an output peak.

In the case of several parameters' measurements (at least two parameters) taken in their correct zones (i.e. their resultant spikes appear with a delay Δt after the measurement), the addition of their conductance curve can cause an output peak of the neuron N1. To filter this peak, we add second neuron N2 taking as input the output of the first neuron (N1) as well as the input times' spikes. Adding the conductances of these two inputs can give two possibilities: if the time of the output peak of the first neuron N1 is close to the time of the measurement (the input data of the first neuron were in a black or a grey area of the parameters) then an output peak of the second neuron N2 will be produced; if the time of the output peak of the first neuron N1 is far from the time of the measurement (the input data of the first neuron was in a white zone of the parameters) then no output peak of the second neuron N2 will be produced (the addition of the conductance signals will not exceed the firing threshold).

Note that, for this AND architecture and the chosen threshold, if there are more than two spike trains as inputs, it is necessary to split the rule into sub-rules to ensure that all the inputs occur in the same time interval.

OR architecture: We want to produce an output spike when one of the input spikes represents a value deviating too much from the patient's average. For this, in addition to the input spikes corresponding to the measurements, we also consider a spike of the input times as an input. As seen in Section IV, one input spike alone cannot produce an output peak of the neuron due to a firing threshold slightly above the maximum conductance of a spike. On the contrary, if at least one measurement input peak coincides with a input time's spike, allowing a small time deviation Δt, then the sum of their conductance curves will exceed the firing threshold and will produce an output peak. The neuron N2 proceeds as for the AND architecture to select relevant information.

For both architecture OR and AND, other neurons can be needed to ensure the selection of relevant information without modifying neuron parameters, especially if neuron N1 has a lot of input data. A complete example is shown in Fig. 12. This rule is made up of two tests. The first one is to check if the positive OR negative values of $P2$ should raise an alert. For this, inputs of the N1 in the OR part are $P2^+$, $P2^-$ and the spike train representing the measurements times. The second one is to check if the positive values of $P1$ AND the results of the first test $T1$ should raise an alert. For this, inputs of the N1 in the AND part are $P1^+$ and the output of the first test $T1$.

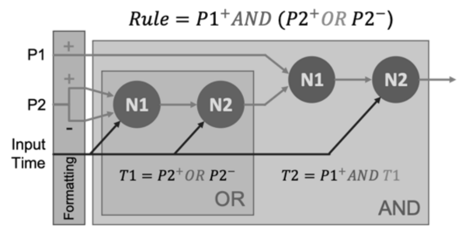

Fig. 12. Example of a rule and its representation with our model.

c) Learning phase

To build our learning model, we wanted to reproduce the information backpropagation operated on ANNs. For this we work on each of the measurements to allow learning according to the result obtained by our SNN model and that expected. To date, having no real measurement, we will carry out our learning phase using the results obtained by the rule-based engine. We have shown that without the learning phase, we were able to have a similarity rate with the rule-based model close to 96%. We seek with the learning phase to improve these results. Our goal is not to get rule-based engine results, competition between models is not necessary. Our model must allow us to adapt to the real values obtained by the patients. A second experiment consists in noising our simulated data to represent data that may be different from the results expected by the rule-based engine and to observe the adaptation of our model. In this case it is possible to show that we are able to build an SNN model on a solid basis - a rule used by doctors - but able to adapt more specifically to a patient

IV. Experiments

Our aim is to compare raised and non-raised alerts for the 9 rules of the rules-based model and their adaptation to the SNN model. As a reminder, to test these rules we use a set of simulated data that represents data possibly collected during a clinical study. This dataset consists of 20,780 measurements (containing all the patient's parameters such as Heart Rate, SPO_2, ...) distributed among 1000 patients. Since this data did not contain the average values for the patients, we computed for each patient the average of each parameter values, except for nausea and vomiting for which the normal value has been set to 0. As explained in Section III-D2, the two parameters τ_{down} and Δt_{max}, corresponding to the rate of restabilization of the conductance peaks and the maximum lag time of input peaks obtained during data formatting, have an influence on the results of the model. We tested the possible values for those parameters in order to identify the combination giving the best results when comparing the rules-based and SNN models. We launched our model on all the values of the dataset for all values of τ_{down} in the interval [14–26] and Δt_{max} in the interval [2 −24] and we evaluated the similarity of the alert raising compared with the rules-based model. Those intervals are determined experimentally, each couple (τ_{down}, Δt_{max}) under or over those intervals gave us too low results. The similarity rate is computed by counting the number of identical alerts raised (or not) by our model compared with the rules-based model, for each couple (τ_{down}, Δt_{max}) The tested values are shown in Fig. 13. The darker a point on this graph is, the more efficient the τ_{down}/ Δt_{max} combination is to obtain results similar to the rule-based model. We had several combinations giving us results greater than 90% similarities. We clearly see a crescent-shaped area with good results. This indicates that our choice may not be the only suggested solution. However, in our problem, we used the one that gave us the best results.

The highest similarity rate between the two models was obtained with the values τ_{down}= 19ms and Δt_{max} = 5ms for the SNN model. With our model and by processing all our dataset, we obtain the confusion matrix in Table II. We add the results of the nine rules and we compare at every measurement the state of our results (True Positive, True Negative, False Positive and False Negative). With this confusion matrix, we can calculate a similarity rate of 96.22%.

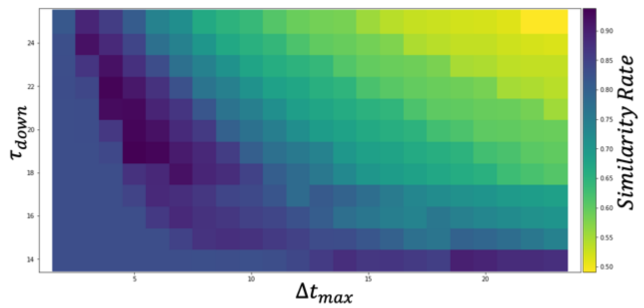

Fig. 13. Results obtained during our experiments.

TABLE II: Confusion Matrix of Our Dataset

		Our Model	
		Alert	No Alert
Rule-based Model	Alert	87953	4762
	No Alert	2143	91973

V. Discussion

The similarity rate of 96.22% between the rules-based and the SNN models shows that it is possible to construct an SNN directly from medical expertise. It is important to note that this similarity rate, obtained by comparing the alerts raised by the two models, does not inform on the actual condition of the patient. We assume that the rule engine produces patient alerts correctly by building it according to the instructions of the medical staff. Reality may be different. A second clinical study is underway to qualify the alerts triggered by the rules engine, which will allow comparing the efficiency of the two models.

Though they are based on the same medical expertise, we can observe differences between the two models. The confusion matrix in Table II shows that the SNN model gives 1,15% false-positives and 2.55% false-negatives compared to the rules-based model. This means that our model tends to not raise alerts for data for which the rules-based model does.

Now we observe the results more closely and give examples of differences between the two models.

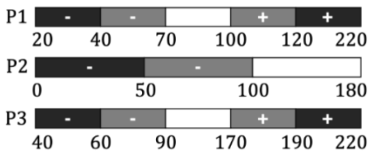

Fig. 14. Areas of three parameters used for our example.

$$Rule = P1^{+} OR\ P1^{-} OR\ P2^{+} OR\ P2^{-} OR\ P3^{+} OR\ P3^{-}$$

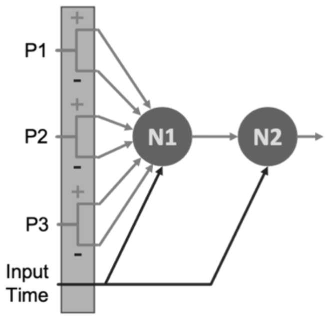

Fig. 15. Example of a rule that can be proposed by the medical team and the resulting model according to our architectures.

Analysis of the results: To illustrate one typical example of false positive, let us consider three parameters P1, P2, and P3 with the areas of severity as in Fig. 14, and consider the rule (P1 + or P1 − or P2 + or P2 − or P3 + or P3 −). Consider now that the values of the parameters for one patient are P1 = 71, P2 = 92 and P3 = 153. The values do not satisfy the rule, so in the rules-based model, no alert is raised. However, the SNN model gives rise to an alert. Indeed, we observe that two of the rules are nearly fulfilled (P1 = 71 ≈ 70 and P2 = 92 ≈ 90). Fig. 15 shows these rules as well as our model obtained.

Thus, in the SNN model, two parameters nearly trespassing their normal values can give raise to an alert, which can make sense from a clinical point of view.

The false-negative cases concern only rules using one patient parameter. Let us consider an example of a typical false negative case, with the rule: (IF ParametreValue ≥ AreaValue THEN raise an alert), where AreaValue denotes the limit value between the white area and the grey area of the parameter. When ParameterValue is slightly greater than AreaValue, the rule-based model detects an alert while ours does not. The sum of the conductance induce by the input time's spike and the value of the parameter observation in our model does not exceed the threshold. This means our formatting gives a delay Δt too important to allows a representative superposition of conductance. This analysis of false positives and false negatives shows that our model is very responsive and requires improvement with additional training. By modifying the weights of the synapses, it is possible to artificially move the thresholds of the zones after formatting the data.

These differences are most certainly because our model does not follow the same scheme for formatting data and defining rules. Indeed, the rules-based engine is defined by strict rules, there are no nuances (i.e. either we go over a limit or not), whereas our model is more nuanced: the alert being raised by observing the crossing of a threshold by the conductance of our neurons, everything depends on how this conductance is constructed. Depending on the arrival time of the different inputs, an alert may not be raised due to a delay (during data formatting) of 1ms.

Advantages of the SNN model: One important point is that the rule-based model doesn't take into account the notion of criticality concerning the recorded data. For example, a heart rate of 110 will be treated as a heart rate of 180 because it is above the threshold value indicated in the rule. With our SNN model, this notion of criticality is taken into account and is represented by the temporal distance between the spike's time and measurement's time.

Moreover, with our SNN model, it is possible to explain the decision making to raise or not the alert since the SNN network topology is built directly from medical expertise. This point is particularly critical in France, indeed, decree number 2017-330 of March 14, 2017 gives any person, that is subject of a decision taken on the basis of algorithmic processing, the right to request the operations carried out by the processing.

Areas for improvement: Remark that we have not used any specific learning method yet. With learning algorithms, we can probably improve our results by personalizing our network to a given patient. Then, as mentioned above, in this work we studied the similarity rate of the two models, but not their actual efficiency for raising alerts. A clinical study is ongoing to compare the decisions of the models with the actual condition of the patients. This 24-month clinical study was planned from November 2020, including 1260 patients spread over 24 hospital centers. At the end of this clinical study, we will therefore be able to know the real efficiency of the rules-based and of the SNN model.

VI. Conclusion

In this article, we have proposed a decision-making model based on Spiking Neural Networks (SNN). Our model offers feedback on the patient's health condition and a raising of the alert if necessary. Our contribution showed that is possible to build an SNN model directly from rules set by medical expertise. To do so, the SNN model reproduced the relationships between them based on a subnetwork assembly method modelling the AND and OR operations. We compared our model to a rules-based model used for a first clinical study on 1000 patients and results show 96% of similar results. The ongoing clinical study will ultimately allow us to decide on the effectiveness of the two models. The SNN thus developed embodies an innate behavior which can then be improved by learning, as STDP or the other efficient unsupervised learning [41], [42]. Hence, our next works will focus on learning, with the objective to personalize the SNN according to the patient.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Sebastien Cohen concucted the research with the help of all others authors. Florian Legendre analyzed the data. All authors wrote the paper. All authors had approved the final version.

Funding

The Smart Angel project is financially supported by the Future Investments Program (PIA) within the framework of the "Structuring Research and Development Projects for Competitiveness" program PSPC 5.

References

[1] 20 statistics that prove connected health is the next big thing in healthcare. [Online] Available: https://bitrebels.com/technology/20-statistics-prove-connected-health-next-big-thing-healthcare/

[2] How is technology impacting home healthcare. [Online] Available: https://appinventiv.com/blog/technology-impact-on-home-healthcare/

[3] Iot healthcare sensor revenues grow globally through patient engagement and accurate data analysis. [Online] Available: https://www.iot-now.com/2019/09/17/98709-iot-healthcare-sensors-revenue-grows-globally-drivenbetter-patient-engagement-increased-accuracy-data-analysis/.

[4] A. Kulshreshtha, J. C. Kvedar, A. Goyal, E. F. Halpern, and A. J. Watson, "Use of remote monitoring to improve outcomes in patients with heart failure: a pilot trial," *International Journal of Telemedicine and Applications*, Article ID 870959, 2010, doi: 10.1155/2010/870959.

[5] D. Dias and J. P. S. Cunha, "Wearable health devices—vital sign monitoring, systems and technologies," *Sensors*, vol. 18, pp. 2414, 2018.

[6] S. Ancoli-Israel, P. Clopton, M. R. Klauber, R. Fell, and W. Mason, "Use of wrist activity for monitoring sleep/wake in demented nursing-home patients," *Sleep*, vol. 20, pp. 24-27, 1997.

[7] J. C. Mentes, B. Wakefield, and K. Culp, "Use of a urine color chart to monitor hydration status in nursing home residents," *Biological Research for Nursing*, vol. 7, pp. 197-203, 2006.

[8] G. Douw *et al.*, "Nurses' worry or concern and early recognition of deteriorating patients on general wards in acute care hospitals: A systematic review," *Critical Care*, vol. 19, pp. 1-11, 2015.

[9] L. P. Malasinghe, N. Ramzan, and K. Dahal, "Remote patient monitoring: a comprehensive study," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 57-76, 2019.

[10] M. Jaensson, K. Dahlberg, and U. Nilsson, "Factors influencing day surgery patients' quality of postoperative recovery and satisfaction with recovery: a narrative review," *Perioperative Medicine*, vol. 8, pp. 1-7, 2019.

[11] S. M. Farhad, M. R. Minar, and S. Majumder, "Measurement of vital signs with non-invasive and wireless sensing technologies and health monitoring," *Journal of Advances in Information Technology*, vol. 8, no. 3, pp. 187-193, 2017.

[12] B. Patel and A. Thind, "Usability of mobile health apps for postoperative care: Systematic review," *JMIR Perioperative Medicine*, vol. 3, no. 2, e19099, Jul. 20, 2020.

[13] J. L. Semple, S. Sharpe, M. L. Murnaghan, J. Theodoropoulos, and K. A. Metcalfe, "Using a mobile app for monitoring post-operative quality of recovery of patients at home: a feasibility study," *JMIR mHealth and uHealth*, vol. 3, no. 1, e18, Feb. 12, 2015.

[14] M. McGillion *et al.*, "Postoperative remote automated monitoring and virtual hospital-to-home care system following cardiac and major vascular surgery: User testing study," *Journal of Medical Internet Research*, vol. 22, no. 3, 2020, doi: 10.2196/15548.

[15] T. Tanantong, E. Nantajeewarawat, and S. Thiemjarus, "False alarm reduction in bsn-based cardiac monitoring using signal quality and activity type information," *Sensors*, vol. 15, pp. 3952-3974, 2015.

[16] I. Bisio, F. Lavagetto, M. Marchese, and A. Sciarrone, "A smartphone-centric platform for remote health monitoring of heart failure," *International Journal of Communication Systems*, vol. 28, pp. 1753-1771, 2015.

[17] F. A. Sonnenberg and J. R. Beck, "Markov models in medical decision making: a practical guide," *Medical Decision Making*, vol. 13, pp. 322-338, 1993.

[18] A. Hussain, W. Rao, X. Zheng, H. Wang, and L. D. S. Aristides, "Personal home healthcare system for the cardiac patient of smart city using fuzzy logic," *Journal of Advances in Information Technology*, vol. 7, pp. 58-64, 2016.

[19] S. S. Khan and J. Hoey, "Review of fall detection techniques: A data availability perspective," *Medical Engineering & Physics*, vol. 39, pp. 12-22, 2017.

[20] A. L. Hodgkin and A. F. Huxley, "Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo," *The Journal of Physiology*, vol. 116, pp. 449-472, 1952.

[21] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, pp. 1659-1671, 1997.

[22] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1569-1572, 2003.

[23] S. R. Kulkarni and B. Rajendran, "Spiking neural networks for handwritten digit recognition—supervised learning and network optimization," *Neural Networks*, vol. 103, pp. 118-127, 2018.

[24] J. J. Hopfield, "Neurodynamics of mental exploration," in *Proc. the National Academy of Sciences*, 2010, pp. 1648-1653.

[25] S. Sharmin, P. Panda, S. S. Sarwar, C. Lee, W. Ponghiran, and K. Roy, "A comprehensive analysis on adversarial robustness of spiking neural networks," in *Proc. International Joint Conference on Neural Networks*, 2019, pp. 1-8, arXiv:1905.02704.

[26] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, pp. 82-99, 2018.

[27] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, pp. 1063-1070, 2004.

[28] A. Biswas, S. Prasad, S. Lashkare, and U. Ganguly, "A simple and efficient SNN and its performance & robustness evaluation method to enable hardware implementation," *ArXiv*, vol. abs/1612.02233, 2016.

[29] S. N. S. Kbah and N. S. Şengör, "Investigating the synchronization of cortical neurons using brian simulator," *IEEE INISTA*, vol. 2013, pp. 1-5, 2013.

[30] M. Hines, A. P. Davison, and E. Muller, "Neuron and python," *Frontiers in Neuroinformatics*, vol. 3, pp. 1, 28 January 2009.

[31] M.-O. Gewaltig and M. Diesmann, "Nest (neural simulation tool)," *Scholarpedia*, vol. 2, pp. 1430, 2007.

[32] D. F. M. Goodman and R. Brette, "The brian simulator," *Frontiers in Neuroscience*, vol. 3, pp. 26, 2009.

[33] A. P. Davison *et al.*, "Pynn: A common interface for neuronal network simulators," *Frontiers in Neuroinformatics*, vol. 2, pp. 11, 2009.

[34] A. Leguilcher, "System, device and method for the personalised monitoring of the health status of at least one at-risk patient," 2017.

[35] T. Chevallier *et al.*, "Feasibility of remote digital monitoring using wireless bluetooth monitors, the smart angel™ app and an original web platform for patients following outpatient surgery: A prospective observational pilot study," *BMC Anesthesiology*, vol. 20, pp. 1-8, 2020.

[36] S. R. Kheradpisheh and T. Masquelier, "Temporal backpropagation for spiking neural networks with one spike per neuron," *International Journal of Neural Systems*, vol. 30, pp. 2050027, 2020.

[37] N. K. Kasabov, *Time-space, Spiking Neural Networks and Brain-inspired Artificial Intelligence*, Springer Publishing Company, 1st edition, 2019.

[38] D. F. M. Goodman and R. Brette, "Brian: A simulator for spiking neural networks in python," *Frontiers in Neuroinformatics*, vol. 2, pp. 5, 2008.

[39] C. Rossant, D. F. M. Goodman, B. Fontaine, J. Platkiewicz, A. K. Magnusson, and R. Brette, "Fitting neuron models to spike trains," *Frontiers in Neuroscience*, vol. 5, pp. 9, 2011.

[40] S. N. S. Kbah, "A computational model of the brain cortex and its synchronization," *BioMed Research International*, vol. 2020, pp. 13, 2020.

[41] M. Meng, X. Yang, S. Xiao, and Z. Yu, "Spiking inception module for multi-layer unsupervised spiking neural networks," *International Joint Conference on Neural Networks*, vol. 2020, pp. 1-8, 2020.

[42] W.-M. Kang *et al.*, "A spiking neural network with a global self-controller for unsupervised learning based on spiketiming-dependent plasticity using flash memory synaptic devices," *International Joint Conference on Neural Networks*, pp. 1-7, 2019.