

# Máquina de estado líquido con reservorio de neuronas pulsantes a tiempo discreto para clasificación espacio-temporal

Christian Alba-Cisneros<sup>1</sup>, Andrés Espinal<sup>2</sup>,  
Manuel Ornelas-Rodríguez<sup>1</sup>, Horacio Rostro-Gonzalez<sup>3</sup>,  
Marco Sotelo-Figueroa<sup>2</sup>, Jorge Soria-Alcaraz<sup>2</sup>,  
Héctor Puga-Soberanes<sup>1</sup>, Raúl Santiago-Montero<sup>1</sup>

<sup>1</sup> Tecnológico Nacional de México,  
Instituto Tecnológico de León,  
México

<sup>2</sup> Universidad de Guanajuato,  
División de Ciencias Económico Administrativas,  
México

<sup>3</sup> Universidad de Guanajuato,  
División de Ingenierías Irapuato-Salamanca,  
México

{m19240999, manuel.ornelas}@leon.tecnm.mx,  
{aespinal, hrostro}@ugto.mx

**Resumen.** La Máquina de Estado Líquido (LSM por sus siglas en inglés -Liquid State Machine-) es un modelo computacional capaz de tratar naturalmente datos espacio-temporales para realizar tareas como clasificación. Básicamente, una LSM genera respuestas descriptivas de los datos espacio-temporales de entrada mediante un reservorio de neuronas pulsantes; el cual no es entrenado y tampoco dependiente de la tarea a realizar. Posteriormente, representaciones de las respuestas son procesadas por una función de aprendizaje simple; que es entrenada y dependiente de la tarea a llevar a cabo. En este trabajo, se propone utilizar un modelo de neurona pulsante relativamente sencillo, de bajo costo computacional y que trabaja a tiempo discreto como unidad de procesamiento del reservorio en una implementación estándar de LSM. La configuración propuesta de LSM es usada para clasificar la base de datos espacio-temporal Poker-DVS. Los resultados presentados y analizados se obtienen a partir de procesos de clasificación para todas las combinaciones a pares de las 4 clases y con el total de clases disponibles en la base de datos.

**Palabras clave:** Máquina de estado líquido, neurona pulsante, clasificación espacio-temporal, poker-DVS.

## Liquid State Machine with Reservoir of Discrete Time Spiking Neurons for Spatio-Temporal Classification

**Abstract.** The Liquid State Machine (LSM) is a computational model capable of naturally processing spatio-temporal data to perform tasks such as classification.

Basically, a LSM generates descriptive responses of spatio-temporal input data by means of a reservoir of spiking neurons; which is neither trained nor dependent on the task to carry out. Later, the descriptions of the responses are processed by a simple learning function; that is trained and dependent on the task to be performed. In this work, it is proposed to use a relatively simple spiking neuron model, with computationally low-cost and working at discrete time as reservoir's processing unit in a standard LSM implementation. The proposed LSM configuration is used to classify the Poker-DVS spatiotemporal database. The results presented and analyzed are obtained from classification processes for all pairwise combinations of the 4 classes and with the total of classes available in the database.

**Keywords:** Liquid state machine, spiking neuron, spatio-temporal classification, poker-DVS.

## 1. Introducción

La Máquina de Estado Líquido (LSM por sus siglas en inglés -Liquid State Machine-) [20] es uno de los modelos fundamentales del *Cómputo Reservoir* (RC por sus siglas en inglés -Reservoir Computing-), paradigma relativamente nuevo que permite llevar a cabo tareas de reconocimiento de patrones espacio-temporales, como reconocimiento de voz [27, 35], predicción de series de tiempo [25], clasificación de estilos musicales [12], entre otros.

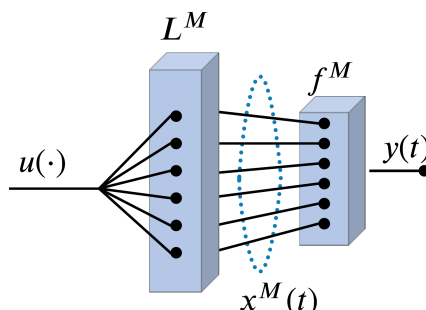
Este paradigma se caracteriza por usar una Red Neuronal Recurrente (RNN por sus siglas en inglés -Recurrent Neural Network-) aleatoria como un reservorio no entrenado del cual se analizan sus respuestas a estímulos de entrada mediante una función externa simple de aprendizaje [33, 16, 17]; por lo tanto, RC provee una forma de aprovechar el poder computacional de las RNNs evitando las dificultades de su entrenamiento [3].

A diferencia de los otros modelos fundamentales del RC, Máquina de Estado de Eco [10] y la Regla de Aprendizaje de Decorrelación de Retropropagación [31], la LSM constituye su reservorio (también llamado líquido) con unidades de procesamiento que tratan naturalmente datos espacio-temporales; estas son neuronas pulsantes [5].

De tal manera, el reservorio de la LSM es una Red Neuronal Pulsante (SNN por sus siglas en inglés -Spiking Neural Network-); las SNNs son consideradas la tercera generación de las Redes Neuronales Artificiales (ANNs por sus siglas en inglés -Artificial Neural Network) [18].

El desempeño de las LSMs se relaciona con sus dos propiedades: separación y aproximación. La propiedad de separación refiere a la capacidad de una LSM para reflejar respuestas distintas para diferentes datos espacio-temporales de entrada y la propiedad de aproximación trata sobre la capacidad de la LSM para distinguir y transformar diferentes estados internos del reservorio en la salida deseada [20, 19].

Diversas investigaciones han enfocado sus esfuerzos en lograr mejoras en el desempeño de las LSM en tareas de reconocimiento de patrones espacio-temporales con base a estas propiedades; por ejemplo: la elección del modelo de neurona pulsante a utilizar en el reservorio [7], la exploración de la arquitectura de la SNN como



**Fig. 1.** Esquema de una máquina de estado líquido, la cual transforma flujos de entrada  $u(\cdot)$  en flujos de salida  $y(\cdot)$ ,  $L^M$  es el líquido, mientras que  $x^M(t)$  es el estado líquido en el tiempo  $t$  el cual es transformado mediante un mapeo de lectura para generar una salida.  $y(t)$  (tomada de [20]).

reservorio [9], el estudio y análisis de la conectividad sináptica del reservorio y sus efectos [11], la exploración y optimización de diferentes arquitecturas de la SNN usada como reservorio [9, 8], diseño de las representaciones de las respuestas descriptivas de la LSM a datos espacio-temporales [24] y reglas de aprendizaje para modificar los pesos de las conexiones sinápticas del reservorio [23, 22].

Un aspecto crucial que impacta ambas propiedades de la LSM es el modelo neuronal usado en el reservorio [7]. El presente trabajo tiene como propósito principal analizar el comportamiento y desempeño del modelo de neurona pulsante Belson-Mazet-Soula [30] (modelo BMS -nombrado así por sus autores [1]-) como unidad de procesamiento del reservorio de una implementación estándar de LSM (es decir, como se propone originalmente en [20]).

La neurona pulsante BMS es un modelo sencillo que trabaja a tiempo discreto y de bajo costo computacional [1], estas características motivan el desarrollo de la presente investigación debido que RNAs densas y con una cantidad significativa de parámetros pueden ser simuladas en equipos de cómputo comerciales; por ejemplo, para toda la experimentación de este trabajo se utilizó una laptop con AMD Ryzen 5 PRO 2500U 2.00 GHz y 8 GB en RAM.

La LSM con reservorio basado en el modelo BMS es probada en tareas de clasificación espacio-temporal usando la base de datos Poker-DVS; con esta base de datos se efectuaron diversas pruebas de clasificación mediante las combinaciones a pares de las 4 diferentes clases (Club, Diamond, Heart, Spade) y usando todas las clases. Los resultados obtenidos son analizados y comparados con algunos reportados en el estado del arte.

El resto del artículo se estructura de la siguiente manera. La sección 2 describe la LSM a través de los elementos que la conforman y la manera en que se utilizan en este trabajo. El modelo BMS de neurona pulsante es explicado en la sección 3. La experimentación y resultados obtenidos se muestran en la sección 4. Por último, las conclusiones y trabajo a futuro son comentadas en la sección 5.

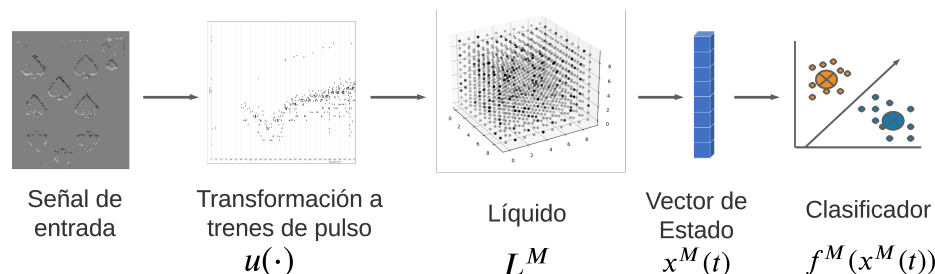


Fig. 2. Descripción extendida de la LSM.

## 2. Máquina de estado líquido

La LSM [20] es un modelo para realizar tareas de reconocimiento de patrones espacio-temporales, el cual aprovecha el poder de las RNN sin efectuar un entrenamiento [3]. Básicamente, la LSM está compuesta por dos partes [22]: una SNN recurrente  $L^M$  que actúa como reservorio (también llamado líquido o filtro) y un mapa de lectura  $f^M(x^M(t))$  que procesa las representaciones  $x^M(t)$  de las respuestas del reservorio a la señal espacio-temporal entrante  $u(s)$  (ver Figura 1).

En otras palabras, una señal entrante  $u(s)$  es procesada por el reservorio  $L^M$ , el cual actúa como un filtro temporal. De la respuesta del reservorio  $L^M$  a la señal entrante  $u(s)$  se extrae una representación  $x^M(t)$  la cual se denomina como vector de estado (Ec. 1):

$$x^M(t) = (L^M u)(t), \quad (1)$$

$$y(t) = f^M(x^M(t)). \quad (2)$$

En este trabajo, el vector estado  $x^M(t)$  es un arreglo binario del tamaño de la cantidad de unidades de procesamiento en el reservorio y su valor es definido para cada componente, como 1 si la neurona correspondiente disparó en el tiempo total de simulación, de lo contrario es 0; a pesar de ser una idea sencilla para definir el vector de estado, se usó de esta forma debido a las características que las neuronas tienen como detectores de coincidencias [14], con el supuesto que, estímulos de entrada de la misma clase activen el mismo conjunto de neuronas y que este difiera lo suficiente para estímulos de entrada de otra clase.

Finalmente, la salida  $y(t)$  (Ec. 2) proviene del mapa de lectura  $f^M(x^M(t))$  que procesa cada momento  $(t)$  el estado líquido actual  $x^M(t)$ . El vector de estado  $x^M(t)$  obtenido del reservorio  $L^M$  es clasificado por una función sencilla de aprendizaje  $f^M(x^M(t))$ ; la cual se encarga de realizar la tarea de reconocimiento de patrones.

En este trabajo, la función de aprendizaje, es un clasificador de mínima distancia (ver [4] para una descripción detallada).

La Figura 2 bosqueja de manera general la implementación en este trabajo, una señal obtenida tomada por el sensor DVS del palo de la baraja es transformada a trenes de pulsos los cuales serán la señal de entrada  $u(s)$ , dicha señal es presentada al reservorio  $L^M$  del cual en el tiempo  $t$  se recupera un vector de estado  $x^M(t)$  (vector binario resultado de la activación -1- o no activación -0- de cada neurona en el reservorio), este

**Tabla 1.** Distribución de patrones en Poker-DVS.

Clase	No de patrones
Club (C)	30
Diamond (D)	43
Heart (H)	23
Spade (S)	35

vector de estado es presentado a la función  $f^M$  (clasificador de mínima distancia) el cual lo clasifica en una de las posibles categorías del problema generando la señal de salida  $y(t)$  de la LSM.

### 3. Modelo BMS de neurona pulsante

El modelo BMS [30] es una discretización derivada del conocido modelo de neurona pulsante de Integración y Disparo [6] (I&F por sus siglas en inglés -Integrate and Fire) la cual proporciona una correspondencia uno a uno entre el potencial de membrana y el estado de activación (representado como un código binario) de la neurona [26]. El modelo BMS está definido por la Ec. 3 y la Ec. 4 que describen el potencial sináptico y el estado de disparo de la  $i$ -ésima neurona  $V_i$  en el tiempo  $k$ ; respectivamente:

$$V_i[k] = \gamma V_i[k-1](1 - Z_i[k-1]) + \sum_{j=1}^N W_{ij} Z_j[k-1] + I_i^{ext}, \quad (3)$$

$$Z_i[k] = \begin{cases} 1 & \text{si } V \geq \theta, \\ 0 & \text{en otro caso,} \end{cases} \quad (4)$$

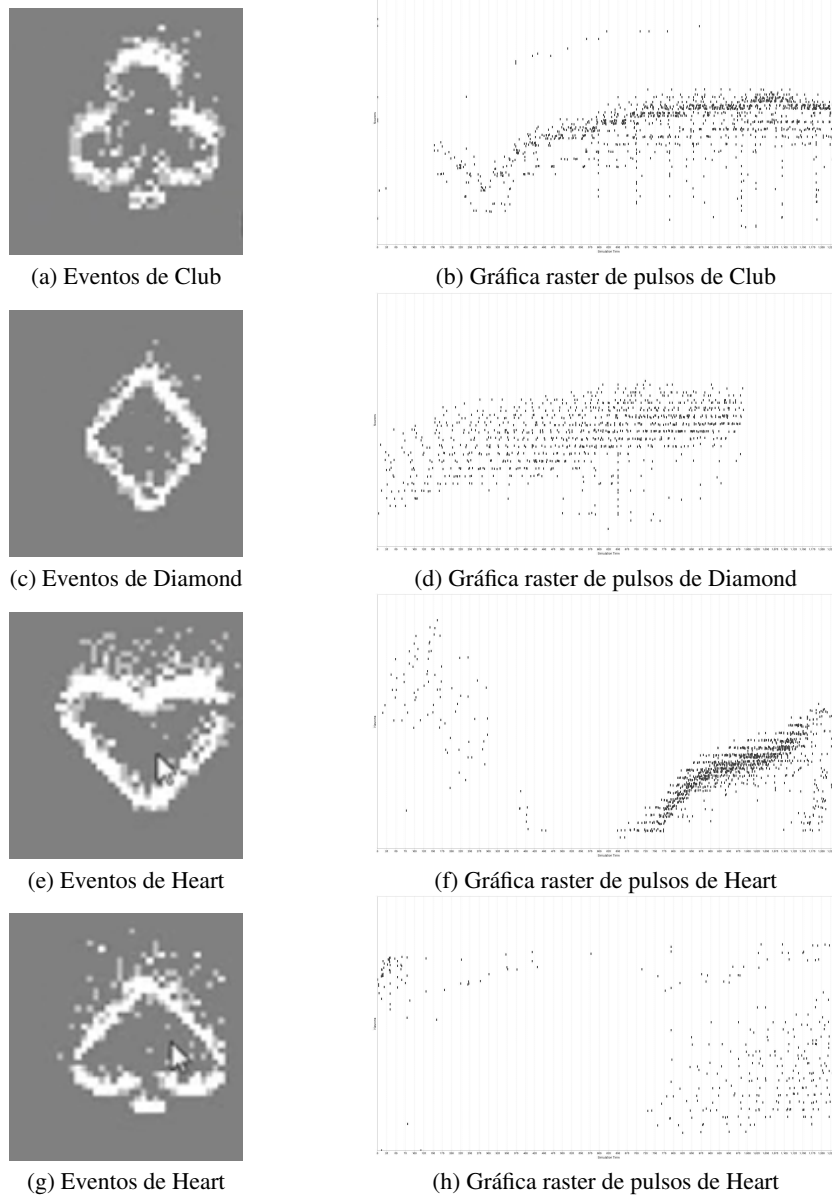
donde  $\gamma \in [0, 1]$  define la tasa de disparo,  $N$  es el número de neuronas en la SNN y  $W$  es la matriz de pesos sinápticos e  $I_i^{ext}$  representa un estímulo externo (en este trabajo  $I_i^{ext} = 0$ ). Cuando el potencial sináptico  $V_i[k]$  alcanza un umbral  $\theta$  dado, se produce un pulso (este es determinado en el término  $Z_i[k]$  de la ecuación(4) y la  $i$ -ésima neurona se restablece mediante el término  $(1 - Z_i[k])$  en la ecuación (3).

### 4. Experimentos y resultados

Para analizar el rendimiento de una LSM que implementa el modelo de neurona pulsante BMS como unidad de procesamiento de su reservorio, se utilizó la base de datos Poker-DVS<sup>4</sup> [28].

Esta base de datos contiene eventos capturados al pasar rápidamente las cartas de una baraja frente a un Sensor de Visión Dinámica (DVS por sus siglas en inglés -Dynamic Vision Sensor-), la tarea a llevar a cabo es la clasificación de los diferentes palos de la baraja.

<sup>4</sup> Disponible en: <http://www2.imse-cnm.csic.es/caviar/POKERDVS.html>



**Fig. 3.** Palos de la baraja representados por imágenes de flujos de eventos (columna izquierda) y gráficas raster de los trenes de pulsos (columna derecha).

La base de datos Poker-DVS contiene 4 clases (Club, Diamond, Heart y Spade) y un total de 131 patrones caracterizados por flujos de eventos de  $32 \times 32$  píxeles. En la Tabla 1 se muestran los nombres de las clases, el identificador de clase y la cantidad de patrones en cada una de ellas.

**Tabla 2.** Valores usados en los parámetros de la LSM.

<b>Parámetros LSM</b>	
Neuronas en capa entrada	1024
Dimensión reservorio	$10 \times 10 \times 15$
Neuronas excitatorias (%) [36]	80 %
Neuronas inhibitorias (%) [36]	20 %
Porcentaje conexión c. entrada-reservorio	0,01 %
Rango de pesos c. entrada-reservorio	(0,001, 1)
$\lambda$	1
Prob. conexión excitatoria-excitatoria [36]	0,3
Prob. conexión excitatoria-inhibitoria [36]	0,2
Prob. conexión inhibitoria-excitatoria [36]	0,4
Prob. conexión inhibitoria-inhibitoria [36]	0,1
Rango de pesos sinápticos reservorio	(0,001, 1)
Tiempo de simulación	1250 [ms]
<b>Parámetros neurona pulsante BMS</b>	
$\gamma$ [26, 2]	0,5
$\theta$ [26, 2]	1

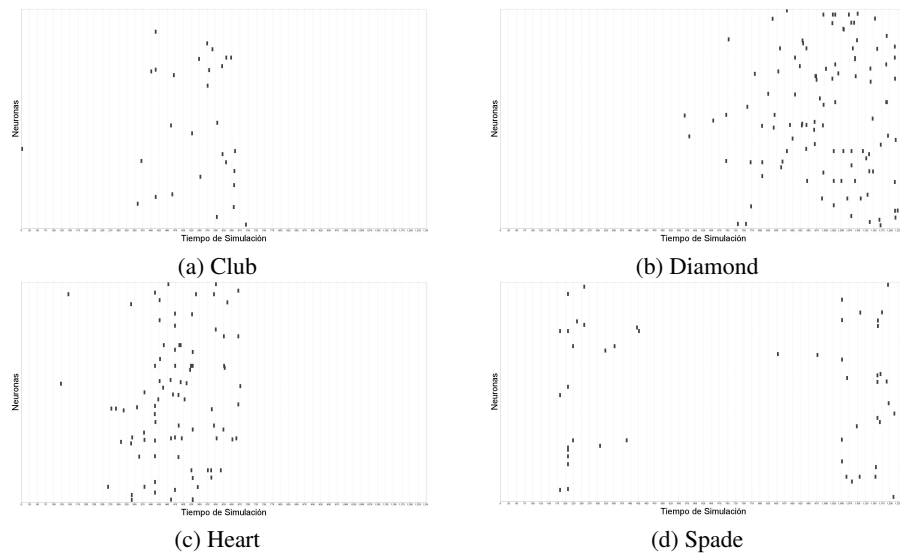
Los flujos de eventos representando cada patrón son interpretados como trenes de pulsos (uno por cada píxel); estos serán las señales de la capa de entrada de la LSM.

En la Figura 3, la columna izquierda muestra imágenes de flujos de eventos capturados por la DVS para cada uno de los diferentes símbolos de la baraja (Club -Figura 3a-, Diamond -Figura 3c-, Heart -Figura 3e- y Spade -Figura 3g-) y la columna derecha muestra la correspondiente gráfica raster de los trenes de pulsos de cada diferente símbolo de la baraja (Club -Figura 3b-, Diamond -Figura 3d-, Heart -Figura 3f- y Spade -Figura 3h-).

Para la experimentación, además de usar la base de datos Poker-DVS con todas sus clases, se generaron a partir de ésta, conjuntos extra con base a todas las combinaciones de pares de clases; dando un total de 7 conjuntos de experimentación.

Debido a la naturaleza aleatoria en la creación del reservorio en una implementación estándar de LSM, cada conjunto fue sometido a 36 procesos de clasificación, donde para cada uno se usó Validación Cruzada de  $K$  Pliegues (KFCV por sus siglas en inglés -K-Folds Cross Validation-) con  $K = 10$  [15] generando en cada etapa individual de la validación un nuevo reservorio y calculando la exactitud para el conjunto de prueba obtenido por la LSM.

La Tabla 2 muestra la lista de valores usados para los parámetros de la LSM, el modelo BMS y la ecuación de conectividad (Ec. 3 en [36]), donde los valores de algunos parámetros fueron extraídos de trabajos del estado del arte (referencia a la derecha del nombre del parámetro) y para otros fueron obtenidos empíricamente o de acuerdo a las características de la base de datos Poker-DVS.



**Fig. 4.** Gráficas de raster de la actividad del reservorio para patrones de las diferentes clases de Poker-DVS (Esta actividad neuronal es transformada a vectores de estado para su clasificación).

La Figura 4, muestra la actividad generada por el reservorio en respuesta a patrones de las diferentes clases en la base de datos Poker-DVS. En la Tabla 3 se muestran para cada conjunto experimental de datos el promedio de exactitud de clasificación del conjunto de prueba obtenido al aplicar KFCV en el respectivo proceso de clasificación y también, en la parte inferior, un resumen estadístico por cada base de datos.

La Figura 5 muestra una gráfica de cajas y bigotes a modo de resumen visual de los estadísticos. A partir de los resultados obtenidos, se puede observar que implementaciones estándar de LSMs usando el modelo de neurona pulsante BMS en su reservorio, en general, tienen poca dispersión (ninguno sobrepasa el 0.028) pese a la naturaleza aleatoria de la creación de su reservorio.

Además, el desempeño de la LSM para problemas de clasificación binaria derivados de la base de datos Poker-DVS es para la mayoría de los casos bueno sobrepasando en promedio el 85 % de exactitud; solo dos subconjuntos que involucran la clase Spade rondan en promedio el 80 %.

El comportamiento de la LSM al trabajar con todas las clases de la base de datos Poker-DVS, resultó en un 76 % de exactitud en promedio, la más baja en todas las pruebas; lo cual podrá deberse, además del incremento de clases, al conflicto que se puede vislumbrar de la clase Spade con Club y Diamond en los resultados de los subconjuntos biclase.

El desempeño de las LSMs con reservorio basado en el modelo BMS para el caso de la prueba de clasificación usando las 4 clases se comparó contra diversos trabajos reportados en el estado del arte; a continuación se describen brevemente dichas investigaciones.

En [34] se propone una SNN de 4 capas, la primera capa convierte las señales de entrada en trenes de pulsos, la segunda capa está conformada por neuronas pulsantes de

**Tabla 3.** Promedios de exactitud de clasificación para los experimentos realizados.

# Exp	C-D	C-H	C-S	D-H	D-S	H-S	Todas
1	0.8900	0.9600	0.7985	0.9500	0.8000	0.9123	0.7449
2	0.8857	0.9675	0.7970	0.9750	0.8162	0.9123	0.7529
3	0.8857	0.9400	0.8485	0.9500	0.8438	0.9046	0.7359
4	0.8757	0.9600	0.7909	0.9583	0.8648	0.9200	0.7315
5	0.8800	0.9475	0.8076	0.9583	0.8295	0.9123	0.7442
6	0.9186	0.9600	0.8152	0.9583	0.8648	0.9123	0.7402
7	0.9000	0.9400	0.7742	0.9667	0.8505	0.9323	0.7909
8	0.8900	0.9550	0.7833	0.9750	0.8086	0.8800	0.7659
9	0.8943	0.9550	0.8242	0.9500	0.7876	0.9323	0.7569
10	0.9086	0.9400	0.7909	0.9333	0.8019	0.9000	0.7406
11	0.9000	0.9800	0.7894	0.9583	0.8019	0.9323	0.7902
12	0.9000	0.9475	0.8045	0.9583	0.7867	0.9446	0.7783
13	0.8800	0.9475	0.8485	0.9417	0.8219	0.9000	0.7322
14	0.9000	0.9475	0.7667	0.9500	0.8457	0.9169	0.7826
15	0.9043	0.9600	0.7727	0.9583	0.7943	0.9123	0.7692
16	0.9186	0.9600	0.7500	0.9500	0.7810	0.9123	0.7572
17	0.9000	0.9400	0.8303	0.9583	0.8724	0.9000	0.7366
18	0.9043	0.9800	0.8076	0.9583	0.7790	0.8923	0.7783
19	0.9000	0.9400	0.7333	0.9500	0.7933	0.9123	0.7783
20	0.9143	0.9600	0.8061	0.9583	0.8362	0.9123	0.7569
21	0.8943	0.9400	0.7894	0.9667	0.8019	0.9246	0.7620
22	0.8857	0.9475	0.8136	0.9583	0.8295	0.8923	0.7529
23	0.9043	0.9475	0.7970	0.9500	0.8171	0.9400	0.7616
24	0.8986	0.9675	0.8242	0.9667	0.8438	0.9200	0.8033
25	0.8857	0.9400	0.7894	0.9500	0.8524	0.8923	0.7909
26	0.9000	0.9675	0.8000	0.9750	0.8362	0.9000	0.7656
27	0.8714	0.9400	0.8061	0.9417	0.8238	0.9200	0.7580
28	0.8900	0.9475	0.8061	0.9583	0.7733	0.9323	0.7529
29	0.9143	0.9475	0.7894	0.9500	0.8305	0.8646	0.7902
30	0.8943	0.9350	0.8242	0.9500	0.8295	0.9000	0.7656
31	0.8943	0.9475	0.8167	0.9500	0.7590	0.9000	0.7779
32	0.9143	0.9475	0.8076	0.9583	0.8295	0.9246	0.7493
33	0.9043	0.9750	0.8061	0.9583	0.8457	0.9200	0.7946
34	0.8514	0.9475	0.8318	0.9417	0.8010	0.9123	0.8029
35	0.8757	0.9600	0.7818	0.9417	0.8305	0.9246	0.7822
36	0.8717	0.9400	0.7818	0.9750	0.8448	0.8800	0.7572
<b>Min</b>	0.8514	0.9350	0.7333	0.9333	0.7590	0.8646	0.7315
<b>Max</b>	0.9186	0.9800	0.8485	0.9750	0.8724	0.9446	0.8033
<b>Med</b>	0.8964	0.9475	0.8023	0.9583	0.8267	0.9123	0.7618
$\mu$	0.8944	0.9524	0.8001	0.9558	0.8202	0.9112	0.7647
$\sigma$	0.0147	0.0121	0.0242	0.0101	0.0280	0.0174	0.0204

I&F las cuales se encargan de extraer las características de los trenes de pulsos que son enviadas a la siguiente capa, la tercer capa se encarga de la etapa de aprendizaje donde cada clase está representada por poblaciones de neuronas y finalmente, la cuarta capa es la capa de decisión donde se combinan las salidas de la capa anterior para realizar la claisificación; la exactitud reportada es de 96,0 %.

Los eventos espacio-temporales son caracterizados en [21] mediante descriptores de Histogramas de Eventos Orientados y Patrón Binario Local con los cuales reportan una exactitud del 98 %.

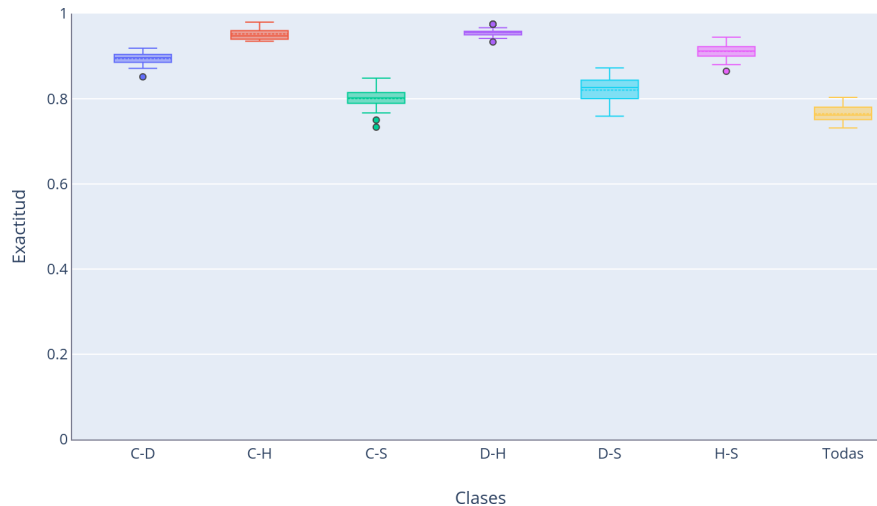


Fig. 5. Boxplot de los resultados.

En [29] se propone una metodología de 3 módulos, el primero se encarga de extraer características binarias de los eventos espacio-temporales, el segundo es un detector de movimiento y el tercero es un ensamble de modelos de aprendizaje máquina basados en una variante de clasificadores Bayesianos semi-ingenuos; se reporta una exactitud de 97 %. Una SNN convolucional se utiliza en [13] reportando una exactitud del 90 %.

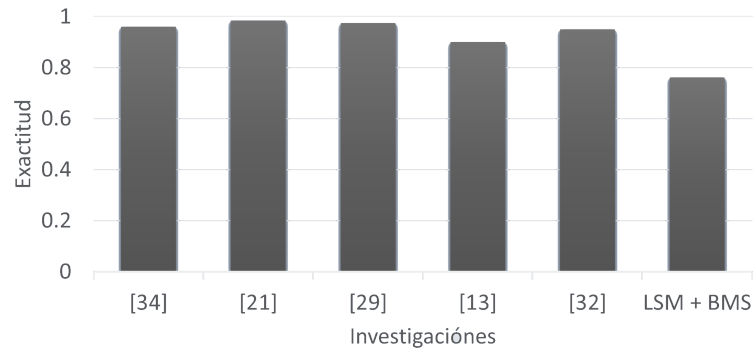
Una regla de aprendizaje supervisado para SNNs es propuesta en [32], la cual usa la actividad generada por una SNN con topología arbitraria para construir histogramas y efectuar un proceso de aprendizaje basado en descenso de gradiente estocástico; se reporta una exactitud entre 94 % y 100 %.

La Figura 6 muestra las exactitudes reportadas en el estado del arte y la obtenida mediante la configuración propuesta en esta investigación (LSM+BMS), se puede observar que las metodologías mejoradas o definidas adhoc logran mejores resultados de exactitud; sin embargo, la configuración propuesta de LSM permite explorar alternativas de mejora que serán comentadas en la sección 5.

## 5. Conclusiones y trabajo futuro

Hoy en día la cantidad y naturaleza de la información nos llevan cada vez más a utilizar instancias de pruebas espacio-temporales con la exigencia de resolver las cosas al instante. Con el presente trabajo vislumbramos un panorama favorecedor en cuanto al rendimiento de LSMs ya que no requieren de efectuar un proceso de calibración de pesos o de definir un patrón de conexiones sinápticas explícitos; los cuales suelen ser necesarios en la mayoría de implementaciones de ANNs, incluyendo las SNNs, que suelen ser procesos que requieren de expertos humanos, requerir de tiempo y recursos de procesamiento.

El modelo BMS es una neurona pulsante de bajo costo computacional que ha sido usado en trabajos con recursos de hardware limitado (como en aplicaciones robóticas),



**Fig. 6.** Comparación de exactitudes obtenidas para Poker-DVS.

estas características hacen que el estudio de su implementación en LSMs sea atractivo para implementaciones en hardware, robótica o simulaciones en equipos de cómputo de recursos limitados.

Los resultados obtenidos en las pruebas a pares de clases de Poker-DVS muestran valores aceptables superando 85 % de exactitud en la mayoría de las instancias con poca dispersión en la repetición de pruebas a pesar de la aleatoriedad en la creación de los reservorios.

Sin embargo, los resultados obtenidos al resolver el problema de clasificación incluyendo las cuatro clases quedan por debajo de los reportados en el estado del arte; reportando un 76 % de exactitud.

Lo anterior, sirve de motivación para explorar alternativas para mejorar los resultados obtenidos; esto mediante la exploración de formas de representación de las respuestas descriptivas del reservorio, estructura de la topología del reservorio, calibración de los pesos exógenos y/o endógenos del reservorio, etc.

Aunado con las ventajas descritas previamente, este trabajo sirve como base para seguir explorando nuevas alternativas de bajo costo computacional en el área de las LSMs, permitiendo entender la naturaleza del modelo para así fortalecer sus debilidades.

En un futuro se propone trabajar en la calibración de los pesos y las conexiones sinápticas dentro del líquido mediante técnicas de optimización evolutivas o bionspiradas, con el objetivo de lograr resultados competitivos con el estado del arte. Además de buscar formas de mejorar las propiedades de las LSM que favorezcan a su desempeño y caracterización.

## Referencias

1. Cessac, B.: A discrete time neural network model with spiking neurons. *Journal of Mathematical Biology*, vol. 56, no. 3, pp. 311–345 (2008) doi: 10.1007/s00285-007-0117-3
2. Espinal, A., Rostro-Gonzalez, H., Carpio, M., Guerra-Hernandez, E. I., Ornelas-Rodriguez, M., Puga-Soberanes, H., Sotelo-Figueroa, M. A., Melin, P.: Quadrupedal robot locomotion: A biologically inspired approach and its hardware implementation. *Computational Intelligence and Neuroscience*, vol. 2016 (2016) doi: 10.1155/2016/5615618

3. Ferreira, A. A., Ludermir, T. B., De Aquino, R. R.: An approach to reservoir computing design and training. *Expert Systems with Applications*, vol. 40, no. 10, pp. 4172–4182 (2013) doi: 10.1016/j.eswa.2013.01.029
4. Friedman, M., Kandel, A.: *Introduction to pattern recognition: Statistical, structural, neural and fuzzy logic approaches*, vol. 32 (1999)
5. Gerstner, W.: Population dynamics of spiking neurons: Fast transients, asynchronous states, and locking. *Neural Computation*, vol. 12, no. 1, pp. 43–89 (2000) doi: 10.1162/089976600300015899
6. Gerstner, W., Kistler, W. M.: *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge University Press (2002)
7. Grzyb, B. J., Chinellato, E., Wojcik, G. M., Kaminski, W. A.: Which model to use for the liquid state machine? In: *Proceedings of the International Joint Conference on Neural Networks*, pp. 1018–1024 (2009) doi: 10.1109/ijcnn.2009.5178822
8. Hazan, H., Manevitz, L. M.: Topological constraints and robustness in liquid state machines. *Expert Systems with Applications*, vol. 39, no. 2, pp. 1597–1606 (2012) doi: 10.1016/j.eswa.2011.06.052
9. Hourdakis, E., Trahanias, P.: Use of the separation property to derive liquid state machines with enhanced classification performance. *Neurocomputing*, vol. 107, pp. 40–48 (2013) doi: 10.1016/j.neucom.2012.07.032
10. Jaeger, H.: The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report, vol. 148, no. 34, pp. 13 (2001)
11. Ju, H., Xu, J. X., Chong, E., vanDongen, A. M.: Effects of synaptic connectivity on liquid state machine performance. *Neural Networks*, vol. 38, pp. 39–51 (2013) doi: 10.1016/j.neunet.2012.11.003
12. Ju, H., Xu, J. X., vanDongen, A. M.: Classification of musical styles using liquid state machines. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7 (2010) doi: 10.1109/ijcnn.2010.5596470
13. Kaiser, J., Zimmerer, D., Tieck, J. C. V., Ulbrich, S., Roennau, A., Dillmann, R.: Spiking convolutional deep belief networks. In: *International Conference on Artificial Neural Networks*, pp. 3–11 (2017) doi: 10.1007/978-3-319-68612-7\_1
14. Kamaruzaman, F., Shafie, A. A., Mustafah, Y. M.: Coincidence detection using spiking neurons with application to face recognition. *Journal of Applied Mathematics*, vol. 2015 (2015) doi: 10.1155/2015/534198
15. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2, pp. 1137–1145 (1995)
16. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, vol. 3, no. 3, pp. 127–149 (2009) doi: 10.1016/j.cosrev.2009.03.005
17. Lukoševičius, M., Jaeger, H., Schrauwen, B.: Reservoir computing trends. *KI-Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371 (2012) doi: 10.1007/s13218-012-0204-5
18. Maass, W.: Networks of spiking neurons: The third generation of neural network models. *Neural networks*, vol. 10, no. 9, pp. 1659–1671 (1997) doi: 10.1016/s0893-6080(97)00011-7
19. Maass, W.: Liquid state machines: Motivation, theory, and applications. *Computability in context: Computation and logic in the real world*, pp. 275–296 (2011) doi: 10.1142/9781848162778\_0008
20. Maass, W., Natschläger, T., Markram, H.: 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) doi: 10.1162/089976602760407955

21. Negri, P.: Shapes characterization on address event representation using histograms of oriented events and an extended LBP approach (2018) doi: 10.48550/ARXIV.1802.03327
22. Norton, D., Ventura, D.: Improving liquid state machines through iterative refinement of the reservoir. *Neurocomputing*, vol. 73, no. 16–18, pp. 2893–2904 (2010) doi: 10.1016/j.neucom.2010.08.005
23. Paugam-Moisy, H., Martinez, R., Bengio, S.: Delay learning and polychronization for reservoir computing. *Neurocomputing*, vol. 71, no. 7–9, pp. 1143–1158 (2008) doi: 10.1016/j.neucom.2007.12.027
24. Rhéaume, F., Grenier, D., Bossé, É.: Multistate combination approaches for liquid state machine in supervised spatiotemporal pattern classification. *Neurocomputing*, vol. 74, no. 17, pp. 2842–2851 (2011)
25. Rosselló, J. L., Alomar, M. L., Morro, A., Oliver, A., Canals, V.: High-density liquid-state machine circuitry for time-series forecasting. *International Journal of Neural Systems*, vol. 26, no. 5 (2016) doi: 10.1142/s0129065715500367
26. Rostro-Gonzalez, H., Garreau, G., Andreou, A., Georgiou, J., Barron-Zambrano, J. H., Torres-Huitzil, C.: An FPGA-based approach for parameter estimation in spiking neural networks. In: *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 2897–2900 (2012) doi: 10.1109/iscas.2012.6271920
27. Schrauwen, B., D’Haene, M., Verstraeten, D., Van Campenhout, J.: Compact hardware liquid state machines on fpga for real-time speech recognition. *Neural networks*, vol. 21, no. 2-3, pp. 511–523 (2008) doi: 10.1016/j.neunet.2007.12.009
28. Serrano-Gotarredona, T., Linares-Barranco, B.: Poker-DVS and MNIST-DVS. Their history, how they were made, and other details. *Frontiers in Neuroscience*, vol. 9, pp. 481 (2015) doi: 10.3389/fnins.2015.00481
29. Shi, C., Li, J., Wang, Y., Luo, G.: Exploiting lightweight statistical learning for event-based vision processing. vol. 6, pp. 19396–19406 (2018) doi: 10.1109/access.2018.2823260
30. Soula, H., Beslon, G., Mazet, O.: Spontaneous dynamics of asymmetric random recurrent spiking neural networks. *Neural Computation*, vol. 18, no. 1, pp. 60–79 (2006) doi: 10.48550/ARXIV.CS/0411052
31. Steil, J. J.: Backpropagation-decorrelation: Online recurrent learning with  $O(N)$  complexity. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*. vol. 2, pp. 843–848 (2004) doi: 10.1109/ijcnn.2004.1380039
32. Stomatias, E., Soto, M., Serrano-Gotarredona, T., Linares-Barranco, B.: An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. *Frontiers in Neuroscience*, vol. 11, pp. 350 (2017) doi: 10.3389/fnins.2017.00350
33. Verstraeten, D., Schrauwen, B., d’Haene, M., Stroobandt, D.: An experimental unification of reservoir computing methods. *Neural Networks*, vol. 20, no. 3, pp. 391–403 (2007) doi: 10.1016/j.neunet.2007.04.003
34. Wang, T., Shi, C., Zhou, X., Lin, Y., He, J., Gan, P., Li, P., Wang, Y., Liu, L., Wu, N., Luo, G.: CompSNN: A lightweight spiking neural network based on spatiotemporally compressive spike features. *Neurocomputing*, vol. 425, pp. 96–106 (2021)
35. Zhang, Y., Li, P., Jin, Y., Choe, Y.: 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) doi: 10.1109/tnnls.2015.2388544
36. Zhou, Y., Jin, Y., Ding, J.: Evolutionary optimization of liquid state machines for robust learning. In: *International Symposium on Neural Networks*, pp. 389–398 (2019) doi: 10.1007/978-3-030-22796-8\_41

