

A Performance Comparison of LSTM and Recursive SID Methods in Thermal Modeling of Implantable Medical Devices

Ayca Ermis
School of ECE
Georgia Tech
Atlanta, GA, USA
aycaermis@gatech.edu

Mi Zhou
School of ECE
Georgia Tech
Atlanta, GA, USA
mizhou2018@gatech.edu

Yen-Pang Lai
School of ECE
Georgia Tech
Atlanta, GA, USA
yenpang_lai@gatech.edu

Ying Zhang
School of ECE
Georgia Tech
Atlanta, GA, USA
yzhang@gatech.edu

Abstract—This paper investigates application of long short-term memory (LSTM) and recursive system identification (RSID) algorithms to predict the thermal dynamics of bio-implants, e.g. UEA under certain assumptions. Both algorithms implemented in this paper predict the temperature readings of heat sensors using a window size of 10 data points. Simulations in COMSOL software as well as experiments using an *in vitro* experimental systems are utilized for validation and comparison of algorithm performances. Mean squared error (MSE) of prediction results based on the LSTM algorithm is compared against that of the competitive RSID algorithm for evaluation. Both simulation and experimental results indicate that the LSTM can accurately predict the thermal dynamics of the system and outperforms the RSID algorithm when certain conditions for inputs hold. According to COMSOL simulations and *in vitro* experiments, the LSTM algorithm returns more reliable predictions for the time period in which the convergence of the adaptive filters in the RSID algorithm is not yet achieved. Alternatively, once the adaptive filters converge, the performance of the RSID algorithm is significantly better than the LSTM for some cases due to its adaptive learning capabilities.

Index Terms—long short-term memory (LSTM), recursive system identification (RSID) methods, implantable medical device (IMD), thermal effect

I. INTRODUCTION

With recent technological advances in capabilities and applications of implantable medical devices (IMDs), thermal management of these devices has become more significant. Such improvements include increased functionality in monitoring, and providing stimulation when required for medical purposes. In certain applications of IMDs, overheating of the electrodes in the IMD may lead to permanent damage in the surrounding soft tissue [1]. Thus, low power consumption is of importance to maintain the long-term operation of the device and to protect patient's health. Potential causes of permanent damage to soft tissue include tissue injury due to electrical over-stimulation and administration of high electrical currents [2], [3]. For example in [4], authors report an incident in which a patient with an implanted deep brain stimulator (DBS) suffered a serious brain damage due to

overheating in brain tissue surrounding the DBS in response to diathermy.

Thermal effect of IMDs has been studied in previous works [5]- [9]. In [5], authors propose the Pennes bioheat transfer equation to model this thermal effect of the IMDs. In [6], finite element analysis (FEA) and finite difference time domain (FDTD) are proposed to solve the Pennes bioheat equation, which is used to model the heat dissipation in IMDs. However the issue with both methods in [6] is that they solve for the heat dissipation and electromagnetic field for the whole computational domain for each time instance, which makes these methods not suitable for real-time applications due to space and time complexity. In [7] and [8], Chai *et al.* proposes a recursive multi-step prediction error minimization method (RMSPPEM) to model implantable devices with a single heat source and achieve the real-time thermal management. In [9], authors implement an algorithm using recursive subspace identification (RSID) methods to investigate the performance of RSID methods in predicting temperature readings of an IMD with multiple heat sources.

Artificial neural networks (ANNs) have been utilized in many applications of system modeling and control. In [10], the authors develop an ANN-based control scheme for real-time control of hexacopter trajectories. Similarly in [11], a multi-layer perceptron (MLP) is used to predict the aerodynamic flow evolution induced by MEMS. According to the results presented in [11], the use of ANNs, more specifically MLP, is promising even with slightly high time complexity for real-time application. LSTM algorithm was first introduced by [12] and has been recently used in many applications of system identification. A survey of benchmark neural network algorithms used for system identification has been presented in [13], which includes the LSTM algorithm. In [14], a new adaptive learning method is proposed for convex-based LSTM algorithm to achieve online control of dynamic systems. Sadiqbacha *et al.* proposes a thermal modeling framework using LSTM networks to detect major hot spots on a commercial multi-core microprocessor with

an infrared thermal imaging in [15]. To the authors' knowledge, there have been few studies addressing the online thermal prediction problem of IMDs utilizing the LSTM algorithm. In this paper, we focus on applying LSTM algorithm with a window size of 10 past data points to predict temperature readings at each time step and compare the results with alternative system identification methods. Additionally, certain conditions under which LSTM algorithm outperforms alternative identification methods are investigated.

II. SYSTEM DESCRIPTION

An implantable neural prosthetics based on Utah Electrode Array (UEA) is chosen to evaluate the performance of the algorithms, since it has become a benchmark in research and applications [16]. The UEA consists of multiple modules, such as radio, power and motherboard modules. There exists multiple heat sources as well as multiple spatially distributed heat sensors on board. Inputs of the system are the power inputs to the heat sources and temperature readings at sensor locations are used as system outputs, i.e. multiple-input multiple-output system.

The main assumption of this paper is that dynamics of the system do not change significantly; thus, no adaptive learning is implemented with the LSTM algorithm. Performance of the LSTM algorithm is then compared to recursive identification methods with adaptive learning to investigate advantages of the LSTM over adaptive identification methods under certain conditions. Authors also hypothesize in this paper that training the LSTM network with randomly generated Gaussian inputs, the algorithm will achieve successful prediction of temperature readings given any type of inputs with the condition that our main assumption holds.

III. LONG SHORT-TERM MEMORY (LSTM) ALGORITHM

A. Background

A general LSTM cell with memory function consists of three gated units: a forget gate, an update gate, and an output gate, and solves the vanishing gradient problem. In Figure 1, a single cell of the LSTM network is illustrated, where x_k is the input and h_k denotes the hidden layer state. σ and $\tanh$ are sigmoid and hyperbolic tangent functions, respectively. The LSTM network can be formulated as follows

$$\begin{aligned} f_k &= \sigma(W_f \cdot [h_{k-1}, x_k] + b_f) \\ i_k &= \sigma(W_i \cdot h_{k-1} + b_i) \\ g_k &= i_k \cdot \tanh(W_c \cdot [h_{k-1}, x_k] + b_c) \\ c_k &= f_k * c_{k-1} + g_k \\ h_k &= \sigma(W_o \cdot [h_{k-1}, x_k] + b_o) * \tanh(c_k) \end{aligned} \quad (1)$$

where f_k , g_k denote the forget gate and update gate respectively, i_k determines the hidden state value to be updated, and c_k represents the updated cell state. W_f , W_e , W_i , W_c , and W_o denote trainable weights for each layer and b_f , b_c , b_i , and b_o are bias corresponding to each layer.

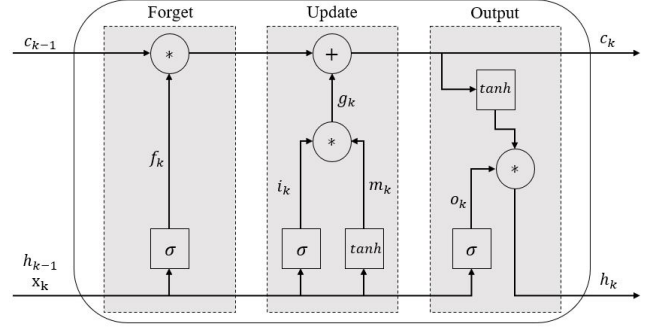


Fig. 1: An illustration of a single LSTM cell [17].

One step-ahead ARX prediction model of the thermal effect of the IMDs can be written as follows using the LSTM network:

$$T(k) = f(T(k-1), T(k-2), \dots, T(k-w); u(k-1), u(k-2), \dots, u(k-w)) \quad (2)$$

where $T(k)$ is the temperature at time step k , w is the window size, $u(k)$ is the input signal at time step k and $f(\cdot)$ is the nonlinear function that will be learned by the LSTM based on the *Universal Approximation Theorem* [18].

B. Architecture of the LSTM Network

One of the main tasks in designing a neural network is to determine how many layers are needed based on the training data. Due to the simplicity of the thermal model, our LSTM network consists of two layers; an LSTM layer and a dense layer with six neurons as shown in Figure 2. Another essential task is to optimize the hyper-parameters, which are dropout rate, batch size and number of neurons in an LSTM layer for our case. In order to optimize the hyper-parameters, a grid search with 10-fold cross validation is conducted to determine the best hyper-parameter combination. For each group of hyper-parameters, the program is run five times and the average MSE is used to be compared to decide the hyper-parameters. As a result, a network with 200 neurons and a batch size of 64 is chosen for the following sections. The Adam optimizer with mean squared error (MSE) optimization function is used to train the network for 10 epochs.

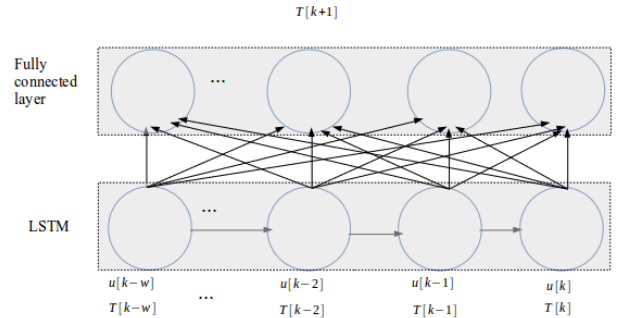


Fig. 2: The overall structure of the LSTM network.

Conventionally, the data set used for training is selected such that it contains more information of system dynamics.

Since randomly generated input signals can provide more information about system dynamics due to the introduced randomness compared to square-wave signals, a training data with two randomly generated Gaussian inputs are used to train the neural network and to obtain a suitable model. The training data set is normalized based on $z = \frac{(x-\mu)}{\sigma}$ to weight the input features equally. The normalized data is then used for the training of the model. Due to the real-time applications of IMDs, standard deviation and mean of the test data can never be acquired prior to validation. Thus, we use mini batches with 10 real-time data points to calculate the standard deviation and mean and normalize the temperature readings accordingly. After the prediction at each time instance k , the normalized prediction result is then converted back to its original form. As a result, the standard deviation and the mean are continuously updated on each time horizon of inputs, thus making the neural network more suitable for online identification and prediction.

Due to the structure of the neural network implemented, a personal computer with Ubuntu 16.04.4, python version 3.5.2, tensorflow 1.4.0 and CUDA version 8.0.6 is used for the training and testing purposes.

IV. RECURSIVE SYSTEM IDENTIFICATION (RSID) ALGORITHM

The recursive system identification algorithm used in this paper for the performance comparison utilizes spatial filtering methods introduced in [19] to divide the data into two components based on spatial dependency. Filtering out the spatial component of the data, predictor-based RSID methods presented in [9] are applied to the filtered data to obtain nonspatial predictions using a vector auto-regressive with exogenous inputs (VARX) model. One-step-ahead VARX predictor can be written as

$$\hat{y}_{k|k-1} = \sum_{i=0}^p \alpha_i u_{k-i} + \sum_{i=1}^p \beta_i y_{k-i} \quad (3)$$

where $\hat{y}_{k|k-1}$ is the nonspatial output at time instant k predicted by the algorithm using a finite window (of size p) of past inputs and outputs.

Additionally, a kernel recursive least squares (KRLS) filter with surprise criterion (SC) is applied to the spatially correlated components to obtain the spatial predictions. Both nonspatial and spatial predictions are then combined to achieve an overall prediction of the system.

V. SIMULATION STUDIES

Thermal dynamics of the UEA is simulated by the multi-physics modeling software COMSOL according to previous work in [16]. The COMSOL model of the UEA is shown in Fig. 3.

The COMSOL board contains two heat sources (H1&H2) and six temperature sensors (S1-S6). Temperature data obtained in the COMSOL software is then used as a reference to demonstrate the performance of the LSTM algorithm.

Eight studies are conducted using the COMSOL model in Figure 3. For all these studies, different square-wave signals

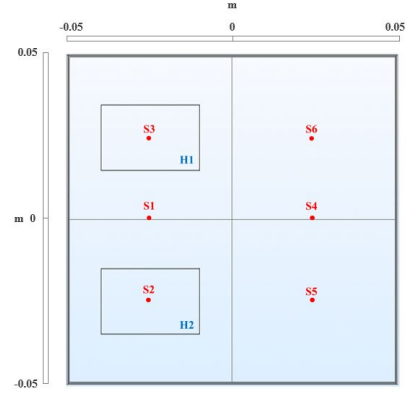


Fig. 3: Board layout in COMSOL with sensors locations labelled in red.

TABLE I: EXPLANATION OF COMSOL STUDIES

Scenarios	Explanation
OD1H	Increase one input from 0.125W to 0.5W while decreasing the other from 0.25W to 0.125W
OD1S	Increase one input from 0.125W to 0.5W while decreasing the other from 0.25W to 0.125W (sudden change)
OS1H	Decrease both input amplitudes from 0.25W to 0.125W
OS2H	Increase both input amplitudes from 0.25W to 0.5W
OS1S	Decrease both input amplitudes from 0.25W to 0.125W (sudden change)
OS2S	Increase both input amplitudes from 0.25W to 0.5W (sudden change)
ODS1H	Decrease inputs of 0.125 and 0.25W to 0.0625W
OSD1H	Increase one input from 0.25W to 0.5W while decreasing the other from 0.25W to 0.125W

of 2400 data samples are generated. In all studies, both inputs have the same 50% duty cycle and a period of 20 seconds. Different COMSOL scenarios are summarized in Table I below. For studies with (*sudden change*) notation in the explanation part in Table I, the change in the input amplitudes happen not exactly at the halfway, i.e. 1200 sec, which is the case for the studies without this notation, but rather the change occurs suddenly at a randomized time instance close to 1200 sec. The plot of power inputs for each study is displayed in Fig. 4.

Figure 5 shows the comparison of simulation results for the scenario "ODS1H". More specifically, in Figure 5(a), comparison between results of the RSID algorithm (in red), results of the LSTM (in green) and temperature readings from the corresponding sensor (in blue) are displayed, and in Figure 5(b) a zoomed-in version of results for sensor 2 is shown.

Mean squared error (MSE) values of the LSTM for each of the simulation studies are summarized and compared to the MSE values of the recursive online identification (RSID) algorithm in Table II. Additionally, the MSE values of the RSID algorithm after the convergence of its adaptive filters,

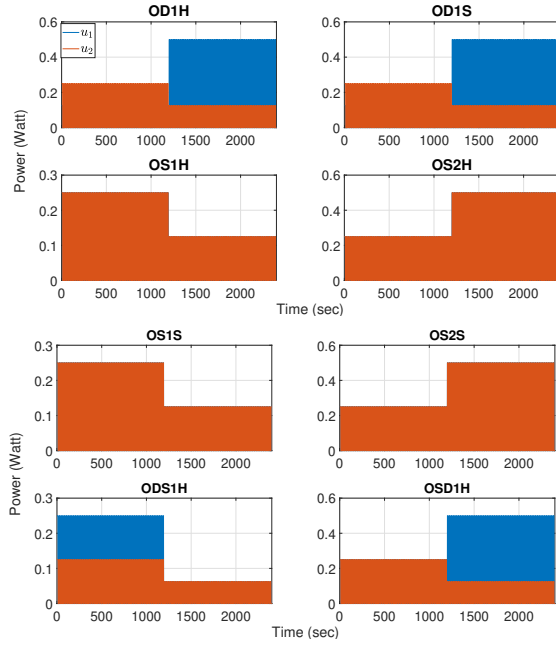


Fig. 4: Plots of power inputs for each simulation study.

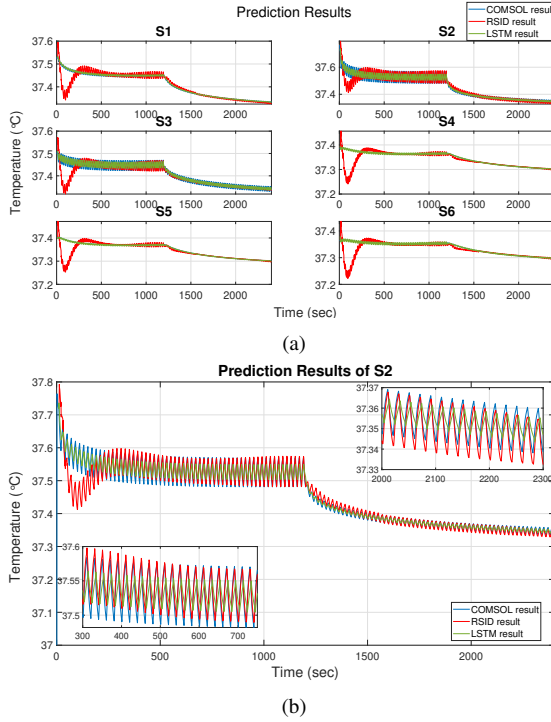


Fig. 5: (a) Simulation results for "ODS1H", (b) Simulation result for S2 (zoomed-in)

i.e. after approximately 400 sec, are included in Table II. The reason for including the "after-convergence" MSE values for RSID is that learning of the system dynamics occurs online due to the structure of the RSID algorithm while the LSTM algorithms uses an offline training to obtain the system model. From the MSE results in Table II, it can be concluded that the LSTM algorithm outperforms the RSID algorithm in most scenarios when the overall performance of

the RSID is examined. However, after the adaptive filters in the RSID algorithm converge, i.e. after approximately 400 sec, the RSID algorithm performs better in half of the simulation scenarios; "OS2H", "OS1S", "OS2S", and "ODS1H". According to the MSE results in Table II, it can be observed that the LSTM algorithm outperforms the RSID algorithm in certain scenarios, mainly in which both inputs have different amplitudes, e.g. "OD1H", "OD1S" etc. Based on the MSE results in Table II and the plots in Figure 5, it can be concluded that the main advantage of implementing the LSTM algorithm is having a stable prediction performance for the period of time for which the convergence of the RSID is not yet achieved. However, once the convergence of the adaptive filters in the RSID is achieved, the recursive algorithm outperforms the LSTM algorithm in some scenarios due to the fact that the LSTM algorithm does not have adaptive learning. Additionally, comparing the prediction results of both algorithms and according to Figure 5, the LSTM algorithm tends to under-predict the temperature values for a given time instance; whereas, the RSID algorithm tends to either predict in the same range as the actual temperature data when the convergence is achieved or it over-predicts for a given time instance. Since the application entails implementing a control scheme to successfully prevent overheating, over-estimating is much preferred than under-estimating. However, the under-prediction in most cases is less than $0.01\text{ }^{\circ}\text{C}$, and could potentially be resolved by implementing an adaptive learning scheme.

Time complexity of both algorithms are averaged across all eight COMSOL studies and compared. The LSTM has a slightly higher time complexity at 9.244 sec compared to the time complexity of the RSID, which is 1.196 sec. However, this higher time complexity of the LSTM algorithm is still acceptable for the real-time application of thermal management in IMDs, since the average time complexity for each data point is approximately 3.866 ms.

TABLE II: MSE RESULTS IN SIMULATION STUDIES ($\times 10^{-3}\text{ }^{\circ}\text{C}$)

Scenarios	LSTM	RSID	RSID*
OD1H	0.272	0.605	0.443
OD1S	0.269	0.696	0.388
OS1H	0.116	0.568	0.119
OS2H	0.476	0.713	0.117
OS1S	0.117	1.117	0.026
OS2S	0.468	1.090	0.191
ODS1H	0.067	0.652	0.028
OSD1H	0.302	0.495	0.316

VI. EXPERIMENTS

A previously developed *in vitro* experimental set-up in [9] with a temperature monitoring and management test vehicle (TMTV) is used in the experimental studies. The TMTV consists of six temperature sensors (LMT70) and two heat sources to emulate the on-board electronics etc. As in [9], a MATLAB GUI is utilized as an interface to display and save the sensor data. The Matlab front-end is connected to

an nRF52 board, through which control signals are sent to the heat sources on the TMTV. The nRF52 board also captures the temperature readings from the TMTV via wireless communication and then transmits them to MATLAB GUI.

For the experiments, the TMTV board is placed in a water-filled container. An illustration of the experiment setup is shown in Figure 7. In order to emulate the heat diffusion effect of blood flow, a marine pump and sponge material are placed in the water-filled container. The marine pump is placed to create water circulation and the sponge is used to ensure a uniform water flow.

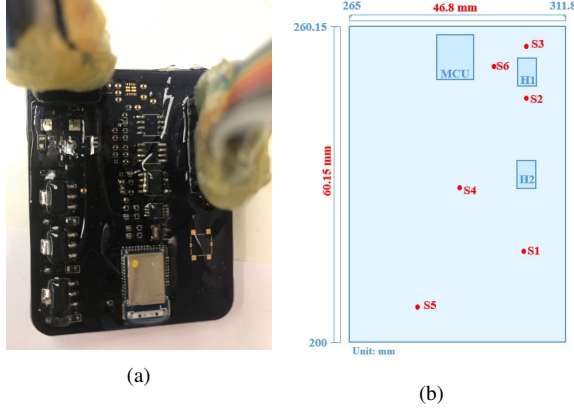


Fig. 6: (a) Developed TMTV system (taken from [9]), and (b) corresponding board layout with two heat sources ($H1$, $H2$) and six temperature sensors shown in red and coordinates of the sensors.

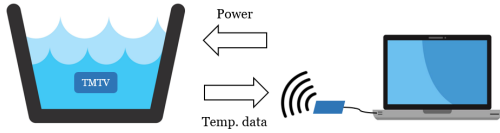


Fig. 7: An illustration of the experimental set-up.

Heat sources on the TMTV admit PWM inputs within the range of $[0, 10000]$. A PWM input of 10000 corresponds to a PWM signal with 0% duty cycle, and similarly a PWM input of 0 denotes a PWM signal with 100% duty cycle [9]. Four experimental studies have been conducted to evaluate the performances of the LSTM and the RSID algorithms under more realistic circumstances. In order to observe the effect of change in PWM inputs, the scenarios used in COMSOL studies have been modified slightly. A description of the four experimental studies are presented in Table III. PWM inputs in all experiments have a period of 20 seconds and the same duty cycle of 50%. For experiments "OD1H", "OD1S", "ODS1H" and "OSD1H" data of lengths 2140, 2053, 2234 and 2041 are acquired respectively.

For all experiments, a training data is used for the batch pre-processing for initialization of the system matrices such that the recursive updating can be initiated after 10 seconds of the input and output data are obtained. Similarly, the same training data is used for the offline training of the LSTM model. For the RSID prediction algorithm, normalized values of the PWM inputs are used. A fixed-lag Kalman filter with zero-mean Gaussian process and measurement noises

TABLE III: EXPLANATION OF EXPERIMENTAL STUDIES

Scenarios	Explanation
OD1H	Increase one PWM input from 5000 to 7500 while decreasing the other from 7500 to 5000
OD1S	Increase one PWM input from 5000 to 7500 while decreasing the other from 7500 to 5000 (sudden change)
ODS1H	Change PWM inputs of 5000 and 2500 to 7500
OSD1H	Increase one PWM input from 5000 to 7500 while decreasing the other from 5000 to 2500

is applied to the experiment data obtained in each study to reduce the noise in data.

Figure 8 shows the comparison of experiment results for the scenario "ODS1H". More specifically, in Figure 8(a), comparison between results of the RSID algorithm (in red), results of the LSTM (in green) and temperature readings from the corresponding sensor (in blue) are displayed, and in Figure 8(b) a zoomed-in version of results for sensor 2 ($S2$) is shown. As can be seen in Figure 8(b), the RSID algorithm performs better in predicting the sudden changes in the system due to its adaptive filters, particularly around 1400 sec when there is a drastic change in temperature reading, but the LSTM algorithm has more stable predictions for approximately the first 400 sec. in which the convergence of the adaptive filters of the RSID is not yet achieved.

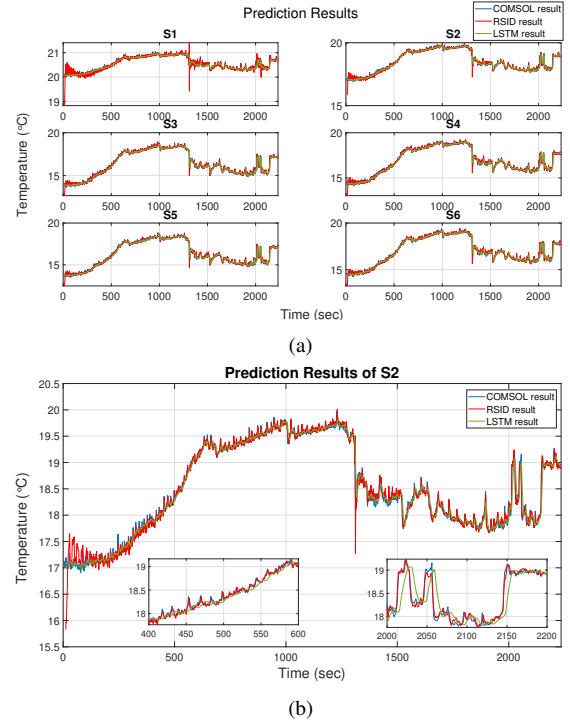


Fig. 8: (a) Experiment results for "ODS1H", (b) Experiment result for $S2$ (zoomed-in).

The MSE values of LSTM for each of experimental studies are summarized and compared to the MSE values of the

RSID algorithm in Table IV. Similar to the COMSOL studies, the MSE values of the RSID algorithm after the convergence is achieved are included as well. From the MSE results in Table IV, it can be concluded that the LSTM algorithm outperforms the RSID algorithm in certain scenarios when the overall performance of the RSID is examined. However, after the initial convergence of the adaptive filters in the RSID algorithm is achieved, the RSID algorithm performs better in two of the four experiments conducted. One exemption to this improved performance of the RSID algorithm after convergence is the "OD1H" scenario in which the MSE value of the RSID algorithm after the convergence is greater than over the overall time period due to the unstable peaks at the halfway when the inputs change. Similar to COMSOL studies, the time complexity of both algorithms for each data point is computed and compared. The average time complexity per data point is 3.704 ms for the LSTM and 0.340 ms for the RSID algorithm respectively.

TABLE IV: MSE RESULTS IN EXPERIMENT STUDIES ($^{\circ}C$)

Scenarios	LSTM	RSID	RSID*
OD1H	0.003	0.358	0.378
OD1S	0.044	0.034	0.010
ODS1H	0.037	0.044	0.008
OSD1H	0.004	0.115	0.065

VII. CONCLUSIONS

As a result of recent technological advances in implantable medical devices, thermal management of such devices plays an important role in maximizing device performance while ensuring safe operation. In this paper, we implemented and compared the long short-term memory network and the recursive system identification method to achieve real-time prediction of the temperature change due to continuous operation of the IMD. This paper focuses on evaluating the performance of both algorithms and investigating the conditions under which one algorithm is more advantageous than the other. Both COMSOL simulations and *in vitro* experiments demonstrate that the main advantage of implementing the LSTM algorithm is to have more reliable predictions for the time period in which the convergence of the adaptive filters in the RSID algorithm is not yet achieved. On the other hand, once the adaptive filters converge, the performance of the RSID algorithm is significantly better than the LSTM in some scenarios due to its adaptive learning capabilities. Additionally, the LSTM algorithm tends to under-predict the temperature values for a given time instance, whereas, the RSID algorithm tends to either predict in the same range as the actual temperature data when the convergence is achieved or it over-predicts for a given time instance. Since the application entails implementing a control scheme to successfully prevent overheating, over-estimating is much preferred than under-estimating. Authors think that this issue of this slight under-prediction in the LSTM results, which is less than $0.01^{\circ}C$ for most cases, could be resolved by implementing an adaptive learning scheme.

For our future work, we are aiming to implement an adaptive learning scheme to the LSTM network to achieve real-time learning of the time-varying system dynamics.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant ECCS-1711447 and Grant CNS-1253390.

REFERENCES

- [1] N. L. Opie, A. N. Burkitt, H. Meffin, and D. B. Grayden, "Heating of the eye by a retinal prosthesis: modeling, cadaver and in vivo study," *IEEE Trans. on Biomed. Eng.*, vol. 59, pp. 339-345, 2012.
- [2] K. Bazaka, M. V. Jacob, "Implantable Devices: Issues and Challenges," *Electronics*, vol. 2, pp.1-34, 2013.
- [3] M. Seemann, N. Zech, M. Lange, J. Hansen, and E. Hansen. "Anesthesiological aspects of deep brain stimulation : special features of implementation and dealing with brain pacemaker carriers," *Der Anaesthetist*, vol. 62, issue 7, pp.549-556, 2013. [Article in German]
- [4] P. S. Ruggera, D. M. Witters, G. von Maltzahn, and H. I. Bassen, "In vitro assessment of tissue heating near metallic medical implants by exposure to pulsed radio frequency diathermy," *Physics in Medicine and Biology*, vol.48, pp.2919-2928, 2003.
- [5] G. Lazzi, "Thermal effects of bioimplants," *IEEE Engineering in Medicine and Biology Magazine*, vol.24, no.5, pp.75-81, 2005.
- [6] S. C. DeMarco, G. Lazzi, W. Liu, J. D. Weiland, and M. S. Humayun, "Computed SAR and thermal elevation in a 0.25-mm 2-D model of the human eye and head in response to an implanted retinal stimulator-Part I: Models and methods," *IEEE Transactions on Antennas and Propagation*, vol.51, no.9, pp.2274-2285, 2003.
- [7] R. Chai, Y. Lai, W. Sun, M. Ghovanloo, and Y. Zhang, "Online Predictive Modeling for the Thermal Effect of Implantable Devices," *Proc. of 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2018.
- [8] R. Chai, Y. Zhang, "Adaptive Thermal Management of Implantable Device," *IEEE Sensors Journal*, vol. 19, issue 3, pp.1176-1185, 2018.
- [9] A. Ermis, Y. Lai, X. Pan, R. Chai, and Y. Zhang, "Recursive Subspace Identification for Online Thermal Management of Implantable Devices," *Proc. of 57th Allerton Conference on Communication, Control, and Computing*, pp.944-949, 2019.
- [10] M. Collotta, G. Pau, R. Caponetto, "A Real-Time System based on a Neural Network Model to Control Hexacopter Trajectories," *2014 International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 2014.
- [11] J.-F. Couchot, K. Deschinkel, M. Salomon, "Active MEMS-based flow control using artificial neural network," *Mechatronics*, vol. 23, issue 7, pp. 898-905, 2013.
- [12] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, 1997.
- [13] A. Richard, A. Mahe, C. Pradalier, O. Rozenstein, and M. Geist, "A Comprehensive Benchmark of Neural Networks for System Identification", *HAL Archives*, HAL Id: hal-02278102, 2019.
- [14] Y. Wang, "A new concept using LSTM Neural Networks for dynamic system identification," *Proc. of 2017 American Control Conference (ACC)*, 2017.
- [15] S. Sadiqbatcha, H. Zhao, H. Amrouch, J. Henkel, and S. X.-D. Tan, "Hot Spot Identification and System Parameterized Thermal Modeling for Multi-Core Processors Through Infrared Thermal Imaging," *Proc. of 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp.48-53, 2019.
- [16] S. Kim, P. Tathireddy, R. A. Normann, and F. Solzbacher, "Thermal impact of an active 3-D microelectrode array implanted in the brain," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol.15, no.4, pp.493-501, 2007.
- [17] B. B. Schwedersky, R. C. C. Flesch and H. A. S. Dangui, "Practical Nonlinear Model Predictive Control Algorithm for Long Short-Term Memory Networks," *IFAC PapersOnLine*, vol. 52, issue 1, pp.468â473, 2019.
- [18] K. Hornik, "Approximation capabilities of multi-layer feed-forward networks", *Neural Networks*, 1991.
- [19] A. Getis, D. A. Griffith, "Comparative Spatial Filtering in Regression Analysis," *Geographical Analysis*, vol. 34, pp. 130-140, 2002.