On the Performance of Hierarchical Temporal Memory Predictions of Medical Streams in Real Time

Noha O. El-Ganainy[1], Ilangko Balasingham[1,2], Per Steinar Halvorsen[2], Leiv Arne Rosseland[3,4].

[1] *Departement of Electronic Systems, Norwegian University for Science and Technology (NTNU), Trondheim, Norway.*

[2]*The Intervention Center, Division of Emergencies and Special care, Oslo university hospital, Oslo, Norway.*

[3]*Departement of Research and Development, Division of Emergencies and Special Care, Oslo university hospital, Oslo, Norway.*

[4]*Institute of Clinical Medicine, Faculty of Medicine, University of Oslo, Oslo, Norway.*

***Abstract*— Machine learning is widely used on stored data, recently it is developed to model real time streams. Applying machine learning on medical streams might lead to a breakthrough on emergency and critical care through online predictions. Modeling real time streams implies limitations to the current state-of-the-art of machine learning and requires different learning paradigm. In this paper, we investigate and evaluate two different machine learning paradigms for real time predictions of medical streams. Both the hierarchical temporal memory (HTM) and long short-term memory (LSTM) are employed. The performance assessment using both algorithms is provided in terms of the root mean square error (RMS) and mean absolute percentage error (MAPE). HTM is found advantageous as it provides efficient unsupervised predictions compared to the semi-supervised learning supported by LSTM in terms of the error measures.**

Keywords—Online Learning, long short-term memory LSTM, hierarchical temporal memory HTM, medical streams.

I. INTRODUCTION

The rise of Internt of things (IoT) along with the wide use of sensors in different real world applications has significantly incresed the availability of real time streams. This has urged research activities to emerge a powerful signal modeling, processing, and prediction tool to help on many aspects. Financial, geological, and climate related applications were the very first applications to request event-detection and decision-making tools in real time [1-2]. The medical sector is widely using sensors with prestigious capabilites of monitoring, measuring, and storing data. Either in an emergency room, intensive care, or surgery room a large amount of sensors are utilized and recent research adressed the need of online predictions and decisions in real time.

There is a wide range of classical modeling and forecasting algorithms among which the most efficient are autoregressive integrated moving average (ARIMA), Hidden Markov Models (HMM), and Holt-Winters. Yet these methods encounter sensitivity to outliers and lack generality as they come with strong assumptions regarding the type of the time-series, ergodic, the noise model, the dependencies between successive bits [2-3]. These factors have urged the need of an efficient processing and modeling tool. Machine learning is a broad class of methods capable to learn, update and accumulate knowledge. It also has the power in specific learning conditions to help reflect understanding insights, and dependencies within the data in a specific framework [1-3]. Yet the rise of streaming applications has motivated the development of machine learning algorithms to model and predict data in real time [2-3].

This paper evaluates and compares the predictions of real time medical streams using two different learning paradigms namely the hierarchical temporal memory (HTM) and the long short-term memory (LSTM). HTM is a recent machine learning paradigm having the power to learn and predict in real time using high order predictions and Hebbian-like learning method that mimics the brain [1-3, 7-9]. While LSTM is a well-established regressive neural network method (RNN) that was originally designed to work on benchmark problems and lately developed to work on real time streams [4-6]. The predictions using both paradigms are observed, compared, and

This work was carried out during the tenure of an ERCIM 'Alain Bensoussan' Fellowship Program.

evaluated in terms of the root mean square error (RMS) and the mean absolute percentage error (MAPE). The practicality of both paradigms to online stream predictions is also discussed. The paper is organized as follows; Section II provides an overview on learning in real time. Section III describes and explains online learning. Section IV presents the methodology and Section V discusses the results. Finally, Section VI presents the paper conclusions.

II. Learning in Real time: Overview

Machine leaning is a powerful modeling and learning tool widely employed by various applications to work on benchmark applications. The data set is entierly available for training, testing, human intervention, and labeling [1]. In contrast to benchmark problems, employing machine learning for streaming applications impose constraints and limitations to the learning framework [2-3].

Recently many machine learning paradigms, both statistical and neural network models, have applied modifications to adapt to the limitation of stream learning [3]. One of the well-established and efficient neural network algorithms is LSTM [4-6]. It is one of the most commonly used RNN models that provides the privilege to maintain the output/input of certain time steps in the past. It has proved to be a powerful modeling tool, is widely employed by many applications, and was recently modified in order to adapt to sequence learning in real time [5- 6].

LSTM-Batch, applied in this work, is a type of LSTM algorithms developed for stream learning. It splits the entire data set into batches, each batch processing represents a phase. In each phase, the batch is fed into the model, the error is calculated and the network's weights are updated relatively to the error. At regular intervals, the data is buffered to retrain the model using the new batch offline. Next, the model is updated online and used for predictions. The scope of this paper is not to find the best version of LSTM for medical streams rather than finding an LSTM benchmark and compare its predictive capabilities to HTM.

In this paper, LSTM-Batch will be considered and applied to the medical stream for modeling and predictions in real time. The performance of LSTM-Batch is evaluated for real time predictions against the online learning paradigm for comparison. Online learning will be briefly explained in the coming section.

III. Online Learning

Online learning is a machine learning paradigm designed to learn from data in real time through the use of HTM algorithm [2-3, 7-8]. The paradigm is capable to learn the temporal patterns within the data and adapt to the changed characteristic of the data in real time. HTM is the heart of the online machine learning paradigm. It is a theoretical algorithm that mimics sequence learning in the cortex. The network consists of a layer of neurons organized as a set of columns. A neuron has three states: active, predictive, and non-active. It also has two different dendritic zones; proximal and distal. The proximal dendrites zone represents the current feed forward input while the distal synapses indicates the temporal context learned at a specific time.

The HTM network learning and activation rules can be explained using four matrixes at time step t for an HTM layer of size $N*M$ where N is the number of columns and M is the number of cells/neurons per column [7-8]. The first matrix is the feed forward input matrix W^t. It represents the set of active columns representing the input. Second is the predictive state matrix Π^t; each element π_{ij}^t is the predictive state of the i^{th} cell in the j^{th} column for the coming time interval.

$$\Pi_{ij}^t = \begin{cases} 1 & if \exists_d \left\| \widetilde{D}_{ij}^d \circ A^t \right\|_1 > \theta \\ 0 & otherwise \end{cases}, \quad (1)$$

where θ is the segment activation threshold, ° is element-wise multiplication, and $\widetilde{D}_{ij}^d$ denotes the connected synapses of segment d of the i^{th} cell in the j^{th} column. Third is the Activation matrix A^t, where a_{ij}^t is the activation state of the i^{th} cell in the j^{th} column.

$$a_{ij}^t = \begin{cases} 1 & if\ j \epsilon W^t\ and\ \pi_{ij}^{t-1} = 1 \\ 1 & if\ j \epsilon W^t\ \ and\ \ \sum_i \pi_{ij}^{t-1} = 0 \\ 0 & otherwise \end{cases}, \quad (2)$$

Equation (2) sums the activation rule as follows; an active cell has to be in an active column. It will be activated regardless of its state in the previous time step; predictive or inactive state.

Otherwise the cell will not be activated. The fourth matrix is D_{ij}^{d}, it represents the permanence matrix of segment d of the i^{th} cell in the j^{th} column. The matrix is updated according to a Hebbian-Style learning; dendritic segments leading to correct predictions are reinforced by increasing its synaptic permanence values by a small amount p^{+} and decreasing all the other synaptic permanence by p^{-}. In case that no cell in an active column was predicted then the cell with the most activated segment is reinforced.

$$\Delta D_{ij}^{d} = p^{+}\dot{D}_{ij}^{d} \circ A^{t-1} - p^{-}\dot{D}_{ij}^{d} \circ (1 - A^{t-1}) \quad , \qquad (3)$$

where $\widetilde{D}_{ij}^{d}$ denotes the connected synapses and $\dot{D}_{ij}^{d}$ is a binary matrix containing only the positive entries in D_{ij}^{d}.

On the other hand, if the prediction at time *t-1* was found wrong at time *t*, i.e. the predicted cell didn't receive enough input, the synaptic permanence that caused the prediction is decreased by a smaller amount p^{--} which is known as long term depression.

$$\Delta D_{ij}^{d} = p^{--}\dot{D}_{ij}^{d} \quad , \qquad (4)$$

where $a_{ij}^{t} = 0$ and $\left\| \widetilde{D}_{ij}^{d} \circ A^{t-1} \right\|_{1} > \theta$, $p^{--} \ll p^{-}$. At time step *t*, the input feed forward input matrix W^{t} changes and the A^{t} is updated as shown in (2). Next, the permanence matrix D_{ij}^{d} is changed and the permanence modified according to the success of the prediction at time *t* as formulated in (3) and (4). The new predictive state matrix Π^{t+1} is consequently modified through (1). The matrixes are now updated and ready for the new input at next time interval.

IV. METHDOLOGY

The aim of this paper is to provide an evaluation of both the LSTM-Batch and HTM as learning and predictive tools for medical streams. Both models are implemented and fed with the data then a thorough performance evaluation of the prediction in the two cases is provided in terms of the RMS MAPE error.

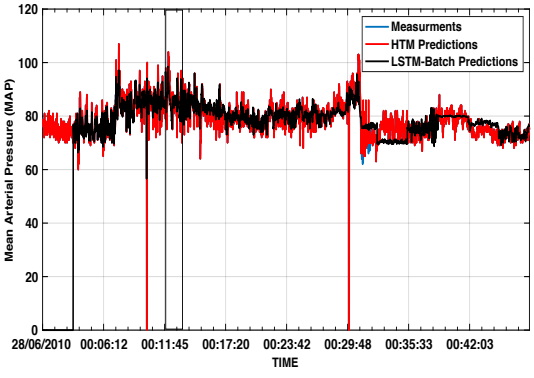
The data set was collected after an observational study at Oslo University Hospital on healthy pregnant women for planned cesarean delivery [10]. The measurements of calibrated invasive systolic arterial pressure (SAP) and estimated cardiac output were continuously monitored for a group of 76 women. The monitoring timeline was 3 minutes on the left lateral position then 3 minutes on the supine position followed by an observation after the spinal anesthesia and until delivery. The data is available as a time series of 3000 to 6000 samples for each patient. Each sample represents the average of the measured parameter over one heartbeat. The data set encloses large variations within the patients' data and there are different temporal patterns between users which provides rich resources to challenge the prediction algorithm and provide a solid comparative benchmark. Our simulations treat the data as a simple stream and aims to provide real time predictions for each time step and overlooks any analysis targeting decision/classification related to the patient-position.

HTM was applied using NuPIC implementation [11]. The model was fed the data in real time without the need to training or modeling. The resulting predictions are compared to the input stream to observe the algorithm's ability to learn the temporal patterns within the data in real time and to evaluate the prediction error.

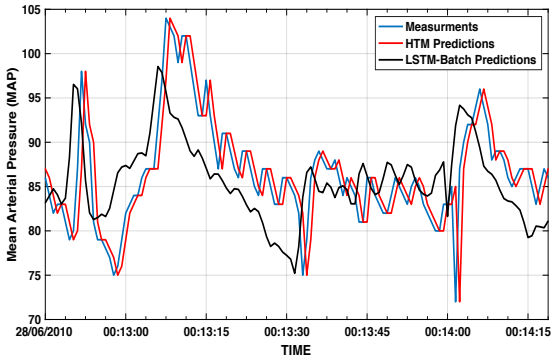
The paper applies the traditional LSTM-Batch method; the data set is divided into several batches each of 250 values. Each batch was split into 50% training and test set. The training set is fed into the model for a max number of 100 epochs, the error is calculated and back propagated to update the weights. The network used an initial learn rate to 0.005 and drop the learn rate after of 50 epochs by using a learn drop factor of 0.2 with the gradient threshold was set to 1. The LSTM model is implemented using MATLAB to predict 2 parameters separately for each patient; the mean arterial pressure (MAP) and heart rate (HR) respectively. The network pursuit the minimization of the RMS and loss through the use of Adam optimizer along with 200 hidden layers.

V. RESULTS

HR and MAP values are fed to HTM separately in real time and the predictions are observed. HTM leads to predictions in real time; for each input value at time t the output represents the algorithm's prediction for the coming input at time *t+1*. As for LSTM-Batch, no prediction is available for the first 250 samples as they are utilized to train the model. After the model is ready, after 250 samples, the stream is fed to it for online predictions; for each

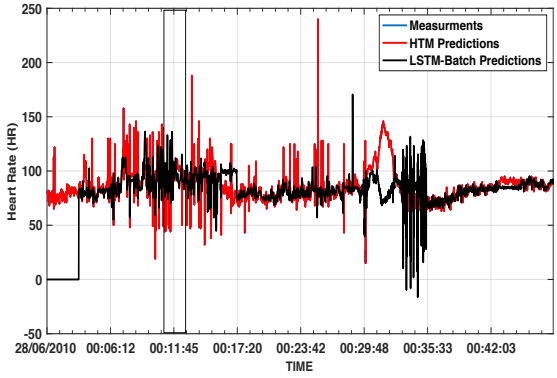


(a)

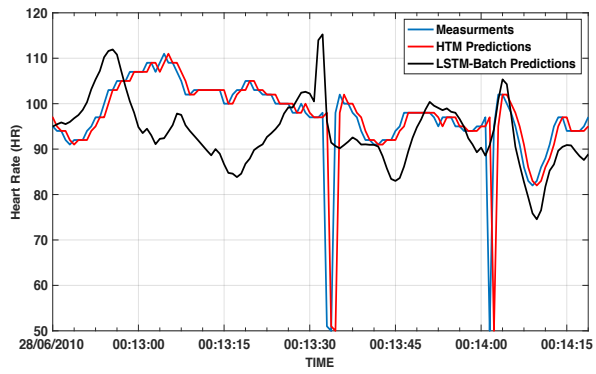


(b)

Figure1. MAP measurements and predictions using both LSTM-Batch and HTM in (a). A window of 125 samples (marked in grey) is shown in (b). The MAP values are displayed with respect to the real time observation duration performed on the 28/06/2010.



(a)



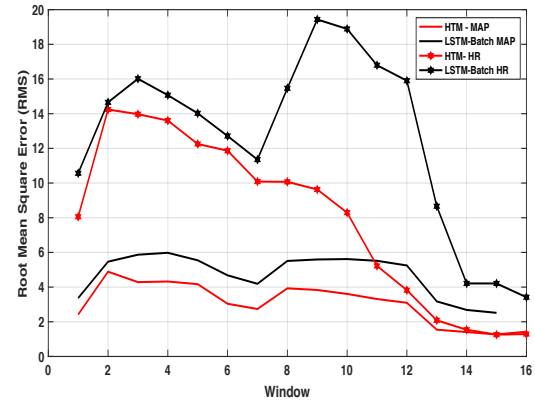
(b)

Figure2. HR measurements and predictions using both LSTM-Batch and HTM in (a). A window of 125 samples (marked in grey) is shown in (b). The HR values are displayed with respect to the real time observation duration performed on the 28/06/2010.

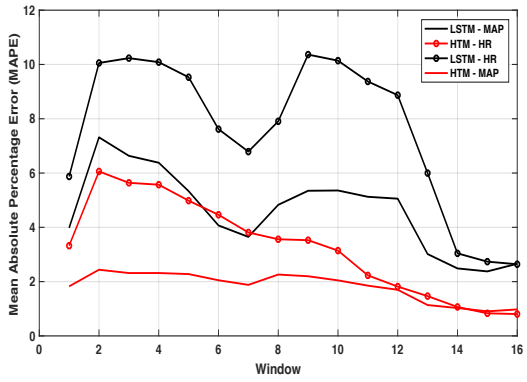
input value at time *t-1* the output represents the prediction for the coming input at time *t*.

LSTM-Batch is retrained offline at regular intervals of 250 input samples to adapt with the variations in the data characteristics while the predictions continue online using the last trained model. The predictions of both HTM and LSTM-Batch are displayed and compared to the original stream for a randomly chosen patient. MAP stream and both predictions are displayed in Fig.1 and the HR stream and predictions are displayed in Fig. 2. A window of 125 samples is carefully investigated to provide detailed observation of the predictions compared to the input stream as displayed in Fig.1.b and Fig. 2.b. It can be clearly stated that for MAP values, the predictions using HTM were able to comply with the temporal pattern and capture even the sudden fluctuations in the data and was shown synchronous with a marginal error. On the other hand, LSTM-Batch was partially able to conform with the temporal pattern with a significant time delay. As for the HR predictions, HTM is still capable to capture the pattern and provide visually similar predictions with minor delay. While LSTM-Batch has obviously lost the temporal pattern and predicted significantly different values compared to the input stream. A fair comparison between HTM and LSTM-Batch should also consider the offline training, buffering, and continuous remodeling done for LSTM compared to HTM which provides online predictions without any offline modeling. There are 2 sources of delays using LSTM-Batch; at the start because of the model training time and the delay resulting from the algorithm adaptivity to the input temporal pattern. It should also be argued that this prediction delays might not be acceptable for applications requiring decisions in real time based on the provided predictions like medical applications.

An evaluation between HTM and LSTM-Batch is performed in terms of the RMS and MAPE. Both parameters are calculated for both the MAP and HR values over a window of 250 samples. The first 250 samples were excluded as they were utilized to train the LSTM-Batch model. The resulting RMS and MAPE are smoothed using the global average



(a)



(b)

Figure 3. The smoothed RMS in (a) and the smoothed MAPE the in (b). The results using HTM are displayed in red and those of LSTM-Batch are shown in black. The RMS and MAPE are computed over a window of 250 samples of the considered stream

for each patient. The RMS error is compared for both LSTM-Batch and HTM in Fig. 3.a. For MAP values. HTM provided lower RMS values in the range of 1 to 5 compared to LSTM-Batch providing a range 3 to 6. As for HR values, HTM provided an RMS ranging from 4 to 14 and LSTM-Batch provided a range from 4 to 19. MAPE values for both HR and MAP is shown in Fig. 3.b. It is in the range of 1-3% using HTM for MAP values and in the range of 3-7% using LSTM-Batch. While for HR values, HTM leads to an error range of 1-6% and LSTM-Batch results in a percentage of 3-11%. Both RMS and MAPE error are shown to be lower using HTM compared to LSTM-Batch for the MAP and HR value. Finally, Table.1 sums up the performance over all patients by illustrating the average RMS and MAPE for HR and MAP using both algorithms. MAPE is in an average range of 15.99% using LSTM-Batch while dropped to

Table1. Average RMS and MAPE of all 76 patients for both MAP and HR using HTM and LSTM-Batch.

Parameter	LSTM-Batch		HTM	
	MAPE %	RMS	MAPE %	RMS
HR	**15.59**	**17.17**	**4.88**	**11.18**
MAP	**12.23**	**19.68**	**3.15**	**8.59**

4.88% using HTM to predict HR values. RMS went from 17.17 to 11.18 when using HTM. On the other hand, for MAP values, RMS decreased from 19.68 to 8.59 using HTM and MAPE has also decreased from 12.23% to 3.15%. Both HTM performance figures are shown to be significantly lower compared to LSTM-Batch.

VI. CONCLUSIONS

This paper compared and evaluated the real time predictions of both hierarchical temporal memory (HTM) and long short-term memory (LSTM) for medical streams. LSTM-Batch was applied in the analysis; the model is retrained over regular time intervals to ensure that it is adapted to the variations on the stream characteristics. A data set collected from an observational study on 76 patients was utilized and both the mean arterial pressure (MAP) and heart rate (HR) parameters are analyzed. The stream length varied from patient to another and the data was rich with various temporal patterns providing a good challenge to the predictive algorithms. HTM was shown to provide efficient predictions and marginal error measures compared to LSTM-Batch in terms of the root mean square error (RMS) and mean absolute percentage error (MAPE) for both MAP and HR. HTM conducted a completely unsupervised prediction without the need for training or prior knowledge of the stream and was capable to learn the temporal pattern within the data and to adapt to the variations. On the other hand, LSTM-Batch needed offline training for a fixed window of the input stream at the start, buffering of the input stream, and model retraining during regular intervals which is considered a semi-supervised process. LSTM-Batch was only partially able to learn the temporal pattern in the data and significant delays were observed as the predictions were not synchronized with the input

stream and failed to adapt to variations in the temporal pattern.

REFERENCES

[1] A. L'Heureux, K. Grolinger, H. El-Yamany, and M. Capretz, "Machine learning with big data: challenges and approaches," IEEE Access, vol.5, pp. 7776-7797, April 2017. Doi: 10.1109/ACCESS.2017.2696365.

[2] S. Ahmed, A. Lavin, S. Purdy, and Z. Agha, Neurocomputing, "Unsupervised Real time Anomaly Detection for Streaming Data," Vo.262, pp. 134-147, November 2017. Doi: 10.1016/j.neucom.2017.04.070

[3] Y. Cui, S. Ahmad, and J. Hawkins, "Continuous online sequence learning with an unsupervised neural network model," Neural Computation, Vol.28, Issue.11, pp. 2474-2504, November 2016. Doi: 10.1162/NECO_a_00893

[4] A. Pulver and S. Lyu, " LSTM with working memory," IEEE joint conference on neural networks IJCNN2017.

[5] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Stwunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," IEEE Trans on neural networks and learning systems, vol. 28, no. 10, pp. 2222-2232. October 2017.

[6] J. Mackenzie, J. F. Roddick, and R. Zito, "An Evaluation of HTM and LSTM for short-term arterial traffic flow prediction, " IEEE Transactions on Intelligent Transportation Systems, pp. 1-11, August 2018. Doi: 10.1109/TITS.2018.2843349.

[7] J. Hawkins and S. Ahmad, ''Why neurons have thousands of synapses, a theory of sequence memory in neocortex,'' *Frontiers Neural Circuits*, vol. 10, no. 23, pp. 1–13, March 2016. Doi: 10.3389/fncir.2016.00023.

[8] S. Ahmad and J. Hawkins, ''Properties of sparse distributed representations and their application to hierarchical temporal memory.'' (2015). [Online]. http://arxiv.org/abs/1503.07469

[9] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, " A distributed anomaly Detection System for In-Vehicle Network Using HTM," IEEE Access, Vol.6, pp. 9091-9098, March 2018. Doi: 10.1109/ACCESS.2018.2799210.

[10] M. Erango, A. Frigessi. L. A. Rosseland, "A three minutes supine position test reveals higher risks of spinal anesthesia induced hypotension during cesarean delivery. An observational study," F1000 Research. Ver. 1. July 2018. doi: 10.12688/f1000research.15142.1

[11] M. Taylor *et al.*, (Feb. 2016). *NuPIC: 0.5.0*. [Online]. Available: https://doi.org/10.5281/zenodo.46074