

# Memristor Models for Early Detection of Sepsis in ICU Patients

Vasileios Athanasiou<sup>1</sup>, Zoran Konkoli<sup>1</sup>

<sup>1</sup> Microtechnology and Nanoscience department, Chalmers University of Technology, Gothenburg, Sweden

## Abstract

*A supervised learning technique is used to carefully train memristor models to predict at an early stage whether a patient in intensive care unit (ICU) has the sepsis. A memristor behaves as a resistor, with a (mem)resistance that changes over time within a bounded interval. The resistance value depends on the full history of an applied voltage difference across the element, in the same way as the state of the brain depends on what a person has experienced in the past. The information contained in a voltage difference time series can be encoded in the resistance value. Clinical variables measured subsequently each hour since the patient's admittance in ICU are transformed into voltage difference signals with transformation functions. The training procedure involves the optimization of the transformation functions. The decision of whether to predict sepsis or not is taken by reading the value of the resistance. The authors have participated in the Physionet 2019 challenge with the name called "the memristive agents" and their best submission resulted to a utility score 0.20 on a hidden test data-set.*

## 1. Introduction

Sepsis is one of the main causes of mortality and morbidity in Intensive Care Unit (ICU). [1] Detecting sepsis at an early stage could result in the avoidance of the negative consequences because antibiotic treatment is more effective at the onset of the disease. [2,3]

Traditional models for the early detection of sepsis have been found to be of low-performance. Instead, there is a need for machine learning models. [4] Such models could be used since they can read a big amount of clinical variable values and identify the risk of sepsis. Machine learning models for early prediction of sepsis have been developed and described in literature for patients in ICU [5] and in neonatal ICU [6]. Such models read clinical variables measured over time to detect the risk of sepsis.

Memristors have been successfully used for dealing with temporal information processing problems.[7, 8] The memristor is one of the simplest electronic components

which accumulate information about their past [9]. It is a non-linear, passive, two-terminal component. It is a resistor with time-varying resistance, the memristance. The memristance's rate of change at a specific time instance depends on the applied voltage difference across the memristor. Therefore, this component has the ability to remember: the memristance value at a time instance depends on the whole history of the applied voltage difference. The memristance is bounded within an interval  $[R_{min}, R_{max}]$ .

This component can be used as an inference unit for predicting sepsis if it is driven by a voltage signal generated through measurements of clinical variables. If the memristance is driven in a region  $R_b < R < R_{max}$  (or  $R_{min} < R < R_b$ ), then, this would indicate sepsis. Depending on how far (or close) the memristance value is from  $R_b$  could be used for estimating the probability of having the sepsis. This paper investigates whether the optimization of clinical variable transformations into voltage signals can contribute to the early accurate prediction of sepsis. Transformations could be trained with supervised learning by using data which consist of measured clinical variables from patients in ICU and a label of diagnosed sepsis for each hour since their admittance in ICU.

## 2. The inference unit

In this article, an inference unit is used as a basic component. The setup of the inference unit is shown in Fig. 1. In this figure, the graphs are hand-drawn examples to illustrate the main ideas. The data consists of 39 measured clinical variables,  $x_j(t)$ , and a label for every hour  $t$  since the patient's admittance in ICU. The label shows sepsis ( $l = 1$ ) six hours before the diagnosis of the disease and each hour afterwards. Data retrieved from 20643 (25000) patients are used as a training (validation) data-set [10].

The voltage signal  $V$  is generated by transforming each clinical variable with a transformation function  $\Phi_j[x_j]$ . Each transformation produces a voltage pulse with a duration  $\tau = t/39$ . If the clinical variable  $x_j(t)$  is not available at time  $t$  ( $NaN$ ), then the transformation function returns zero *i.e* a decision cannot be made whether the memristance value should move towards the sepsis region or not.

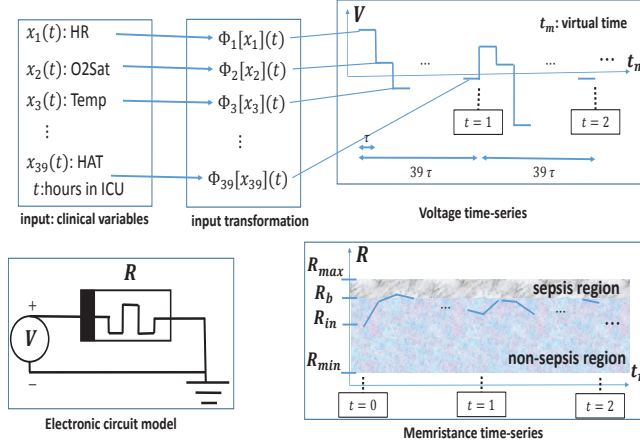


Figure 1. The setup of an inference unit which consists of one memristor element.

The notation  $\Phi$  is used to denote that the value of the transformation at time instance  $t$ ,  $\Phi_j[x_j](t)$ , depends on the values  $x_j(t)$ ,  $x_j(t-1)$ ,  $x_j(t-2)$ ,  $\dots$ ,  $x_j(0)$ . In particular, one example of such transformation is the following:

$$\Phi_j^k[x_j](t) = w_{jb} + w_{j0} x_j(t) + w_{j1} x_j(t-1) + \dots + w_{jk} x_j(t-k) \quad (1)$$

We also use another type of transformation which depends on the memristance difference  $\Delta R(t) = R(t-1) - R(t-2)$ , where  $R(t-1)$ ,  $R(t-2)$  is the memristance value at the end of hour  $t-1$  and  $t-2$  respectively:

$$\Phi_j^\Delta[x_j](t) = w_{jb}^\Delta + w_{j0}^\Delta x_j(t) + w_j^\Delta \Delta R(t) \quad (2)$$

The produced voltage signal  $V(t)$  is applied across a memristor model. The simulation of the memristance results in a memristance time-series  $R(t)$ . A hand-drawn time-series is shown in Fig. 1. The memristance is always initialized at the value  $R_{in}$  at time  $t=0$ . Then, the memristance changes depending on the applied voltage. If the voltage is positive (negative), the memristance increases (decreases). It remains the same if the voltage is zero. For the specific example of Fig. 1, if the memristance is larger than the value  $R_b$ , then the unit predicts sepsis.

Initially, when  $R = R_{in}$  it is assumed that the probability of a patient to have the sepsis is 0 since the model has not been exposed to any input yet. Therefore, the sepsis region cannot involve  $R_{in}$ . Instead, the sepsis region should involve  $R_{max}$  or  $R_{min}$ . If the sepsis region involves  $R_{max}$  ( $R_{min}$ ) then  $R_b > R_{in}$  ( $R_b < R_{in}$ ).

One of the simplest memristor models is used, the Pershin Di Ventra model. [11] The dynamical equation of the memristance change  $dR$  in a time-interval  $\tau$  ( $\dot{R} = dR/\tau$ )

is the following:

$$\begin{aligned} \dot{R}(t) &= \beta V(t) + 0.5 (\alpha - \beta) \\ &[|V(t) + V_{thr}| - |V(t) - V_{thr}|] \\ &\theta[R(t) - R_{min}] \theta[R_{max} - R(t)] \end{aligned} \quad (3)$$

where, the parameters  $\beta$ ,  $\alpha$ ,  $V_{thr}$ ,  $R_{min}$  and  $R_{max}$  are considered fixed in the rest of the paper. The function  $\theta$  is the unit step function,  $\theta(x) = 0$ , if  $x \leq 0$ , otherwise  $\theta(x) = 1$ .

The goal is to train the parameters  $w$  of the transformation functions used and the parameters  $R_b$  of the models involved. To grade the fitness of a choice of such parameters, a utility score  $\nu$  [10] is used. This score is 0.0 if a system predicts always non-sepsis and 1.0 if the system predicts correctly all the cases of sepsis and non-sepsis. It increases if sepsis is correctly predicted up to 9 hours before and three hours after the diagnosis of sepsis. The maximum reward is given for 6 hours before sepsis happened. Otherwise, if sepsis is not correctly predicted (false positive), the utility score decreases. For every true negative, there is neither increase nor decrease in the score. Additionally, every false negative after the diagnosis of the sepsis decreases the score.

### 3. Predicting with one inference unit

Firstly, the training procedure is shown with one-memristor models. For a specific type of transformation function, its' corresponding parameters  $w$  are trained given a training data-set. Since we have not calculated any analytical gradients of the utility score with respect to the parameters  $w$ , it is not possible to use a gradient descent optimization. A searching algorithm is used. The space of the parameters  $w$  is searched to find high utility scores. In the first iteration, the best solution is searched among a broad space of possible solutions. A huge number of random solutions is evaluated by measuring the utility score on the training data-set. Fifty solutions with the highest utility score on the training data-set are also evaluated on the validation data-set. Finally, the solution with the highest sum of utility scores on the training and validation data-sets is kept as the best solution of the first iteration. In the second iteration, the space narrows centered around the best solution of the first iteration. Generally, in each next iteration, the space narrows centered around the best solution of the previous iteration until the space narrows to a local maxima. To increase the chances that a local maxima is also a global maxima, the space should be searched extensively during each iteration.

The maximum utility for every set of parameters  $w$ ,  $W_c$ , is calculated with the following algorithm:

- 1. set the parameters of the transformation function to  $W_c$

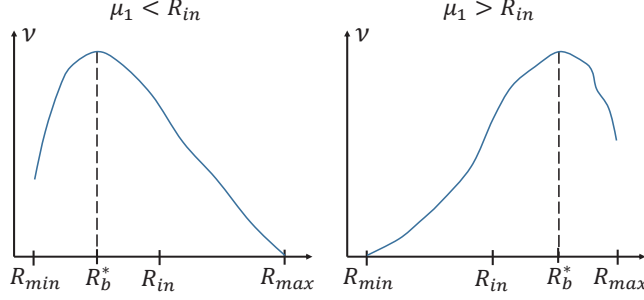


Figure 2. The expected profile of the utility score  $\nu$  as a function of the decision boundary  $R_b^*$  for two different cases.

- 2. simulate the model for every input of the training data-set
- 3. during the simulation of each input, record the mean value of the memristance during each hour
- 4. calculate and record the mean value of the memristance when there was sepsis,  $\mu_1$ , during the simulations of all inputs.
- 5. Use the recorded values to find the value of  $R_b$ ,  $R_b^*$ , for which the utility score is maximum.

The last step of the above algorithm is executed within the optimization of searching the space. It should be relatively fast so as not to slow down the optimization algorithm. For this reason, we use a divide and conquer algorithm. The profile of the utility score is expected to be similar to the one shown in Fig. 2, a concave function of  $R_b$  with one global maxima. Two cases are considered, if  $\mu_1 < R_{in}$  then  $R_{min} < R_b^* < R_{in}$  and if  $\mu_1 > R_{in}$  then  $R_{in} < R_b^* < R_{max}$ . Additionally, the probability for sepsis is calculated for the two different cases as follows. In the first case this probability is calculated as:

$$Pr(R) = \begin{cases} 0 & \text{if } R \geq R_{in} \\ 0.5 \frac{R - R_{in}}{R_b^* - R_{in}} & \text{if } R_{in} > R \geq R_b^* \\ 0.5 \frac{R - R_b^*}{R_{min} - R_b^*} + 0.5 & \text{if } R < R_b^* \end{cases} \quad (4)$$

and, in the second case:

$$Pr(R) = \begin{cases} 0 & \text{if } R \leq R_{in} \\ 0.5 \frac{R - R_{in}}{R_b^* - R_{in}} & \text{if } R_{in} < R \leq R_b^* \\ 0.5 \frac{R - R_b^*}{R_{max} - R_b^*} + 0.5 & \text{if } R > R_b^* \end{cases} \quad (5)$$

If  $Pr(R) > 0.5$  then the system predicts sepsis.

The steps of the algorithm are as follows:

```

1. IF ( $\mu_1 < R_{in}$ )
   $b_{low} = R_{min}$ ,  $b_{high} = R_{in}$ 
ELSE IF
   $b_{low} = R_{in}$ ,  $b_{high} = R_{max}$ 
END IF

```

2. choose two candidate solutions  $R_{c1}$  and  $R_{c2}$  so that  $b_{low} < R_{c1} < R_{c2} < b_{high}$

3. DO\_LOOP

1. measure the utility scores  $\nu(R_{c1})$  and  $\nu(R_{c2})$

2. if ( $\nu(R_{c1}) > \nu(R_{c2})$ ), then,  $b_{high} = R_{c2}$

3. if ( $\nu(R_{c1}) < \nu(R_{c2})$ ), then,  $b_{low} = R_{c1}$

WHILE\_LOOP ( $|\nu(R_{c1}) - \nu(R_{c2})| > \epsilon$ ), for  $\epsilon \rightarrow 0^+$

4.  $R_b^* = R_{c1}$

The system was trained with the transformation function  $\Phi_j^0$ . The utility score on the training and validation data-set is shown in table 1. It was found 0.3140 in the training data-set and 0.2411 in the validation data-set.

#### 4. Predicting with many inference units

Up to here we have considered only one inference unit. However, if there are many inference units  $I_1, I_2, \dots, I_N$  different to each other, then, it would be possible to take advantage of this heterogeneity to improve the prediction accuracy. For example, different units can be used for classifying based on different characteristics of the input. Herein, inference units are used in parallel to improve the prediction accuracy. If each inference unit  $I_i$  is trusted with a probability  $Tr_i$ , where  $\sum_{i=1}^N Tr_i = 1$  and predicts sepsis with a probability  $Pr_i$ , then, the whole system would predict sepsis with a probability:

$$Pr^N(R) = \sum_{i=1}^N Tr_i Pr_i(R) \quad (6)$$

In this work, training such systems is done in two steps. Firstly, the model  $I_1$  is trained as it is explained in section 3, as an one inference unit. Secondly, the model  $I_2$  is trained as an one inference unit, but by providing the probabilities of the model  $I_1$ ,  $Pr_1(R)$ . The probability of sepsis, which is used for training  $I_2$ , is considered as:

$$Pr^2(R) = \frac{0.5}{0.5 + p_2} Pr_1(R) + \frac{p_2}{0.5 + p_2} Pr_2(R) \quad (7)$$

where, the parameter  $p_2$  is optimized additionally to the parameters of the transformation function of the unit  $I_2$ .  $p_2$  is used to quantify the contribution of  $I_2$  to calculating the total probability. The parameter  $p_2$  is optimized after optimizing  $R_b$  with a similar algorithm. It is assumed that the utility score is a concave function of  $p_2$  with one global maximum.

Similarly, the probabilities  $Pr_1(R)$  and  $Pr_2(R)$  could be used for training the model  $I_3$  by optimizing an additional parameter  $p_3$ . So on, the models  $I_4, \dots, I_N$  could be trained in the same way.

A system was trained when the model  $I_1$  consisted of the transformation function  $\Phi_j^0$  and the model  $I_2$  consisted of the transformation function  $\Phi_j^1$ . The utility scores are

Table 1. The maximum utility scores  $\nu$  resulted from the training processes.

	$\Phi_j^0$	$\Phi_j^0 \Phi_j^1$	$\Phi_j^0 \Phi_j^1 \Phi_j^\Delta$
training	0.3140	0.3223	0.3241
validation	0.2411	0.2487	0.2501

shown in the table 1 under the column  $\Phi_j^0 \Phi_j^1$ . Additionally, a third model  $I_3$  was trained with transformation function  $\Phi_j^\Delta$ . The utility scores of the three models combined are shown in the same table under the column  $\Phi_j^0 \Phi_j^1 \Phi_j^\Delta$ . All the trained models of the different combinations of transformation functions were submitted for the physionet 2019 challenge. The best score on a hidden test data-set was received as 0.20.

The utility scores, which are summarized in table 1 show that by additionally training a second and third model improved the utility scores on both training and validation data-sets. The question here is if training additional models always improves the utility scores. For example, one more particular question is if the utility scores would increase given that all the models  $I_3, I_4, \dots, I_N$  consisted of the same transformation function  $\Phi_s$ . Moreover, which transformation function  $\Phi_s$  would be suitable? Another question is if the usage of more complex transformation functions, such as neural networks, could improve the utility scores. All these questions are to be answered in future work.

## 5. Conclusion

This work introduces a new method for training memristor models to predict at an early stage whether an ICU patient has the sepsis. The results show that simple models with a small number of free parameters (79) can perform with utility scores in the range 0.2411–0.3140. A way was found to improve the utility scores by training additional models. Further work is needed to improve the utility scores. In particular, one could use more complex models as transformation functions, *e.g.* feed-forward neural networks. Additionally, one could also use models of memristor networks as echo-state networks for reservoir computing instead of a single memristor element. All those developments are left for future work.

## References

- [1] Singer M, Deutschman CS, Seymour CW, Shankar-Hari M, Annane D, Bauer M, Bellomo R, Bernard GR, Chiche JD, Coopersmith CM, Hotchkiss RS, Levy MM, Marshall JC, Martin GS, Opal SM, Rubenfeld GD, van der Poll T, Vincent JL, Angus DC. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA* 2016; 315(8):801–810. ISSN 1538-3598 0098-7484.
- [2] Seymour CW, Gesten F, Prescott HC, Friedrich ME, Iwashyna TJ, Phillips GS, Lemeshow S, Osborn T, Terry KM, Levy MM. Time to treatment and mortality during mandated emergency care for sepsis. *The New England journal of medicine* 2017;376(23):2235–2244. ISSN 1533-4406 0028-4793.
- [3] Kumar A, Roberts D, Wood KE, Light B, Parrillo JE, Sharma S, Suppes R, Feinstein D, Zanotti S, Taiberg L, Gurka D, Kumar A, Cheang M. Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock\*. *Critical Care Medicine* 2006;34(6):1589–1596. ISSN 0090-3493.
- [4] Islam MM, Nasrin T, Walther BA, Wu CC, Yang HC, Li YC. Prediction of sepsis patients using machine learning approach: A meta-analysis. *Computer Methods and Programs in Biomedicine* 2019;170:1–9. ISSN 0169-2607.
- [5] Nemati S, Holder A, Razmi F, Stanley MD, Clifford GD, Buchman TG. An interpretable machine learning model for accurate prediction of sepsis in the icu. *Critical Care Medicine* 2018;46(4):547–553. ISSN 0090-3493.
- [6] Masino AJ, Harris MC, Forsyth D, Ostapenko S, Srinivasan L, Bonafide CP, Balamuth F, Schmatz M, Grundmeier RW. Machine learning models for early sepsis recognition in the neonatal intensive care unit using readily available electronic health record data. *PloS one* 2019;14(2):e0212665–e0212665. ISSN 1932-6203.
- [7] Athanasiou V, Konkoli Z. On using reservoir computing for sensing applications: exploring environment-sensitive memristor networks. *International Journal of Parallel Emergent and Distributed Systems* 2017;.
- [8] Du C, Cai F, Zidan MA, Ma W, Lee SH, Lu WD. Reservoir computing using dynamic memristors for temporal information processing. *Nature Communications* 2017; 8(1):2204. ISSN 2041-1723.
- [9] Chua L. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory Sep.* 1971;18(5):507–519. ISSN 0018-9324.
- [10] Reyna M, Josef C, Jeter R, Shashikumar S, Westover M, Nemati S, Clifford G, Sharma A. Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. *Critical Care Medicine* 2019;In press.
- [11] Pershin YV, Di Ventra M. Experimental demonstration of associative memory with memristive neural networks. *Neural Networks* 2010;23(7):881–886.

Address for correspondence:

Vasileios Athanasiou  
Electronics Materials and Systems Laboratory  
Kemivgen 9, MC2, Chalmers University of Technology, Gothenburg, Sweden  
vasath@chalmers.se