Predicting Performance using Approximate State Space Model for Liquid State Machines

A. Gorad, V. Saraswat, U. Ganguly
Indian Institute of Technology, Bombay, India, Phone +91 22 2576 7698
ajinkyagorad@ee.iitb.ac.in

***Abstract*—Liquid State Machine (LSM) is a brain-inspired architecture used for solving problems like speech recognition and time series prediction. LSM comprises of a randomly connected recurrent network of spiking neurons. This network propagates the non-linear neuronal and synaptic dynamics. Maass et al. have argued that the non-linear dynamics of LSM is essential for its performance as a universal computer. Lyapunov exponent (μ), used to characterize the non-linearity of the network, correlates well with LSM performance. We propose a complementary approach of approximating the LSM dynamics with a linear state space representation. The spike rates from this model are well correlated to the spike rates from LSM. Such equivalence allows the extraction of a memory metric (τ_M) from the state transition matrix. τ_M displays high correlation with performance. Further, high τ_M systems require fewer epochs to achieve a given accuracy. Being computationally cheap (1800$\times$ time efficient compared to LSM), the τ_M metric enables exploration of the vast parameter design space. We observe that the performance correlation of the τ_M surpasses that of Lyapunov exponent (μ), ($2-4\times$ improvement) in the high-performance regime over multiple datasets. In fact, while μ increases monotonically with network activity, the performance reaches a maxima at a specific activity described in literature as the edge of chaos. On the other hand, τ_M remains correlated with LSM performance. Hence, τ_M captures the useful memory of network activity that enables LSM performance. It also enables rapid design space exploration and fine-tuning of LSM parameters for high performance.**

***Index Terms*—LSM, SNN, State Space model, performance prediction, dynamics, neural networks**

I. Introduction

The brain inspired computational framework of Liquid State Machines (LSMs) was introduced by Maass in 2002 [1]. LSMs consist of a large recurrent network of randomly connected spiking neurons called the reservoir. The components of this network, namely the neurons and synapses, all follow highly non-linear dynamics. Depending on the extent and strength of connectivity, the reservoir propagates this non-linear dynamics in a recurrent manner. Maass et al. have argued that these non-linear operations performed by LSMs allow them to display high performance and universal computational capabilities [1]. Many applications are based on the belief that non-linearity of the LSMs enable powerful data processing [2]–[4]. In [2], non-linear computations were performed on time series data. In [4], LSM was used for movement prediction task and was shown to perform a non-linear technique called Kernal Principal Component Analysis. These are few of the many versatile applications for which LSMs have demonstrated excellent performance [3], [5]–[7].

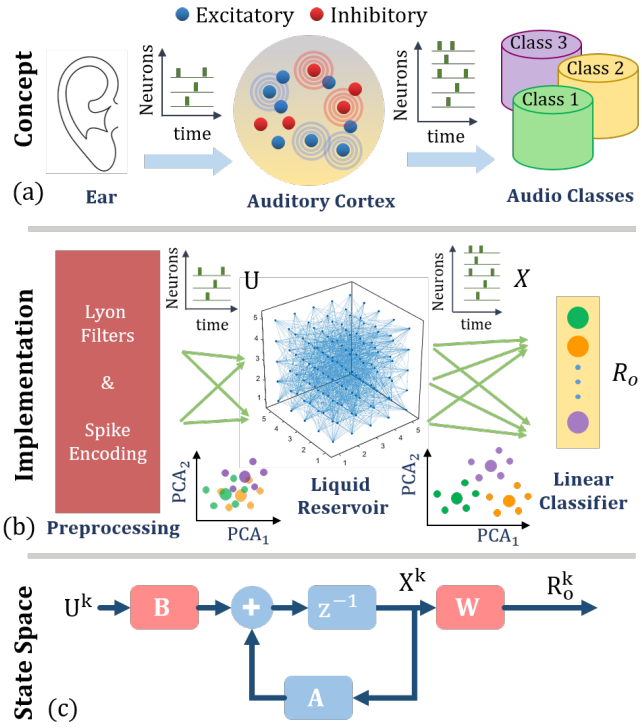

Fig. 1. Architecture of Liquid State Machine (LSM) showing an example for speech recognition. (a) Conceptual description: Spikes generated in the cochlea (ear) travel to the auditory cortex where a randomly connected recurrent neural network acts as a reservoir of a liquid where input spikes produce a "ripple-like" memory in the network. The output of the network transform the non-linearly separable input in low dimensions to a linearly separable output in a higher dimension to enable a simple classifier to perform accurate classification. (b) Implementation of the LSM concept in MATLAB consists of a preprocessing layer (using Lyon's Filter [8]) to generate a spike input, a liquid reservoir of randomly connected recurrent spiking neurons and a linear readout classifier [9]. (c) Proposed a Discrete Linear State Space approximation of LSM with input mapping **B**, state transition matrix **A** and readout weight matrix **W** (described in Section III).

LSMs have a vast set of design parameters with no direct relation with the performance. It is also known that, depending on the reservoir connectivity, an LSM may function in the region of low propagation i.e. resulting reservoir spikes to fade away quickly, while higher propagation may ultimately lead to chaotic dynamics [10], [11]. However, in order to achieve the best performance, the network has to function at an optimal level of activity or, as the literature terms it, at the "edge of chaos" [12]. Many attempts have been made to define metrics

978-1-7281-2009-6/$31.00 ©2019 IEEE paper N-20283.pdf
Personal use is permitted, but republication/distribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.

to capture the network dynamics that correlates with this trend in performance of LSMs [13]–[15], like the class separation [16] and matrix-rank method [10] which was further improved in [17]. Lyapunov exponent is the most successful metric which is well-correlated with the LSM's performance [13] and has the ability to capture the nonlinear dynamics of networks. It characterizes the network by measuring the difference in corresponding outputs for two almost identical inputs as the measure of the network's ability to resolve small differences in inputs. As such, it is not a model for an LSM.

State space is an established mathematical modelling framework especially for time-evolution in linear systems. This framework provides intuition and robust techniques for stability analysis, feedback design for controllers - few of the many utilities that state space modelling has to offer [18].

In this work, we model the LSM as a linear state space model (Fig. 1 (c)). We demonstrate that the spike rates obtained from the linear state space representation are well correlated with the spike rates from the exact simulations of the LSM. The level of similarity or correlation of the linear model with the exact dynamics depends on the region of performance the LSM is operating in. An LSM operating in the region of high accuracy is indeed well modelled by the linear state space. We show that key advantage of a first order model is that it allows us to evaluate a "memory" metric (τ_M) for the system which shows high correlation with the network performance. τ_M provides a computationally cheap approach for exploring the design space of the LSMs which have a vast number of design parameters. In addition, we find that the new proposed metric τ_M surpasses the performance prediction capabilities of the Lyapunov exponent. τ_M is able to capture the optimal performance achieved as a function of the network activity which is believed to be occurring at the "edge of chaos" [12]. For fine tuning performance of LSMs, τ_M is shown to be better than Lyapunov exponent over multiple datasets.

The paper is organized as follows: Section II highlights how LSMs have been implemented to achieve accuracy comparable to the state-of-the-art for the TI-46 speech dataset. In Section III, the state space model is presented to calculate the linear approximation of the LSM behavior and extract the memory metric (τ_M). Section IV consists of the results and discussions, where τ_M is calculated for LSMs exhibiting a range of performances. We show that τ_M significantly outperforms Lyapunov exponent for high performance LSMs.

II. Background

Liquid State Machine (LSM) framework [1] is analogous to the brain. For e.g. external sensory stimuli like sound is converted to spikes through the cochlea in ears. These non-linearly separable spikes go to the auditory cortex which create ripples of network activity and transform them to a linearly-separable output in higher dimensions (Fig. 1 (a)). The implementation of LSM architecture consists of (i) a preprocessing layer where sound is encoded into spike trains. (ii) These spikes are introduced to the randomly connected spiking neural network called liquid or reservoir (Fig. 1 (b)). Here, input spikes produce a wave of spikes which are similar to a pond where raindrops cause ripples to propagate (carrying the memory of the raindrops) and eventually fading away. Thus, a network of fixed synaptic weights and delays in the reservoir spreads the input across time and space (i.e. among neurons). The small number of inputs are translated by the large network of neurons in the LSM to a higher dimensions which improves the performance of a linear classifier.

A. Speech preprocessing

Speech preprocessing stage for an LSM consists of a human ear-like model, namely Lyon's Auditory Cochlear model [8]. It consists of a cascade of second order filters to produce a response for each channel, where it is rectified and low-pass filtered to get a smooth signal. The shape of the second order filters is determined by the quality factor Q and their step size Δ_{step} [19]. Human ears have a large dynamic range (60-80 dB), thus, an Automatic Gain Control (AGC) stage with particular target amplitude A_{m} and timescale τ_{AGC} is incorporated in the Lyon's model. Since the samples in the TI-46 dataset have the sampling rate of 12.5 KHz, the output of this AGC, called cochleogram was decimated in time by the decimation factor *df* of 12 for simulation purposes. This was done with the help of Auditory Toolbox in MATLAB [19]. This decimated output is converted into spike trains using BSA algorithm [20]. A second order filter h_{BSA} was used for this encoding scheme with the time constants of τ_{b1} & τ_{b2} (1), where H is the unit step function. Finally, 77 spike trains were generated for a corresponding input speech sample in the preprocessing stage. Relevant parameters for this stage are given in the Table I.

$$h_{BSA} = (e^{-\frac{t}{\tau_{\text{b1}}}} - e^{-\frac{t}{\tau_{\text{b2}}}})H(t) \tag{1}$$

TABLE I
PREPROCESSING PARAMETERS

Parameter	Value	Parameter	Value
Q	8	*df*	12
Δ_{step}	0.25	τ_{b1}, τ_{b2}	4, 1 ms
A_m	0.0004	τ_{AGC}	32 ms

B. Liquid Reservoir

Liquid Reservoir is taken to be a 5×5×5 3D grid of Leaky Integrate and Fire (LIF) neurons [9] with fraction f_+ as excitatory and the rest as inhibitory neurons in the simulations. Neural dynamics are given by (2)-(3), where V_i is the membrane potential evolving with time constant τ_{Neu}, refractory period T_{rp}, threshold voltage V_{th} and elicited spike time t_s. v is the synaptic input to i^{th} neuron (N_i). w_{ij} is the connection strength from N_i to N_j.

$$\frac{\partial V_i}{\partial t} = -\frac{V_i}{\tau_{\text{Neu}}} + \sum_i \sum_j w_{ij} v_j \tag{2}$$

$$V_i > V_{th} \rightarrow V_i = 0 \ \forall \ t_s < t < t_s + T_{rp} \tag{3}$$

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.

TABLE II
DEFAULT NETWORK PARAMETERS

Parameter	Value	Parameter	Value	Parameter	Value
W_{EE}	3	K_{EE}	0.45	$\tau_{1\mathrm{E}}$	8 ms
W_{EI}	6	K_{EI}	0.3	$\tau_{2\mathrm{E}}$	4 ms
W_{IE}	-2	K_{IE}	0.6	$\tau_{1\mathrm{I}}$	4 ms
W_{II}	-2	K_{II}	0.15	$\tau_{2\mathrm{I}}$	2 ms
W_{in}	± 8	f_+	0.85	λ	2
F_{in}	4	d_s	1 ms		

Neurons are connected with synapses of specific weights by the probabilistic rule given in (4), where $D(N_1, N_2)$ is the distance between neurons N_1 and N_2 in the reservoir, λ is the effective synaptic distance. K is the multiplication factor in connection probability and it can take the values K_{EE}, K_{EI}, K_{IE}, K_{II}. For e.g. K_{EI} represents the probability of connection from excitatory (E) to inhibitory (I) neuron.

$$P(N_1, N_2) = K \cdot e^{-\frac{D^2(N_1, N_2)}{\lambda^2}} \tag{4}$$

Similarly, synaptic weights w_{ij} can take values W_{EE}, W_{EI}, W_{IE}, W_{II} in the liquid (e.g. W_{IE} denotes weight of the connection from inhibitory to excitatory neuron) and are constant in time. Synaptic delays are fixed to d_s for every synapse. All synaptic dynamics are second order and follow the timescales of ($\tau_{1\mathrm{E}}$, $\tau_{2\mathrm{E}}$) for excitatory and ($\tau_{1\mathrm{I}}$, $\tau_{2\mathrm{I}}$) for inhibitory synapses in the reservoir. These timescales correspond to τ_1 and τ_2 in (5).

$$v = \frac{1}{\tau_1 - \tau_2}(e^{-\frac{t-t_s}{\tau_1}} - e^{-\frac{t-t_s}{\tau_2}})H(t - t_s) \tag{5}$$

Each input spike train from the preprocessing stage is given to F_{in} randomly chosen neurons in the reservoir with uniformly distributed synaptic weights of $\pm W_{\mathrm{in}}$. For these synapses, only excitatory timescales were used. Default network parameters for simulations on TI-46 dataset are given in Table. II and III, which are also mentioned in [9].

C. Classifier

To recognize the class of the input from the multidimensional liquid response, a fully connected layer of spiking readout neurons is used. Excitatory timescales were used for all synapses connecting the reservoir to the classifier. Since the only weight updates in an LSM are to happen here, there needs to be a learning rule which learns the weights to extract useful features from the reservoir. After training, the class corresponding to the most spiked neuron is regarded as the classification decision of the LSM for the given input.

We briefly describe the biologically inspired learning rule proposed in [9]. It uses calcium concentration, c which represents the activity of the neuron in the classifier and enables selective weight change during a training phase. Calcium concentration is defined by a first order equation (6) with timescale τ_c. Steady state concentration c_s is approximately given by (7) and is the indicator of spike rate (f_r) of the neuron. Supervised learning with a large forcing current I_{teach} is used depending on the activity and desirability (or undesirability) of the neuron (8). The desired (or undesired) neuron is supplied I_∞ (or $-I_\infty$) current to spike more (or less) for the present training input. For each output neuron, this rule uses a probabilistic weight update according to (9), (10) for the corresponding synapse whenever a pre-neuron in the reservoir spikes. A constant weight update Δw is added or subtracted, with a probability p^+ or p^- respectively, to the present synaptic weight depending on the input class and the limit on the activity (decided by c_θ & Δc). Hence, the weight is increased (or decreased) if the pre-neuron spikes and the classifier neuron is the desired (or undesired) neuron (9), (10). This weight is artificially limited by value W_{lim}. The default parameters are mentioned in Table. III.

$$\frac{\partial c}{\partial t} = -\frac{c}{\tau_c} + \delta(t - t_s) \tag{6}$$

$$c_s = \frac{1}{e^{\frac{1}{\tau_c f_r}} - 1} \approx \tau_c f_r \tag{7}$$

$$I_{teach} = \begin{cases} +I_\infty \cdot H((c_\theta + \delta c) - c), & \text{if desired} \\ -I_\infty \cdot H(c - (c_\theta - \delta c)), & \text{if undesired} \end{cases} \tag{8}$$

$$w \xleftarrow{p^+} w + \Delta w;\ c_\theta < c < c_\theta + \Delta c \tag{9}$$

$$w \xleftarrow{p^-} w - \Delta w;\ c_\theta - \Delta c < c < c_\theta \tag{10}$$

TABLE III
NEURON & CLASSIFIER PARAMETERS

Parameter	Value	Parameter	Value	Parameter	Value
τ_{Neu}	64 ms	c_θ	10	W_{lim}	8
T_{rp}	3 ms	Δc	2	$p^{\pm}$	0.1
V_{th}	20 mV	δc	1	Δw	0.01
I_∞	10000	τ_C	64 ms		

D. Performance

We replicated the setup from [9] and matched the state-of-the-art performance for the chosen reservoir size of 5×5×5 for TI-46 Digits speech dataset (Table. IV). The system is trained for 200 epochs on 500 TI-46 spoken digit samples. Samples used in training and testing consisted of a uniform distribution of 50 samples for each digit '0-9' and 100 samples for each speaker, among 5 female speakers. A time step of 1 ms was used in all the simulations. Performance is evaluated using 5 fold testing where accuracy is averaged over the last 20 epochs [9]. Each simulation was done in Matlab and took a wall-clock time of 14 hours for each run on a Intel Xeon processor running at 2.4 GHz.

E. Reservoir-less Network

A feed-forward, fully-connected, reservoir-less network is used to benchmark against the LSM. We train the same preprocessed input on the same sized classifier. The difference between the performance by this method and the performance by the LSM is the gain (or loss) in the performance by introducing the reservoir. Generally, LSMs benefit from higher dimensional mapping and recurrent dynamics of the network.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
LSM PERFORMANCE ON SPOKEN DIGIT RECOGNITION

Work	Dataset	Accuracy (%)
This work	**TI-46**	**99.09**
Zhang et al. [9]	TI-46	99.10
Verstraeten et al. [3]	TI-46	99.5
Wade et al. [21]	TI-46	95.25
Dibazar et al. [22]	TIDigits	85.5
Tavanaei et al. [23]	Aurora	91

III. METHODOLOGY

This section describes how the discrete state space approximation was developed for the LSM dynamics described in the background and the resulting memory metric.

A. State Space Approximation

To study the dynamics of the spiking trajectories of LSM, we consider the spike rate column vectors for input (U^k), reservoir (X^k) and readout (R_o^k), where each row of the vector corresponds to a instantaneous spike rate at time k for a neuron. Spike rate is calculated as the average number of spikes in a rectangular window of 50 ms. This spike rate activity spans a trajectory in a multi-dimensional space (each dimension represents a different neuron) with time [24].

The input activity U^k gets mapped to the higher dimensional space of the reservoir. The future activity or state of the liquid X^{k+1} can be written as a function f of its present input U^k and the current state X^k (11). Also, since the readout function does not possess any memory, it can simply be the function f_w of reservoir activity X^k (12).

$$X^{k+1} = f(X^k, U^k) \tag{11}$$
$$R_o^k = f_w(X^k) \tag{12}$$

Dynamics of the LSM are approximated using a state space model which is first order and linear (Fig. 1 (c)). Reservoir (11) and the readout (12) are approximated by (13) and (14) respectively using constant matrices A, B and W. Matrix A accounts for the recurrent dynamics of the reservoir, B mimics the input mapping to the reservoir and W resembles the learnt weights.

$$X^{k+1} = A \cdot X^k + B \cdot U^k \tag{13}$$
$$R_o^k = W \cdot X^k \tag{14}$$

Let U, X and R_o be the actual LSM simulated spike rates over time of all the 10 samples after the training. We denote shifted version of matrix X by 1 time step in future as X_{+1}. Hence, (13) for all time steps is represented by (15). We estimate A, B and W by taking the Moore-Penrose inverse (*pinv*) of the combined matrix of X and U by knowing the spike rate of the inputs, reservoir and the readout neurons on the chosen 10 samples (16), (17). For a system with M input neurons, N reservoir neurons and L readout neurons, we get size of A to be $N \times N$, that of B to be $N \times M$ and that of W to be $L \times N$. Concatenation of matrices X and U is represented as $[X|U]$.

$$X_{+1} = [A|B] \cdot [X|U]^\intercal \tag{15}$$
$$\Rightarrow [A|B] = X_{+1} \cdot \text{pinv}([X|U]^\intercal) \tag{16}$$
$$R_o = W \cdot X \Rightarrow W = R_o \cdot \text{pinv}(X) \tag{17}$$

Once A, B and W are determined and by only knowing the input U, we estimate the $\hat{X}$ and, from that $\hat{R}_o$. To evaluate the effectiveness of state space modelling of an LSM, we find the correlation coefficient of the actual response X with the predicted response $\hat{X}$ knowing the input ($U \rightarrow \hat{X}$). We also evaluate other combinations of predictions to see how well this approximation holds. Forward combinations include $U \rightarrow \hat{X}$, $X \rightarrow \hat{R}_o$ and $U \rightarrow \hat{X} \rightarrow \hat{R}_o$. Reverse combinations include $R_o \rightarrow \hat{X}$, $X \rightarrow \hat{U}$ and $R_o \rightarrow \hat{X} \rightarrow \hat{U}$. This correlation coefficient qualifies the ability of a state space to model the LSM (discussed in Section IV).

B. Concept of Memory

For an N dimensional state space represented by X there are N time constants τ_i (18). Memory of such a system can be defined as the mean of the time constants τ_i (19).

$$\dot{X} = -diag(\frac{1}{\tau_1}, \frac{1}{\tau_2}, ..., \frac{1}{\tau_N}) \cdot X \tag{18}$$
$$\tau_M = \frac{1}{N} \sum_{i=1}^{N} \tau_i \tag{19}$$

A discrete first order system with time constant τ_i is represented as (20) using the Euler method. Here, h is the time step of the discretized system which is 1 ms.

$$\frac{x_i^{k+1} - x_i^k}{h} = -\frac{x_i^k}{\tau_i} \;\Rightarrow\; x_i^{k+1} = (1 - \frac{h}{\tau_i}) x_i^k \tag{20}$$

For a system matrix A of size $N \times N$ in a discrete state space system which defines the time dynamics, we get its diagonal entries in vector a (21). From this, we propose the memory metric τ_M for an approximate model of the reservoir (13) to be (22).

$$a = diag(A) \tag{21}$$
$$\tau_M = \frac{1}{N} \sum_{i=1}^{N} \frac{h}{1 - |a_i|} \tag{22}$$

This memory metric is calculated and its relation with performance is explored (presented in Section IV). It is also compared to the previously identified state of the art performance prediction metric Lyapunov exponent (μ) [13]. Lyapunov exponent characterizes the chaotic behaviour of the network by measuring the difference in output for two almost identical inputs as the measure of the network's ability to resolve small differences in inputs. It is calculated as the average over the scaled exponents (μ_i) for classes $i = 1, 2, ..., 10$,

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.

where for each class i, μ_i is calculated from two samples u_{1i}, u_{2i} and their reservoir responses x_{1i}, x_{2i} using (23).

$$\mu_i = ln\frac{||x_{1i}(t) - x_{2i}(t)||}{||u_{1i}(t) - u_{2i}(t)||} \tag{23}$$

C. Simulation Methodology

The reservoir in an LSM has a large number of parameters defining its design space. Study by [25] identified few key parameters, which included synaptic scaling α_w and effective connectivity distance λ. We scale the reservoir weights by a constant factor α_w. We simulate the LSM for speech recognition task using TI-46 dataset and evaluate the performance over 12 different α_w, each comprising of 4 randomly generated reservoir structures for the same parameters.

IV. Results and Discussion

A. Similarity to State Space

We estimate the reservoir spike rates $\hat{X}$ and find them to be well matched with the actual spike rates X for the transformation $U \rightarrow \hat{X}$ using the state space approximation with Pearson Correlation Coefficient (PCC) of 0.92. This slight deviation arises due to non-linearity of spiking neurons. Figure 2 shows the exact reservoir spike rate and the estimated spike rate for each neuron in the reservoir in the high performance region of operation.

Further, correlation coefficients for all estimated spike rates with exact spike rates are evaluated as a function of performance for both the forward and reverse pathways (Fig. 3). Forward estimation gave higher correlation values than the reverse estimation. When estimating the reservoir spikes from the input spikes ($U \rightarrow \hat{X}$), we find that this model is a good fit when error is small. In other words, the mapping of an LSM to a state space is more precise when the accuracy of the network is high.

We obtain the state space model estimation of the readout f_w by W with correlation greater than 0.85 for all ranges of error ($X \rightarrow \hat{R}_o$). The transition $U \rightarrow \hat{X} \rightarrow \hat{R}_o$ is the combination of intermediate transitions $U \rightarrow \hat{X}$ & $X \rightarrow \hat{R}_o$ and hence, the correlation between R_o and $\hat{R}_o$, given the input U, is less than the correlations of both the intermediate transitions.

The mapping from readout to reservoir ($R_o \rightarrow \hat{X}$) gives low correlation coefficients. This is expected as the readout need not represent all the information in the liquid and also the classifier has a lower dimensionality of 10 (compared to 125 in the reservoir). In comparison, when we estimate the input (of dimensionality 77) from the reservoir $X \rightarrow \hat{U}$, the correlation is close to 0.8, and the overall estimation correlation $R_o \rightarrow \hat{X} \rightarrow \hat{U}$ is less than that of both $R_o \rightarrow \hat{X}$ and $X \rightarrow \hat{U}$.

B. Performance as a function of Network Activity

We calculate the performance and the memory metric in different regimes of the LSM operation. For different reservoir synaptic weight scaling α_w, we obtain low-propagation (sparse activity $\alpha_w = 0.5$), significant-propagation (normal activity $\alpha_w = 0.8$), in-chaos (high activity $\alpha_w = 2$) and saturation (all neurons spiking $\alpha_w = 5$), as shown in Fig.4 (a). Performance of the LSM for various synaptic weight scaling is calculated (Fig. 4 (b)). Also τ_M of the state space modelled system is calculated using (22), and its variation with α_w is shown in Fig. 4 (c).

For low-propagation in the reservoir (α_w = 0.5), there is mainly higher dimensional mapping of the input and very limited dynamics that take place in the reservoir. This higher dimensional mapping gives a 5% increase to the accuracy i.e. 98% as compared to 93% from the reservoir-less approach (Fig. 4). Increasing the weights ($\alpha_w = 0.8$) allows the input spikes to propagate through the reservoir and the memory

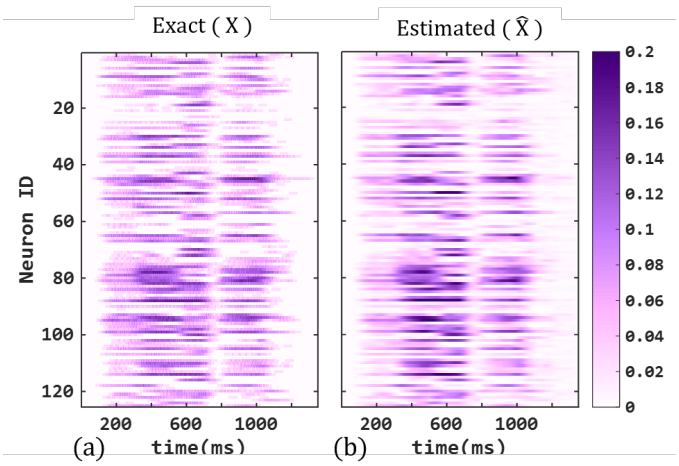

Fig. 2. Well correlated (Pearson Correlation Coefficient i.e. PCC = 0.92). Reservoir spike rates for (a) LSM X and (b) state space estimation $\hat{X}$ for each neuron in the reservoir as a function of time for a speech sample from TI-46 dataset.

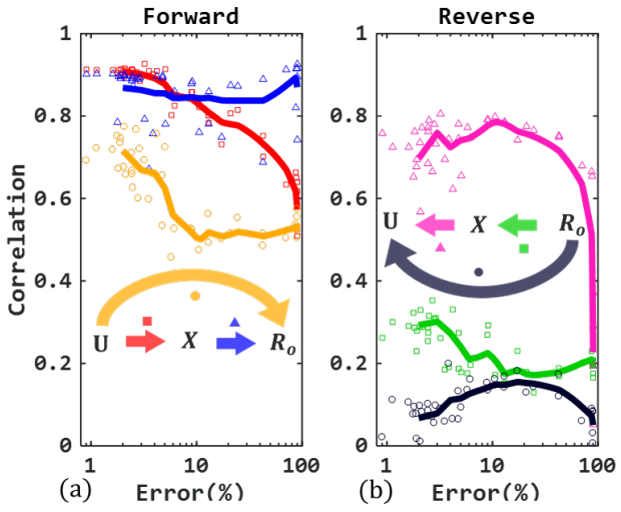

Fig. 3. Average correlation coefficient between LSM and State Space Model spike rates vs. error (log scale) for all the possible transformations on 500 samples of TI-46 Digits. Forward correlations (a) are stronger than backward correlations (b) and they improve in low error regime.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.

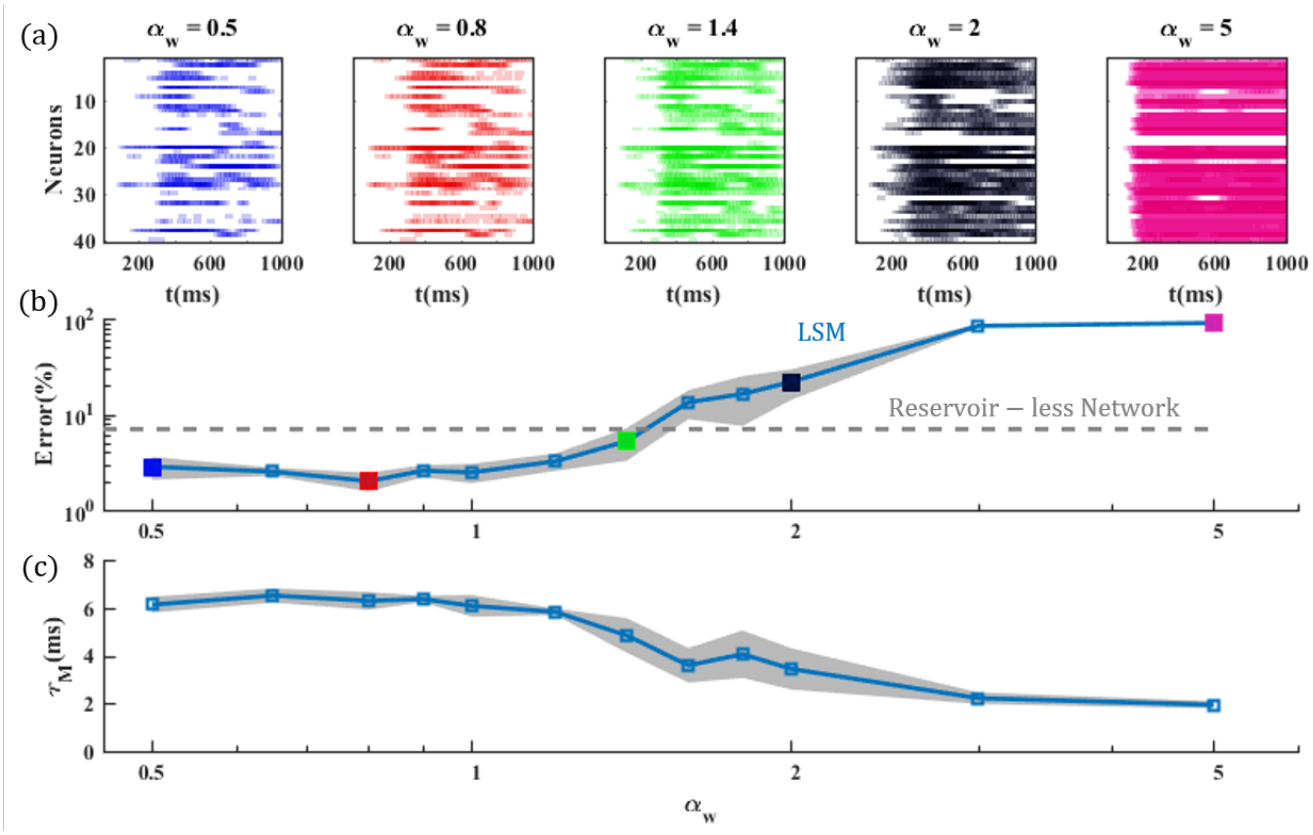

Fig. 4. (a) Spike rate in the reservoir vs. synaptic scaling (α_w) for 40 neurons resulting from a TI-46 speech sample. At the optimal value of α_w = 0.8, the spike propagation is optimal which related to high accuracy. At lower α_w (<0.8), the spike propagation is weak and reduces accuracy. For higher α_w (>0.8), the spike propagation is excessive, leading to degraded performance. Higher α_w = 5, leads to relentless spiking. (b) Error vs. synaptic weight scaling and (c) Memory metric τ_M vs. synaptic scaling both show non-monotonic but correlated dependence. One sigma variation over different runs is also plotted in grey in (b) and (c). The performance of a reservoir-less network, where the input is directly connected to the classifier, is shown in comparison to an LSM.

dynamics which contribute to the performance. This results in the reduction of error to 1.5%. The size of the LSM and the activity together maximize the accuracy which corresponds to an increase in the memory metric .

However, increasing the activity does not always correspond to better performance and hence it may not contribute to the memory of useful information. τ_M decreases as the activity (or disorder) further increases and the LSM enters chaos. In saturation regime (α_w = 5), there is little information as the pattern is lost permanently.

We think that the linear state space approach helps in characterizing the memory of LSM since useful information is a function of the past reservoir activity and the current input. This allows a good approximation of reservoir trajectories which provides an estimate of how the system evolves with the addition of new input. As the LSM enters chaotic regimes, useful information is in fact destroyed due to the disturbance of this trajectory. Such breaks in the trajectory cannot be captured by a smooth state space transition.

C. Comparison with the Lyapunov Exponent

From the above discussion, the key property of the proposed memory metric (τ_M) that emerged on varying the synaptic weight scaling (α_w) is that τ_M has a very high correlation with the recognition performance, i.e. large memory results in higher accuracy (Fig. 5 (a)). A more significant trend is observed in high accuracy region. PCC of τ_M with the accuracy was found to be 0.93 over all the possible error values and it increased to 0.95 for the points with small errors (Fig. 5 (b)). As a comparison benchmark, we also calculated the absolute PCC of the previously identified [13] state-of-the-art performance prediction metric, Lyapunov exponent (μ) with accuracy as 0.95 which rapidly dropped down in the high accuracy region of operation (Fig. 5 (b)). In other words, the overall performance correlation for both the metrics is at par in general, however, τ_M (Fig. 5 (c)) performs significantly better than μ (Fig. 5 (d)) for the low error regions. This is due to the monotonic nature of μ with respect to synaptic scaling. In general the accuracy of LSMs increases as the activity increases but then reduces with the onset of chaos. τ_M captures this optimal activity threshold precisely increasing its utility over the μ.

In addition, this behavior of τ_M is general and can be extended to other datasets. A well known strategy for generating a test dataset [25], [26] is used. Here, 10 input spike train templates of 40 Hz (Poisson distributed), each comprising of 10 spike channels with a sample length of 200 ms are

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.

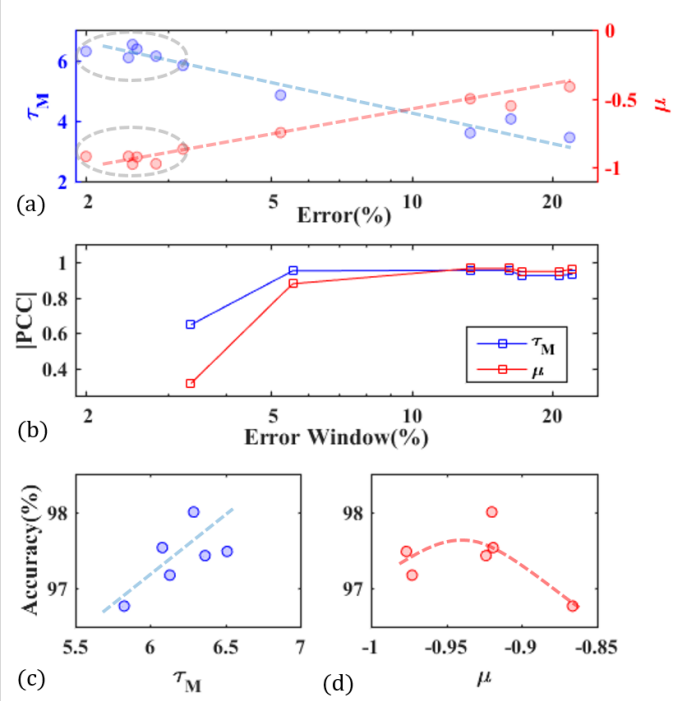

Fig. 5. (a) Correlation of τ_M and μ with error for TI-46 speech dataset. (b) For overall error range, the correlation values are comparable for τ_M and μ (0.96). Correlation of τ_M (0.64) with accuracy is 2× higher than that of μ (0.31) at lower errors is also seen by the low error expanded view behaviors in (c) and (d) of the encircled regions in (a)

constructed. The spikes in each template are jittered temporally with standard deviation of 16 ms to generate 50 patterns. This dataset is trained for 100 epochs with a set of reservoir parameters mentioned in [25] and the synaptic scaling is varied from 0.1 to 4. PCC of τ_M with performance was found to be 0.87 (Fig. 6 (a)), which is significantly greater than PCC of 0.18 (Fig. 6 (b)) for μ. This suggests τ_M is a better predictor for performance across datasets.

Another property of τ_M is the associated time efficiency

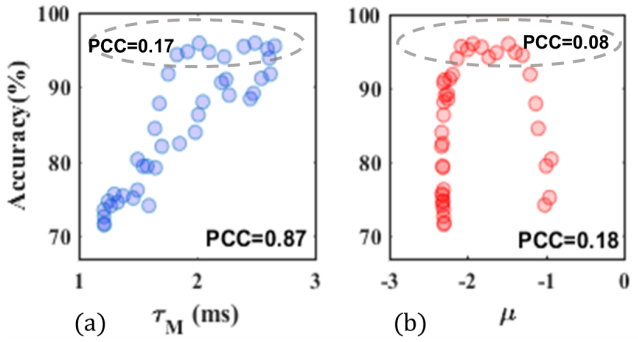

Fig. 6. Performance dependency with τ_M and μ for generated Poisson spike dataset. Synaptic weight scaling α_w was varied. Average accuracy over last 20 epochs from total of 100 epochs for 2-fold testing is shown. Significantly better correlation ($> 4\times$) to performance is observed in case of τ_M (0.87) compared to μ (0.18) for the network. Correlation in high performance region was found to be 0.17 for the τ_M, which is $2\times$ compared to 0.08 for μ.

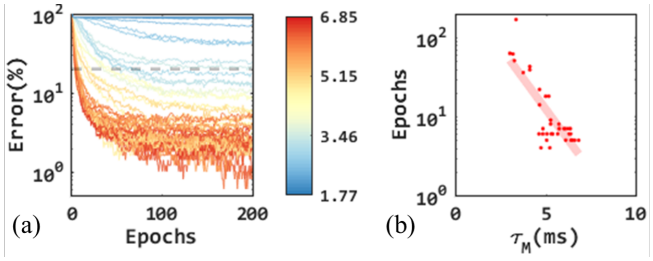

Fig. 7. (a) Error vs epochs for different τ_M (ms) (color) for TI-46 dataset. (b) Number of epochs required for achieving 80% accuracy (dashed line in (a)). For higher τ_M lower epochs are required to achieve the same accuracy.

of the system to achieve a given accuracy. Figure 7 (a) shows that a large number of high τ_M systems achieve greater performance in fewer epochs. For accuracy to fall below a specified error, an exponential relationship is observed between the number of epochs required and the associated memory metric of the system (Fig. 7 (b)). Hence, the τ_M has a direct impact on the learning rate of the classifier with faster learning being enabled in high τ_M systems.

D. Computational Efficiency

Given a parameter set and having established that high τ_M implies high performance, in order to calculate τ_M from State Space it takes 10 samples and to calculate μ it takes 20 samples. However, to calculate accuracy from the LSM, it takes 500 samples and a reasonable number of training epochs ($\sim$200) for weights to converge. Hence, design space optimization based on τ_M is highly accelerated. One simulation of an exact LSM takes 1.5 hours on average, if the system is trained for 20 epochs. In comparison, one τ_M extraction from state space modelling takes only 3 seconds providing 1800× speed up. Further, τ_M calculation is $2\times$ faster than the calculation for the μ.

E. Design Space Search

Given the performance predictive properties of τ_M, the design space exploration is greatly simplified for LSMs. As highlighted in the simulation methodology, LSMs can be tuned by varying a large number of parameters. To highlight the utility of τ_M in this regard, we use previously identified synaptic scaling (α_w) and effective connectivity λ as the key performance parameters [25]. We compare the accuracy with the calculated τ_M and μ over the same parameter space. The accuracy shows a region of maxima (dark green - Fig. 8 (a)) and falls off on either side over the parameter space. Similar trend is seen for the τ_M obtained over the same parameter space (Fig. 8 (b)). Hence, calculation of τ_M from state space approximation provides an efficient method to identify the correct parameter values for high performing LSM. Again, as discussed earlier in Fig. 5, μ does not capture the performance optima unlike τ_M and is monotonic in nature with the network activity (Fig. 8 (c)). Correlation was found to be 0.90 for the memory metric and 0.18 for the Lyapunov exponent in the high

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.

performance region (Accuracy$> 85\%$) for the design space search conducted over α_w and λ (Fig. 8).

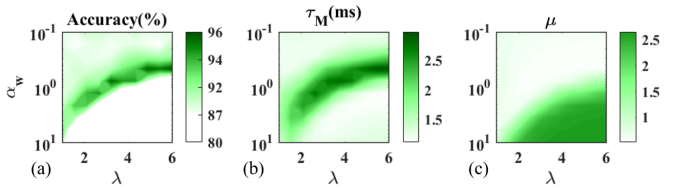

Fig. 8. Design space exploration over reservoir synaptic scaling α_w and effective connectivity λ for Poisson spikes dataset. On the grid of 10x10, shown are the (a) LSM average accuracy over last 5 epochs from total 20 epochs with 2-fold testing (b) τ_M (c) μ. High performance regions are clearly captured by the memory metric, while Lyapunov exponent does not capture the optimal space as it shows a monotonic trend. Higher PCC (5×) with performance was observed for τ_M (0.90) compared to μ (0.18).

V. Conclusion

It is widely believed that a recurrent network of non-linear elements producing highly non-linear information processing enables high performance in recognition tasks. In this paper, we present an alternative interpretation of LSM, where we approximate it with a linear state space model. We demonstrate high correlation of the model response with the LSM. Equivalence with the state space allows the definition of a memory metric which accounts for the additional performance enhancement in LSMs over and above the higher dimensional mapping. The utility of τ_M as a performance predictor is general across datasets and is also responsible for efficient learning of the classifier. We compare and highlight the advantages of τ_M over the state-of-the-art performance predictor Lyapunov exponent μ, and a 2-4× improved correlation to accuracy is observed for τ_M. τ_M captures the maximum performance region (optimal network activity) called the "edge of the chaos". In contrast, Lyapunov exponent is monotonic with network activity - resulting in poor performance prediction at high performance. Further, the computational efficiency of the state space model to compute τ_M (1800× more efficient compared to LSM) enables rapid design space exploration and parameter tuning.

References

[1] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *NEURAL COMPUTATION*, vol. 14, no. 11, pp. 2531–2560, 2002.

[2] Maass, Wolfgang and Natschläger, Thomas and Markram, Henry, "Computational models for generic cortical microcircuits," *Computational neuroscience: A comprehensive approach*, vol. 18, pp. 575–605, 2004.

[3] D. Verstraeten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout, "Isolated word recognition with the liquid state machine: a case study," *INFORMATION PROCESSING LETTERS*, vol. 95, no. 6, pp. 521–528, 2005.

[4] H. Burgsteiner, M. Kröll, A. Leopold, and G. Steinbauer, "Movement prediction from real-world images using a liquid state machine," *APPLIED INTELLIGENCE*, vol. 26, no. 2, pp. 99–109, 2007.

[5] R. de Azambuja, F. B. Klein, S. V. Adams, M. F. Stoelen, and A. Cangelosi, "Short-term plasticity in a liquid state machine biomimetic robot arm controller," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 3399–3408.

[6] J. L. Rosselló, M. L. Alomar, A. Morro, A. Oliver, and V. Canals, "High-density liquid-state machine circuitry for time-series forecasting," *INTERNATIONAL JOURNAL OF NEURAL SYSTEMS*, vol. 26, no. 05, p. 1550036, 2016.

[7] A. Das, P. Pradhapan, W. Groenendaal, P. Adiraju, R. T. Rajan, F. Catthoor, S. Schaafsma, J. L. Krichmar, N. Dutt, and C. Van Hoof, "Unsupervised heart-rate estimation in wearables with liquid states and a probabilistic readout," *NEURAL NETWORKS*, vol. 99, pp. 134–147, 2018.

[8] R. Lyon, "A computational model of filtering, detection, and compression in the cochlea," in *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7. IEEE, 1982, pp. 1282–1285.

[9] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A digital liquid state machine with biologically inspired learning and its application to speech recognition," *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, vol. 26, no. 11, pp. 2635–2649, 2015.

[10] B. Schrauwen, L. Büsing, and R. A. Legenstein, "On computational power and the order-chaos phase transition in reservoir computing," in *Advances in Neural Information Processing Systems*, 2009, pp. 1425–1432.

[11] Y. Jin and P. Li, "Performance and robustness of bio-inspired digital liquid state machines: A case study of speech recognition," *NEUROCOMPUTING*, vol. 226, pp. 145–160, 2017.

[12] R. Legenstein and W. Maass, "What makes a dynamical system computationally powerful," *New directions in statistical signal processing: From systems to brain*, pp. 127–154, 2007.

[13] J. Chrol-Cannon and Y. Jin, "On the correlation between reservoir metrics and performance for time series classification under the influence of synaptic plasticity," *PLOS ONE*, vol. 9, no. 7, p. e101792, 2014.

[14] E. Goodman and D. Ventura, "Spatiotemporal pattern recognition via liquid state machines," in *The 2006 ieee international joint conference on neural network proceedings*. IEEE, 2006, pp. 3848–3853.

[15] W. Maass, R. A. Legenstein, and N. Bertschinger, "Methods for estimating the computational power and generalization capability of neural microcircuits," in *Advances in neural information processing systems*, 2005, pp. 865–872.

[16] D. Norton and D. Ventura, "Improving liquid state machines through iterative refinement of the reservoir," *NEUROCOMPUTING*, vol. 73, no. 16-18, pp. 2893–2904, 2010.

[17] Q. Wang, Y. Jin, and P. Li, "General-purpose lsm learning processor architecture and theoretically guided design space exploration," in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2015, pp. 1–4.

[18] N. S. Nise, *Control Systems Engineering*. John Wiley & Sons, 2007.

[19] M. Slaney, "Auditory toolbox," *Interval Research Corporation, Tech. Rep*, vol. 10, no. 1998, 1998.

[20] B. Schrauwen and J. Van Campenhout, "Bsa, a fast and accurate spike train encoding scheme," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 4. IEEE, 2003, pp. 2825–2830.

[21] J. J. Wade, L. J. McDaid, J. A. Santos, and H. M. Sayers, "Swat: a spiking neural network training algorithm for classification problems," *IEEE TRANSACTIONS ON NEURAL NETWORKS*, vol. 21, no. 11, pp. 1817–1830, 2010.

[22] A. A. Dibazar, D. Song, W. Yamada, and T. W. Berger, "Speech recognition based on fundamental functional principles of the brain," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, vol. 4. IEEE, 2004, pp. 3071–3075.

[23] A. Tavanaei and A. S. Maida, "A spiking network that learns to extract spike signatures from speech signals," *NEUROCOMPUTING*, vol. 240, pp. 191–199, 2017.

[24] D. V. Buonomano and W. Maass, "State-dependent computations: spatiotemporal processing in cortical networks," *NATURE REVIEWS NEUROSCIENCE*, vol. 10, no. 2, p. 113, 2009.

[25] H. Ju, J.-X. Xu, E. Chong, and A. M. VanDongen, "Effects of synaptic connectivity on liquid state machine performance," *NEURAL NETWORKS*, vol. 38, pp. 39–51, 2013.

[26] S. Luo, H. Guan, X. Li, F. Xue, and H. Zhou, "Improving liquid state machine in temporal pattern classification," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2018, pp. 88–91.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY. Downloaded on October 07,2021 at 06:16:58 UTC from IEEE Xplore. Restrictions apply.