frontiers in Neuroscience

ORIGINAL RESEARCH
published: 08 November 2019
doi: 10.3389/fnins.2019.01085

Bio-Inspired Evolutionary Model of Spiking Neural Networks in Ionic Liquid Space

*Ensieh Iranmehr[1], Saeed Bagheri Shouraki[1], Mohammad Mahdi Faraji[1], Nasim Bagheri[1] and Bernabe Linares-Barranco[2]**

[1] Artificial Creatures Laboratory, Electrical Engineering Department, Sharif University of Technology, Tehran, Iran, [2] Instituto de Microelectrońica de Sevilla (CSIC and Univ. de Sevilla), Seville, Spain

One of the biggest struggles while working with artificial neural networks is being able to come up with models which closely match biological observations. Biological neural networks seem to capable of creating and pruning dendritic spines, leading to synapses being changed, which results in higher learning capability. The latter forms the basis of the present study in which a new ionic model for reservoir-like networks, consisting of spiking neurons, is introduced. High plasticity of this model makes learning possible with a fewer number of neurons. In order to study the effect of the applied stimulus in an ionic liquid space through time, a diffusion operator is used which somehow compensates for the separation between spatial and temporal coding in spiking neural networks and therefore, makes the mentioned model suitable for spatiotemporal patterns. Inspired by partial structural changes in the human brain over the years, the proposed model evolves during the learning process. The effect of topological evolution on the proposed model's performance for some classification problems is studied in this paper. Several datasets have been used to evaluate the performance of the proposed model compared to the original LSM. Classification results via separation and accuracy values have shown that the proposed ionic liquid outperforms the original LSM.

Keywords: spiking neural network, ionic liquid space, genetic algorithm, evolutionary model, synaptic plasticity, intrinsic plasticity, structural plasticity, liquid state machine

OPEN ACCESS

Edited by:
Arindam Basu,
Nanyang Technological University,
Singapore

Reviewed by:
Subhrajit Roy,
IBM Research, Australia
Priyadarshini Panda,
Yale University, United States

****Correspondence:***
Bernabe Linares-Barranco
bernabe@imse-cnm.csic.es

Specialty section:
This article was submitted to
Neuromorphic Engineering,
a section of the journal
Frontiers in Neuroscience

Received: *08 June 2019*
Accepted: *25 September 2019*
Published: *08 November 2019*

Citation:
Iranmehr E, Shouraki SB, Faraji MM, Bagheri N and Linares-Barranco B (2019) Bio-Inspired Evolutionary Model of Spiking Neural Networks in Ionic Liquid Space. Front. Neurosci. 13:1085. doi: 10.3389/fnins.2019.01085

1. INTRODUCTION

Artificial Intelligence, also known as AI, is intelligence demonstrated by machines. Also an area of computer science, AI is one of the most developed scientific fields which has brought so much attention to itself over the past few years. Despite the remarkable development of AI systems over the past few years, designing a system which holds the capabilities of the human brain seems rather hard to achieve. Many AI-based computational systems have been developed so far; yet, none of them can compare to the processing mechanism of the human brain. Designing intelligent systems with the ability to carry out computations similar to the way the human brain does, is studied in the fields of neural networks and fuzzy systems.

Since the most common data obtained from the brain's response to external stimuli are spatiotemporal data, the brain is considered to be a spatiotemporal-data-processing machine. Despite the development of precise brain models (Markram, 2006; Toga et al., 2006; Izhikevich and Edelman, 2008), the available models cannot be used for machine learning and or recognizing spatiotemporal patterns, since they are designed to model the brain's structure as well as its function and, therefore, are unable to mine and learn from the brain data. Creating an integrated computational structure in order to process spatiotemporal data is therefore, one of the main challenges of brain simulation.

Spiking neural networks seem suitable for creating such structures since they are designed to process spiking information which creates spatiotemporal data. In other words, considering the fact that there is information at the exact time a spike appears, spiking neurons send out the information via spikes instead of firing rates (Haykin, 1998). Research shows that temporal coding is used to represent and process the information in brain cortex (Ikegaya et al., 2004; Butts et al., 2007). Besides, spiking neurons are more dynamic since they use time domain (Rolls and Tovee, 1995). Hence, in this paper, spiking neural networks are used to create a computational structure for both comprehending and learning spatiotemporal data.

A large amount of data is processed in the neocortex via stereotypical neural micro circuitry. Therefore, the introduced model is based on the idea of Dynamic Reservoir Networks (DRN) (Schrauwen et al., 2007), i.e., Liquid State Machines (LSM) and Echo State Networks (ESN) (Jaeger, 2001; Maass et al., 2002; Natschläger et al., 2002), which are examples of recursive neural networks with a random structure. The natural dynamic of these networks makes them suitable for online processing of time variant inputs. The mentioned reservoir networks receive the input streams and transform them into non-linear patterns in higher dimensions and show a fading memory of the recent inputs. The reservoir acts as a filter. The state of the reservoir or state vector is then used as inputs for the readout layer. The readout layer of the mentioned networks is used for online processing of the time series. This layer needs to be trained using some simple algorithms such as linear regression methods. The dynamic of the reservoirs directly affects the performance of these networks.

A random generated liquid may not act as a useful filter. Researchers are therefore forced to generate many random liquids until a useful filter is found. On the other hand, there are a lot of studies on improving the liquid in order to find a more useful filter (Lazar et al., 2007; Norton and Ventura, 2010; Rhéaume et al., 2011; Hazan and Manevitz, 2012; Notley and Gruning, 2012; Wojcik, 2012; Hourdakis and Trahanias, 2013; Ju et al., 2013; Sillin et al., 2013; Xue et al., 2013; Roy and Basu, 2016). These studies focus on enhancing the liquid performance by applying different learning rules or neuron models to the liquid. For example Norton and Ventura (2010), proposed a learning rule for training the liquid of LSM in order to construct a suitable liquid, Roy and Basu (2016) proposed an online structural plasticity rule to generate better liquids, Hourdakis and Trahanias (2013) used a measure in an evolutionary framework to generate liquid with appropriate parameters and Wojcik (2012) used Hodgkin and Huxley neurons instead of LIF neurons.

This paper proposes an ionic model of reservoir-like networks in which spiking neurons are located. Connections between spiking neurons are provided by ionic density. One of the exciting aspects of this model is that the link to ion fields invokes an abstraction of biologically plausible processes which may set a foundation for possible future research into neural network dynamics, integrating both spiking and field-based computation in biology. Since all neurons in the ionic liquid can connect to each other, the introduced model can be considered a recursive network. In an ionic space, ionic diffusion is one of the most important factors which affects the connection between neurons. Several algorithms have been developed based on diffusion, including Active Learning Fuzzy Modeling Method (ALM) (Shouraki and Honda, 1997, 1999; Murakami and Honda, 2007). In ALM, features are extracted using an operator known as Ink Drop Spread (IDS) (Murakami and Honda, 2007) which is mainly based on diffusion.

Both synaptic and non-synaptic plasticity make the brain capable of learning. A synapse's ability to change in strength over time is known as synaptic plasticity while intrinsic plasticity involves changes in the electrical properties within a single neuron. The contribution between these two different types of plasticity and how it results in the dynamic and the structure of the cortical network, leads to the development of cortical information processing and coding theory. Hebb's rule (Hebb, 1949) is a learning method based on biological observations which form the basis of "memory" and "learning." So far, many learning methods (Bienenstock et al., 1982; Oja, 1982; Song et al., 2000; Bi and Poo, 2001; Panchev and Wermter, 2004; Ponulak, 2005; Ponulak and Kasiński, 2010; Mohemmed et al., 2012) have been developed based on Hebb's rule such as the well-known Spike-timing-dependent plasticity (STDP).

Also, Lazar et al. (2007) shows how the computational capability of LSM can be improved by combining STDP and Intrinsic Plasticity (IP). Hence, in the introduced model, intrinsic plasticity has been used to maintain the homeostasis of neuronal performance (Desai et al., 1999; Daoudal and Debanne, 2003). Homeostasis is in fact a biological term describing the stability of the environment in which cells exist and is defined as the will to create an almost-stable balance between dependent elements. A lot of learning rules for training networks of spiking neurons are proposed through various forms of synaptic plasticity (Ponulak and Kasiński, 2010; Kuhlmann et al., 2013; Sporea and Grüning, 2013; Gardner et al., 2015). Most of them have explored weight plasticity which refers to modifying the synaptic strengths. Diehl and Cook (2016) have shown that inclusion of synaptic plasticity within reservoirs can help in learning and inferring relationships between inputs. Also, Panda and Roy (2017) have shown that inclusion of Hebbian and non-Hebbian plasticity at the same time helps in the learning of stable contextual dependencies between temporal sequences. There is another form of plasticity, known as structural plasticity, which is explored in a few works (Roy et al., 2016; Roy and Basu, 2017). This form of plasticity revolves around training a network through formation and elimination of synapses. Recently, structural plasticity has been used to train feed forward neural networks (George et al., 2015b), recurrent neural networks (George et al., 2015a), and reservoir networks (Roy and Basu, 2016). The proposed model therefore uses synaptic plasticity caused by dendritic spines (known as dendritic plasticity) to improve learning (Roberts et al., 2010; Tschida and Mooney, 2012). Dendritic spines increase the learning ability in humans by creating and removing synapses. The disability to learn in children suffering from Down Syndrome and/or Fragile X Syndrome is in fact due to the weakness of dendritic spines (Wang et al., 2012). The proposed model therefore tries to create new synapses and removes the pre-existing ones by considering the movement of dendrites, which somehow leads to improving the ability to learn with fewer neurons.

This paper, just like other works, focuses on improving the liquid through structural plasticity (Roy and Basu, 2016) and intrinsic plasticity (Lazar et al., 2007) but with a major difference. Training in the proposed model is performed through ionic liquid. Ionic liquid makes structural plasticity as well as weight plasticity possible, using diffusion, which induces online learning due to simple computations. Both of these types of plasticity are motivated from biological observations.

The proposed model seems to be a better way to model reservoirs because of the following two advantages: (1) The proposed connectivity, which is provided by diffusion in the ionic space, is one of the few dynamic connectivity models that does not involve complex computations. (2) Since the connectivity information is transmitted locally through the ion field, none of the neurons require any knowledge of the global state to update their connections.

The human brain has evolved over more than five million years. Therefore, proposing precise models of the brain requires modeling evolution basics in nature. Based on biological findings, the human brain has evolved over time and therefore, we have tried improving the efficiency of the introduced model using evolutionary algorithms (Simon, 2013). Evolutionary algorithms such as the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have several applications in machine learning (Faraji et al., 2017; Iranmehr et al., 2017) and neural networks (Gao et al., 2019). In this paper, the topology of the introduced model is optimized using a genetic algorithm as well as some experiments. For each problem, the topology of the introduced model can be optimized so that it would have the best performance. The term "optimizing the topology" in this paper can refer to optimizing the total number of network's neurons, number of dendrites per neuron and the location of dendrites and axons' terminals. In other words, in this paper, in order to solve a particular problem, we try to optimize the topology of the introduced model so that the efficiency increases. Since the proposed model is suitable for classifying spatiotemporal data, its performance when classifying some time-varying datasets has been analyzed.

Concisely, this paper proposes a new model of reservoir networks where the neural dynamics are modeled by the ionic diffusion formulae and also focuses on topologic optimization of the introduced model using evolutionary algorithms in order to improve its performance. The proposed model which is suitable for processing spatiotemporal or time varying data has been used in classifying different datasets such as N-MNIST (Orchard et al., 2015), TIMIT (Garofolo et al., 1993), and FSDD (Jackson, 2016). In section 2, we discuss the introduced model and provide information about the structure of the proposed model, the ionic space in which the neurons are located, and ionic diffusion which leads to neuronal connections being made. Section 3 studies the performance of the proposed model using different metrics such as separation, approximation and generalization. Section 4 focuses on optimizing the topology of the introduced model. Section 5 includes the results obtained from evaluating the classification performance of the proposed neural network on the N-MNIST dataset. Moreover, the results obtained from comparing our proposed model with some other works are included in section 5. Finally, we conclude the paper in section 6.

2. THE PROPOSED SPIKING NEURAL NETWORK IN IONIC LIQUID SPACE

Here, a computational model based on spiking neurons is introduced, considering biological observations as well as reservoir networks. Since the electrochemical transition of neurons occurs via ions, it is assumed that the neurons are placed in an ionic liquid space which we refer to as ILS. The density of ions in the ionic liquid space strengthens some of the connections between neurons and allows neurons to connect to each other via the ionic liquid. In conventional neural networks, each synapse is modeled by a coefficient. However, in the introduced model, the synapses are modeled using ionic diffusion in ILS. Since the main factor in diffusion is time delay, it can be said that in the introduced model, synapses are somehow modeled dynamically. In this section, after explaining the structure of the introduced computational model, we describe the network, considering both intrinsic and synaptic plasticity.

2.1. Structure of the Proposed Computational Model

What leads neurons to respond to stimuli and transmit spikes is the imbalanced distribution of ions in the intracellular and extracellular space of nerve cell membranes. Therefore, in the introduced model, it is assumed that spiking neurons are placed in an ionic liquid space. The mentioned liquid space can be either two-dimensional or three-dimensional based on the defined problem. To simplify, the ionic liquid space is considered to be two-dimensional in the present paper. The ion density gradient, which is due to the presence of stimulus in this space, leads to diffusion since ions move from a point with higher concentration to a point with lower concentration. The difference in ion density leads to voltage difference. Furthermore, ions diffuse through cell membrane at a molecular level. Therefore, in order to model the diffusion process, we first need to quantize the ILS to a particular number of bins and then, assign a voltage level to the ion density of each bin. Diffusion in the ILS causes the voltage level of each bin to change over time. **Figure 1** shows an example of the computational model introduced in this paper. The $2D$ ionic liquid space is quantized to 100×100 bins. In this ILS, there are $25 \times 25 = 625$ neurons which are depicted in blue. Each neuron consists of two dendrites and one axon which are depicted in magenta. The terminals of axons and dendrites are depicted with red stars and green squares, respectively. The terminals of axons and dendrites are each placed in a bin so that the dendrites can receive the bin voltage, transmit it to the soma and also, to enter

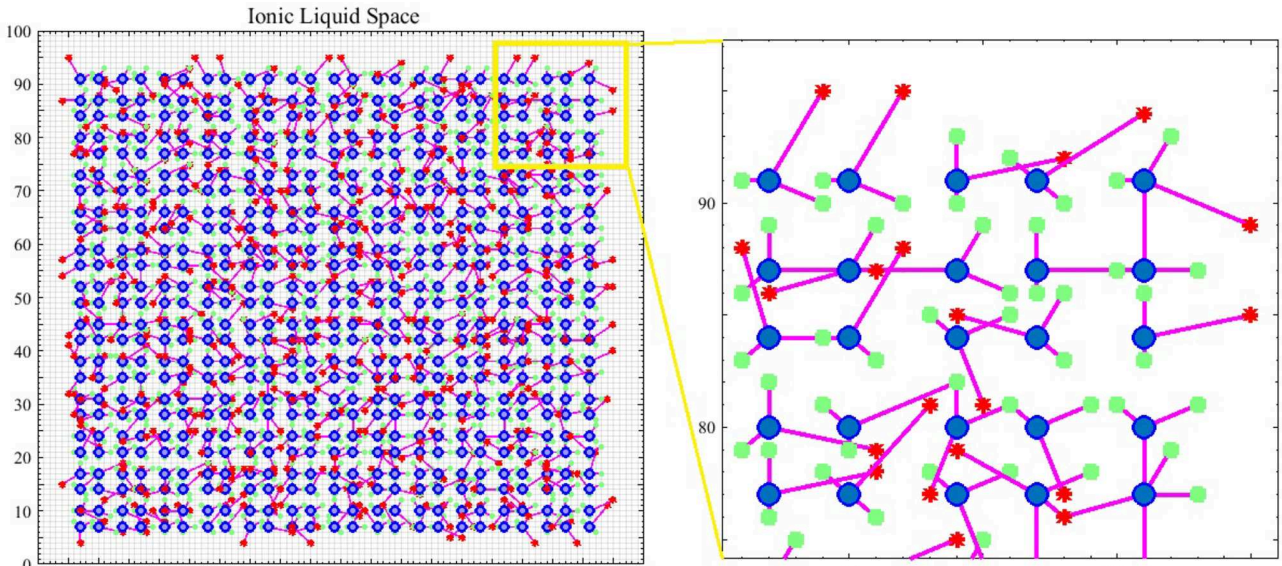

FIGURE 1 | An example of a spiking neural network in ionic liquid space. A total number of 25 × 25 = 625 spiking neurons are placed in a 100 × 100 ILS. In this example, each neuron has two dendrites and one axon. In this figure, neurons are depicted by blue circles while the location of dendrites and the terminals of axons are depicted by green squares and red stars, respectively. A small area of the ILS is zoomed in on and depicted in this figure. It can also be seen that in the proposed model, the neurons are not directly connected and instead, are connected via ILS.

the output spike into a bin from the ionic liquid space via axons at the time a neuron is fired. External stimulus can be applied to each of the bins of ILS which leads to an increase in voltage in that bin. The difference in voltage between the mentioned bin and its neighbor leads to diffusion. In order to model the diffusion process in this ionic liquid space, it is assumed that each bin located in (x, y) in the time step t holds a voltage equal to V_{xy}^{t}. Considering all eight neighborhoods of the mentioned bin, we can model the diffusion process by Equation (1) in which α is a coefficient representing the transmitted current between bins and β is another coefficient which keeps the ionic liquid space from saturating.

$$V_{xy}^{t+1} = (1-\beta) \times V_{xy}^{t} - \alpha \sum_{m=-1}^{1} \sum_{n=-1}^{1} (V_{xy}^{t} - V_{(x+m)(y+n)}^{t}) \quad (1)$$

Due to the difference in ionic density in the ionic liquid space which leads to voltage difference among different bins, ion trajectories are created. These trajectories depend on the applied stimuli and the diffusion process. In other words, each applied stimulus creates a different ion trajectory in the ionic liquid space. The created trajectories cause the neurons to activate and therefore, lead to the creation of connections between neurons. Ion trajectories can gradually crystalize or even fade. Once an ion trajectory crystalizes, it strengthens. The fading of an ion trajectory on the other hand, basically means that it gets weaker. Therefore, the connection between neurons in the introduced model is different from that of a conventional neural network. In the introduced model, the connections are made via the ionic space and the created trajectories. It could be understood that every single neuron in the ionic liquid space is capable of making connections with other neurons depending on the created trajectory.

At first, there are no proper connections between neurons in the ionic liquid space just like in regular neural networks in which the weights are initialized randomly. As the diffusion process begins, proper connections begin to develop between neurons. The connection topology of neurons in this ionic liquid space is similar to natural structures as well as brain structures (Zuo et al., 2017). In this topology, neurons are more likely to connect to neighboring neurons. So far, the general structure of the proposed computational model based on spiking neurons has been explained. The following discusses the model of spiking neurons in the ionic liquid space.

The model provided for a spiking neuron represents the creation of an action potential and the effect of neuron inputs on its membrane potential. Several models have been introduced for spiking neurons. The basic idea behind defining spiking neurons is electrical conductance. One of the most well-known electrical models for spiking neurons was introduced by Hodgkin and Huxley (1952). In this model, the electrochemical data transmitted between neurons has been modeled using an electrical circuit consisting of resistors and capacitors (Hodgkin and Huxley, 1952). This model has a high computational cost. Another model which balances computational cost with how close the model is to biological observations, is the Izhikevich model (Izhikevich, 2003). The Leaky Integrate and Fire (LIF) model (Stein, 1965; Abbott, 1999; Gerstner and Kistler, 2002) is derived from Hodgkin-Huxley's model and has, in comparison, a lower computational cost. In the LIF model, each spike represents a uniform event which is defined at the time a spike is created. Due to its simplicity and low computational cost, the LIF model has been used in the structure of the proposed model.

2.2. Description of the Proposed Computational Model

Learning capacity of a neural network depends on the plasticity of synapses. In other words, synaptic plasticity is much more important than the number of synapses and or the connections. Due to the fact that the neurons are not directly connected in the proposed model, the introduced network has high plasticity. In fact, the high plasticity of the proposed model is due to the possibility of recreation and or pruning of synapses.

Another effective factor in a network's learning is the intrinsic plasticity that keeps the performance of neurons in a relatively stable equilibrium. While describing the proposed model the intrinsic plasticity should be considered in a way that all neurons can activate. To do so, the firing threshold of neurons in the ionic liquid space should be changed in each time step.

Assuming that the number of spiking neurons in a network is N, the activity of the *ith* neuron in the time step t is represented by $x_i(t)$. If the *ith* neuron is fired, $x_i(t) = 1$, while $x_i(t) = 0$ for when it is not fired. Each neuron in the ionic liquid space has a specific number of dendrites through which the inputs are entered and an axon through which the output is transferred. If the weighting factor of the *jth* dendrite of the *ith* neuron is represented by w_{ij} and all connections are considered to be excitatory, the pre-activation value of the *ith* neuron in $t + 1$ time step which is represented by $p_i(t+1)$, can be obtained from Equation (2). It should be noted that in the proposed network, excitatory connections mean having positive weights ($w_{ij} \geq 0$). In this equation, $V^t(i,j)$ represents the voltage of a bin in ILS where the *jth* dendrite of the *ith* neuron is located in the time step t. $\nu_i(t)$ represents the threshold applied to the *ith* neuron in the time step t in order to maintain relatively stable equilibrium of neuronal activity. After finding the pre-activation value of all neurons in a specific time step, the first k neurons with the highest pre-activation values are obtained using the k-winner-take-all function (Maass, 2000). The activity status of these k neurons is set to 1 while the activity of the other $N - k$ neurons is set to 0. The k-winner-take-all function is represented by kWTA and the activity of neurons is given by Equation (3). In this equation, X is a state vector representing the activities of all neurons while P is a vector representing their pre-activation values. In this paper, ν_i is adjusted using a simple model in Lazar et al. (2007). In this simple model, ν_i is set in a way that the threshold applied to an active neuron increases while it decreases for a deactivated neuron. As a result, the active neuron will need more membrane potential to re-activate while the deactivated neuron will need less. Therefore, more neurons can be active in the proposed model. This leads to higher dynamicity of the introduced model. Equation (4) represents the simple method used to set the value of ν_i. In the given equation, η is a coefficient representing intrinsic plasticity rate.

$$p_i(t+1) = \sum_j (w_{ij} \times V^t(i,j)) - \nu_i(t) \tag{2}$$

$$X(t+1) = kWTA(P(t+1)) \tag{3}$$

$$\nu_i(t+1) = \nu_i(t) + \eta(x_i(t) - \frac{k}{N}) \tag{4}$$

2.3. An Example to Show Diffusion in ILS and the Activity of Its Neurons

A simple example, depicted in **Figure 2**, is employed to explain the diffusion process as well as neuronal activities in the ionic liquid space, based on the proposed model which we described by considering the intrinsic plasticity. In this example, higher

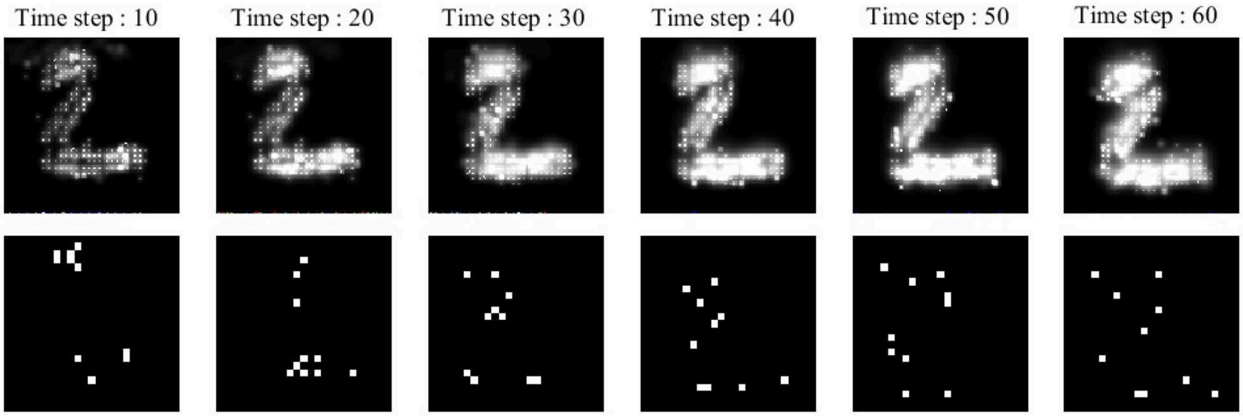

FIGURE 2 | A simple example to show how the proposed model learns a sample of the N-MNIST dataset. Each sample with a duration of about 300 *ms*, is injected into the ionic liquid space through time steps. **Upper row:** represents the diffusion effects on the voltage level of ionic liquid space at six different time steps. **Lower row:** represents the activity of neurons inside ionic liquid space considering $k = 10$. Each sample activates different neurons at different time steps which leads to better separation between samples of different classes.

voltage levels are depicted in light colors while lower voltage levels are shown in dark colors. The topology of the proposed network's model is similar to the one presented in **Figure 1**. The term "topology" refers to the local structure of neurons in the ionic liquid space as well as the number of bins on this plane, the number of neurons and the number of dendrites per neuron. In this example, the N-MNIST dataset is used. Every sample of this dataset, which holds a duration of about 300 ms, has been injected into the ionic liquid space over time. In the first row of **Figure 2**, the voltage level of the ionic liquid space at six different time steps is depicted. The second row represents the activity of neurons in the ionic liquid space. Active neurons are depicted in white while the deactivated neurons are shown in black. Using the description provided for the network in section 2.2, k is set to 10 arbitrarily. As mentioned before, k represents the number of active neurons of the network in each time step. Having a close look at **Figure 2**, it can be understood that in each time step, 10 of the neurons in the ILS are active while the rest are deactivated.

The N-MNIST dataset (Orchard et al., 2015) is in fact the neuromorphic version of the MNIST dataset (LeCun et al., 1998). Each sample from this dataset represents a number between 0 and 9. The N-MNIST dataset uses a biological idea known as saccades. Each sample from the MNIST dataset is displayed on a monitor and then, recorded by a motor-driven DVS camera, moving in three different directions in a triangular form. The duration of each N-MNIST sample is about 300 ms. Since the size of each N-MNIST picture is 34 × 34 pixels, the intensity of each pixel is applied to a bin in an ionic liquid space with a total number of 100 × 100 bins. The mentioned mapping is one to one. In this paper, in order to map each pixel to a bin in an ILS, the ILS is divided to regions overlaying one another. The number of these regions is equal to the number of pixels in the picture. Each pixel is randomly mapped to a bin in the region associated with it in ionic liquid space. Once the image is mapped to ionic liquid space in each time step, the voltage level of bins in ionic liquid space changes. Diffusion too leads to changes in the voltage level. The changes in voltage lead to changes in the activity of neurons. For each sample, different neurons in ionic liquid space are activated through time steps which results in more separations between samples of different classes.

3. EVALUATING THE PERFORMANCE OF THE PROPOSED MODEL

In this section, the performance of the proposed model is studied using popular metrics including separation, approximation, and generalization. Separation is a metric used to determine the effectiveness of the liquid which addresses the amount of separation between state vectors that are caused by different input streams. Approximation is the capability to distinguish and transform different state vectors into the given target outputs. This section also contains an example to show the performance of the proposed model in classification problems using a sound dataset.

3.1. Separation, Approximation, and Generalization

One of the methods used for measuring separation is pairwise separation considered in Maass et al. (2002) in which the separation between two different spike trains is computed as Equation (5). In this equation, the ILS internal states at the time t are represented by $X_u(t)$ and $X_v(t)$ in which $u(.)$ and $v(.)$ are two spike trains and $||.||_k$ denotes the L_k norm.

$$Pairwise\ Separation = \sum_t (||X_u(t) - X_v(t)||_2) \tag{5}$$

To show the separation capability of the proposed model in comparison with a random generated liquid, an experiment was designed. In this experiment, 200 different spike train pairs $u(.)$ and $v(.)$ with the length of 500 ms are generated and given as input to both the random generated liquid and the proposed liquid. Both of the liquids consist of $N = 25$ neurons for a fair comparison. The internal state distance separation averaged over 200 trials for two different input distances $d(u, v) = 0.1, 0.3$ is plotted in **Figure 3**. The proposed liquid and random generated LSM specification are brought in **Table 4**. This figure clearly shows the better separation capability of the proposed model in comparison with the random generated liquid. It can also be seen that more input distance causes more state distance in both of these liquids. To define distance $d(u, v)$ between the spike train pairs $u(.)$ and $v(.)$, each spike in the spike trains was replaced by a Gaussian kernel and $d(u, v)$ was then determined by the distance between the resulting continuous functions.

Another separation measure devised by Goodman and Ventura (2006) is based on the mean distance between the state vectors obtained from each of the classes in a problem. Norton and Ventura (2010) revised Goodman's definition of separation by adding the variance between state vectors of the same class. In Norton's definition of separation, the separation is divided into two parts: inter-class distance c_d and intra-class variance c_v. Assuming that n is the total number of classes, a set of state vectors X is divided into n subsets, X_l, one for each class. x represents an individual state vector. Inter-class distance c_d is defined by Equation (6) in which $\mu(X_l)$ is the center of mass for each class. It is calculated by Equation (7) where $|.|$ denotes the set cardinality. Also, the mean variance of state vectors of every class c_v is defined by Equation (8) in which $\rho(X_l)$ is the average variance of each state vector x_m within the class l from the center of mass for that class $\mu(X_l)$. It is calculated by Equation (9). Then, the separation between n input classes can be defined using Equation (10).

$$c_d = \sum_{l=1}^{n} \sum_{m=1}^{n} \frac{||\mu(X_l) - \mu(X_m)||_2}{n^2} \tag{6}$$

$$\mu(X_l) = \frac{\sum_{x_m \in X_l} x_m}{|X_l|} \tag{7}$$

$$c_v = \frac{1}{n} \sum_{l=1}^{n} \rho(X_l) \tag{8}$$

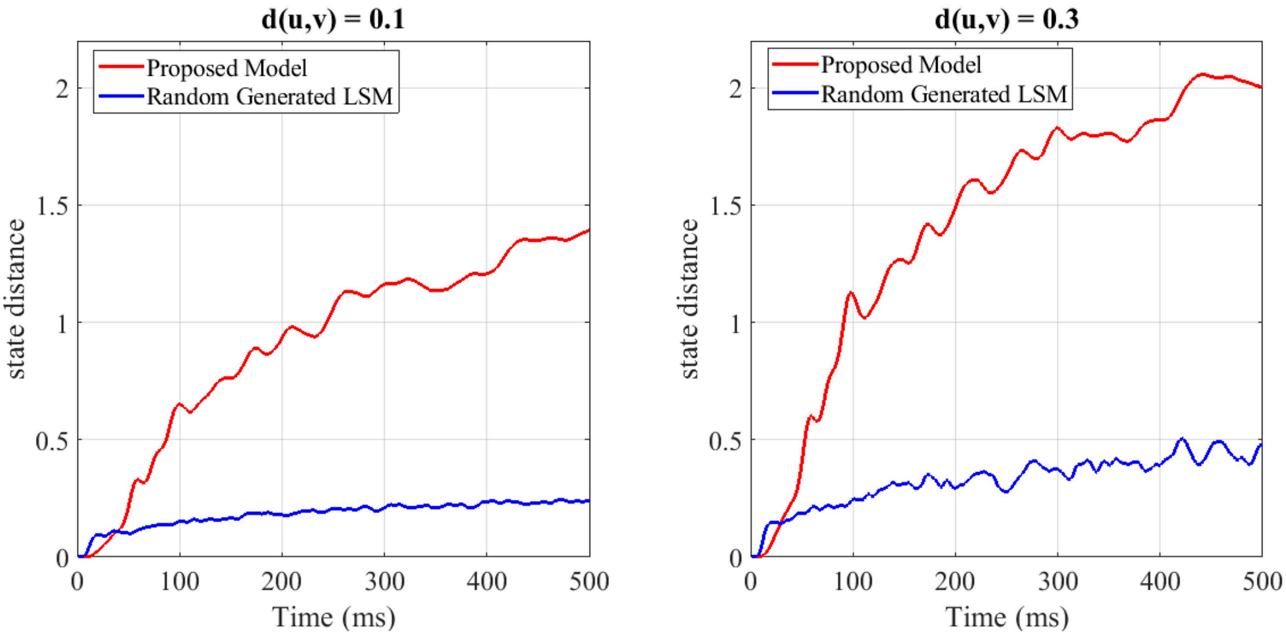

FIGURE 3 | The average state distance as a function of time for two different input distances $d(u, v) = 0.1$, 0.3 and for both the random generated liquid and the proposed liquid. It is clearly seen that the proposed liquid outperforms the random generated liquid.

$$\rho(X_l) = \frac{\sum_{x_m \in X_l} ||\mu(X_l) - x_m||_2}{|X_l|} \tag{9}$$

$$Sep(X) = \frac{c_d}{c_v + 1} \tag{10}$$

A stronger measure (linear separation) was proposed by Maass et al. (2005) in order to determine whether a readout would be able to produce given target outputs for the input streams. To evaluate the linear separation property of a liquid of N neurons for m different input streams, the rank of the $N \times m$ matrix M has to be computed. The column m of matrix M is the state vector $X_{u_m}(t)$ of the input stream u_m at the specific time (t). The rank of the matrix (r) is a measure to show the computational power of the liquid as well as the number of degrees of freedom that a linear readout has in assigning target outputs to the inputs. Higher r corresponds to more computational power or kernel quality. The readout maps are drawn from a class of functions satisfying the approximation property[1]. Besides the separation and the approximation capabilities, the generalization to non-seen or noisy inputs noticeably affects the performance of the model. Maass et al. (2005) quantifies the generalization capability in terms of VC-dimension of the class of hypothesis (H) (input-output map). The lower the VC dimension than the size of training set (S_{train}), the greater the generalization capability. Maass et al. (2005) showed the difference of the kernel quality and the VC-dimension (H) measures predicting overall computational performance.

To show the generalization capability of the proposed model, the performance of the proposed model for spike pattern classification is computed. In this experiment, we want to classify the spike patterns into two classes using a linear readout. Eighty spike patterns each consisting of 4 Poisson spike trains at 20 *Hz* over 200 *ms* are considered as input data. The noisy variations of these spike patterns built with Gaussian jitter with 10 *ms* standard deviation are used as test samples. The linear readout, trained by linear regression with 500 training samples, is responsible for classifying the test samples. The fraction of test samples (200) which are correctly classified determines the correctness. It should be noted that 20 target classification functions from 2^{80} possible classification functions are selected randomly and their correctness averaged. Classification of these spike patterns using the proposed model results in a mean correctness of 0.71 which is comparable with the correctness achieved by LSM in Maass et al. (2005). It is concluded that the proposed model has both the approximation and the generalization capabilities.

3.2. An Example to Show the Performance of the Proposed Model

To show the proposed liquid's separation property for between more than two classes of input streams, we use Norton's definition of separation for identifying free spoken digit dataset (FSDD) (Jackson, 2016) as a second test. This dataset consists of 2000 recordings of digits 0 through 9 (10 classes) obtained from 4 speakers, sampled at 8 *kHz*. To transform the digit WAV files into spike trains, first, each digit WAV file is converted into its 49 Mel frequency cepstral coefficients (MFCCs) which are then converted into spike trains. For generating spike trains,

[1] A class of functions F has the approximation property if for any $n \in N$, any closed set $X \subseteq \mathbf{R}^n$, any continuous function $h : X \to \mathbf{R}$, and any given $\rho > 0$, there exist some $f \in F$ so that $|h(x) - f(x)| \leq \rho, \forall x \in X$.

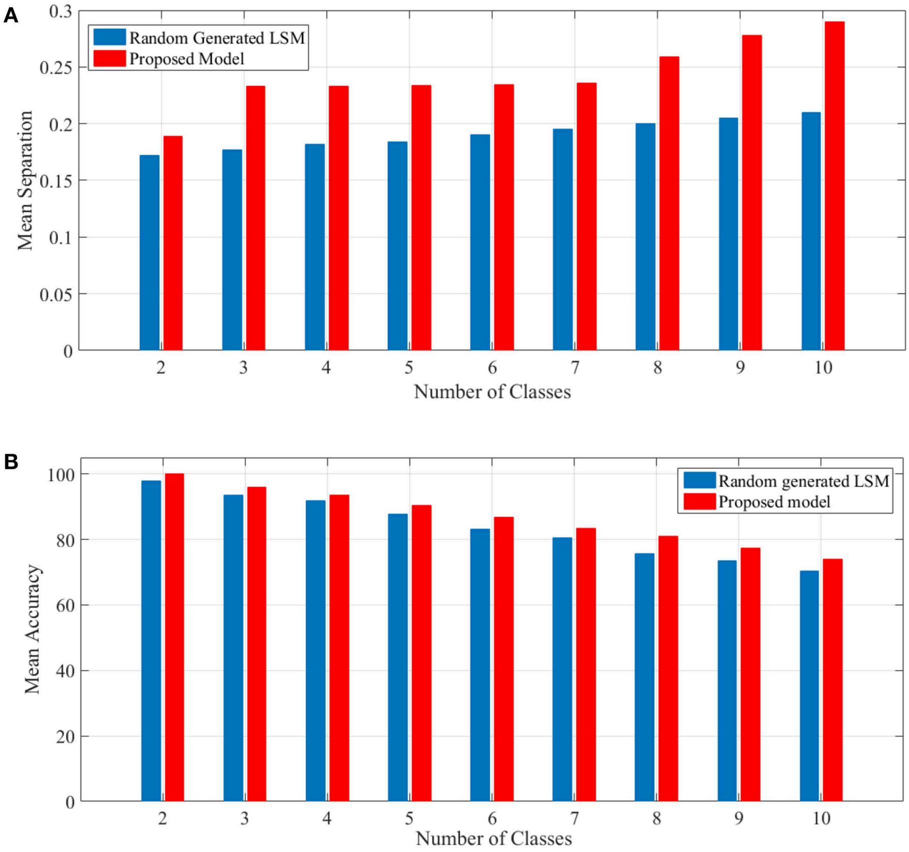

FIGURE 4 | (A) Mean separation of the proposed model in comparison with that of the random generated LSM for spoken digit problem using FSDD dataset. Separation considering different number of classes are shown. It is clearly seen that the separation capability of the proposed liquid is better than that of the random generated LSM. **(B)** Mean accuracy of the proposed model in comparison with that of the random generated LSM for spoken digit classification using FSDD dataset.

we use a method brought in Goodman and Ventura (2006). The spike trains are then injected into the random generated LSM and the proposed liquid for which the specifications are brought in **Table 4** and their separations are obtained. A comparison of mean separation between the proposed liquid and random generated liquid is shown in **Figure 4A**. Better separation capability of the proposed liquid compared to the random generated liquid can be clearly seen from **Figure 4A**.

In this experiment, 80 % of data which are selected randomly, are used as the training set and the remaining data are used for computing the accuracy. To compute the accuracy, a simple universal approximators consisting of a single layer of perceptrons is considered as the readout layer such as Maass et al. (2002). The readout layer has several modules equal to the number of classes intended. For training the readout layer, the p-delta rule (Auer et al., 2008) is used. The mean accuracy shown in **Figure 4B** depicts the outperformance of the proposed model compared to the random generated LSM. It is worth mentioning that the results are achieved by considering the problems for 50 liquids either generated randomly for LSM or the proposed model. Since the accuracy was computed using test (unseen) data, the generalization capability of the proposed model was somehow shown by this example.

4. OPTIMIZING THE NETWORK TOPOLOGY IN IONIC LIQUID SPACE

In this section, the topological parameters of the proposed neural network in ionic liquid space are optimized using an evolutionary algorithm in a way that higher classification accuracy is achieved. To do so, some parameters need to be fixed first. Assuming that the number of neurons in ionic liquid space and also the number of ILS bins are fixed, the optimized number of dendrites per neuron can be obtained by doing some trials each of which is performed for a specific number of dendrites for each neuron in ILS. After optimizing the number of dendrites for each neuron, the location of dendrites and axon terminals are optimized using genetic algorithm (GA).

In genetic algorithm, first, a population of individuals is generated based on some presumptions. This population of individuals (generation) starts to evolve toward a better solution. **Figure 5A** shows the overall process of finding the best solution using the proposed genetic algorithm. In each generation, the individuals have to be evaluated using a fitness function. The best individual of each generation is obtained based on the fitness values computed using the fitness function. The next generation of individuals is generated using the best individual

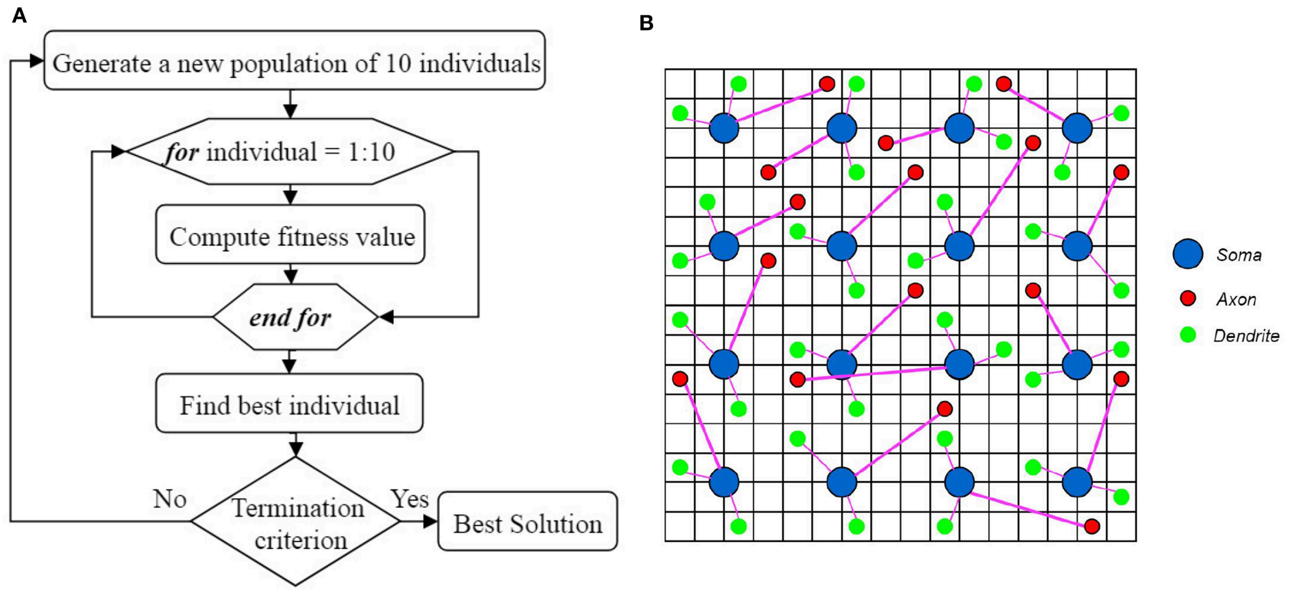

FIGURE 5 | (A) The overall process of finding the best solution. **(B)** A topology of a neural network in ILS, represented by an individual of a generation. The terminal locations of axons and dendrites of all the neurons in ILS which are shown by red and green circles respectively, are used to define each individual. Considering that this individual is the best in its generation, the terminal coordinates of axons and dendrites in the next generation of individuals are updated by randomly moving the red and green circles to the neighboring locations.

Algorithm 1: Proposed Genetic Algorithm

```
Initialization: randomly generate a population of individuals (first generation);
while termination criterion is not met do
    for every individual in the current generation do
        Evaluate each individual using a fitness function;
        Find the individual with the highest fitness value as the best individual;
    end
    Generate the next generation based on the best individual by applying small random changes to the terminal locations of
    axons and dendrites;
end
```

of each generation. The best individual in all generations is then considered as the best solution. In order to optimize the topology of the neural network in ILS, the individuals need to be represented first and then, a fitness function must be defined to evaluate each individual.

In this paper, some presumptions, including the number of dendrites per neuron as well as the length range of dendrites and axons, are obtained through trial and error. By applying these presumptions, we can reduce the number of parameters to be optimized in the proposed model. By changing these parameters, a population of individuals can be generated. In this paper, changing the location of dendrites and axons in ILS results in the creation of an individual. An individual which is a topology of a neural network in ILS, is depicted in **Figure 5B**. In this figure, the ILS consists of 16 × 16 bins and 16 neurons. Each neuron consists of 2 dendrites and 1 axon. The neurons are fixed at the specified locations. Each individual is specified by the locations of the terminals of axons and dendrites which are represented by red and green circles, respectively. Considering the fact that each individual has its own unique chromosome, each individual can be represented by its chromosome. In the N-MNIST classification problem in which we have 25 × 25 neurons in ionic liquid, considering that each neuron consists of 2 dendrites and 1 axon, an individual or a chromosome owns a total number of $(2 + 1) \times 25 \times 25 = 1875$ genes. In this problem, the first generation consisting of 10 individuals is generated randomly. It should be noted that a simple version of the genetic algorithm shown in Algorithm 1 has been employed for this problem. In this algorithm, in order to generate a new population of individuals, the best individual of the current generation has to be chosen. The best individual in a generation is the one with the highest fitness value. In this paper, the best individual is a topology of the proposed neural network in ILS which leads to higher classification accuracy. Using the best individual, a population of

Algorithm 2: The Algorithm for Evaluating an Individual

```
Result: fitness value of each individual which is a number between 0 and 1. (Better fitness is represented by numbers closer to 1)
Presumption: adding a readout layer consisting of 10 neurons (each neuron is associated with a specific class);

Training Stage: using 70 % of the training data;
for every class do
    1. Calculate the activity rate of every neuron in ILS for all the data samples associated with this class using Eq. 12;
    2. Sort the neurons in ILS based on their activity rate in descending order;
    3. Choose the specific number of neurons with higher activity rate;
    4. Connect the chosen neurons to a readout neuron associated with this class with a weight proportional to its activity rate
       using Eq. 13;
end

Evaluation Stage: using the remaining 30 % of the training data;
for every data sample do
    1. Calculate the total activity rate of the neurons ($\alpha$) in ILS;
    2. Calculate the membrane potential of readout neurons using the weights ($w$) obtained in training stage;
    3. Choosing a readout neuron with the highest membrane potential as the class for input data sample;
    4. Comparing the actual class with the obtained class: if the answer is wrong, increase the number of errors by 1;
end
Calculating the fitness value of the individual considering the number of errors using Eq. 14;
```

10 individuals for the next generation is then generated. Since each individual is represented by terminal locations of axons and dendrites of all the neurons in ILS as shown in **Figure 5B**, the next generation of individuals is generated once the red and green circles move to their neighboring locations. This makes the individuals of the next generation only slightly different from those of the current generation. By considering this simple model, we no longer need to define crossover and or mutation.

In order to obtain the best individual for a classification problem, a fitness function is defined by which the performance of each individual is evaluated. Algorithm 2 represents the evaluation method applied to compute the fitness values of individuals. In order to evaluate each individual, a layer called the readout layer needs to be added to the proposed model. For example, in N-MNIST classification problem, this layer consists of 10 neurons each of which represents a class. The neurons of ionic liquid layer are fully connected to the neurons of the readout layer. N and C represent the number of neurons in ionic liquid space and the neurons of readout layer, respectively.

In order to set the weight values of this layer, 70 % of the training data is used. In this training algorithm, after injecting the data of each class into the ILS, the activity rate of the neurons inside ILS are obtained. The activity rate of the *ith* neuron inside ILS for a given data sample is computed using Equation (11) in which T represents the total number of time steps and the activity status of the *ith* neuron in t time step is represented by x_i^t. By applying all the data samples of each class, the total activity rate of the *ith* neuron (α_i) is computed using Equation (12) in which a_i^s represents the activity rate of the *ith* neuron for the s^{th} data sample of a class. The specific number of neurons which are more active compared to other neurons in ILS are connected to the readout neuron associated with that class with a weight value proportional to their activity. The weight value of less active neurons is set to 0. By assuming A^c as the set of the most active neurons for the *cth* class, the weight between the *cth* readout neuron and the *ith* neuron inside ILS is computed using Equation (13). If the *ith* neuron inside ILS (N_i) is a subset of A^c, the weight is computed based on α_i^c while it is set to 0 if N_i is not a subset of A^c.

$$a_i = \frac{1}{T}\sum_{t=1}^{T} x_i^t \tag{11}$$

$$\alpha_i = \frac{1}{S}\sum_{s=1}^{S} a_i^s \tag{12}$$

$$w_{ic} = \begin{cases} \alpha_i^c - \frac{1}{C-1}\sum_{j=1,\, j\neq c}^{C} \alpha_i^c & N_i \subset A^c \\ 0 & N_i \not\subset A^c \end{cases} \tag{13}$$

In evaluation stage, using the weight values obtained in training stage, the remaining 30 % of the training data is classified. The neuron in the readout layer with the highest membrane potential represents the class of input data. Next, the number of total errors which occurred while classifying this remaining 30 % of training data is computed. Using Equation (14), a fitness value is calculated for each individual. In this equation, D_s and E_s indicate the desired and estimated class of the *sth* data sample of the remaining 30 % of the training data, respectively. Also, in this equation, S represents the number of data samples in the remaining 30 % of the training data. Higher fitness values represent better individuals. In summary, this section provided an explanation on how to obtain the best topology of a neural network in ILS for classification problem using optimization algorithm.

$$Fitness = 1 - \frac{1}{S}\sum_{s=1}^{S}(D_s \neq E_s) \qquad (14)$$

5. EXPERIMENTAL RESULTS

In this section, we first show the classification results on N-MNIST dataset using the optimized ionic liquid. Then, optimized ionic liquid is compared to the original LSM via separation and classification accuracy considering the same readout learning rule. Also, the performance of the proposed model in comparison with some reservoir models is studied via classification problems derived from the well-known TIMIT dataset (Garofolo et al., 1993). Finally, the scalability, stability and robustness to noisy input of the proposed ionic liquid are discussed.

5.1. Classification Results on N-MNIST Dataset by Optimizing the Topology of the Proposed Reservoir

Based on what was discussed in sections 2 and 3, it could be understood that the proposed model is appropriate for classifying neuromorphic and time variant data since diffusion plays a key role in this model. Hence, the N-MNIST dataset (Orchard et al., 2015)—which is the neuromorphic version of the MNIST dataset—has been used for classification. By applying time variant inputs to the ILS it could be said that the state of ILS is also time variant. The state of the ILS depends on whether the neurons of ILS are active or not. One of the biggest challenges of classification problem using the proposed model is being able to provide a constant output for a time-varying ILS. To do so, similar to all other reservoir networks, we need to add a readout layer to the proposed model. Here, a two-layered fully-connected readout layer similar to the one depicted in **Figure 6** is used. In order to train this network, scaled conjugate gradient backpropagation method (SCG) has been used (Møller, 1993).

In this section, the effect of the proposed model's topology evolution on its performance using genetic algorithm is shown. In fact, we would like to show that as the topology of the proposed model evolves, better classification results will be obtained. To do so, we optimize the parameters of the proposed model using a proposed genetic algorithm as well as different experiments in order to realize its effect on the classification accuracy.

The following includes an explanation on the optimization method used to optimize the topological parameters of the proposed model. In order to optimize the parameters a group of tests are designed. The parameters of each test are shown in **Table 1**. These parameters include the number of dendrites per neuron as well as the length of dendrites and axons. In the first test, 20 different topologies of the proposed model are built considering the parameters of **Table 1**. In the first test, 4 dendrites are considered per neuron and the length of dendrites and axons is a random number in a specific range. The results obtained from the classification of the N-MNIST test dataset are depicted in **Figure 7A**. Since our purpose was to study the topological parameters of the proposed model and not the readout layer, in order to eliminate the effects of the readout layer on classification accuracy, it is trained 10 times for each topology. The classification accuracy obtained in each training is different from the others. Therefore, the diagram has been depicted around the mean values of classification accuracy with confidence intervals.

The second test is designed to evaluate the effect of the number of dendrites on ionic liquid's performance. Therefore, the length of dendrites and axons are set to a constant value and the number of dendrites varies throughout the test. In this test, different topologies are generated by considering a different number of dendrites for each neuron. Similar to first test, the readout layer of the proposed neural network is trained 10 times for each topology. **Figure 7B** shows the effect of the number of each neuron's dendrites on classification performance. As shown in this figure, better classification accuracy is obtained when the number of dendrites per neuron is set to 2. Hence, in the next test, the number of dendrites per neuron is considered to be 2 as well.

In the third test, each neuron is considered to have 2 dendrites while the length of dendrites and axons is chosen randomly from the ranges given in **Table 1**. In this test, 10 different topologies of the proposed model are generated. **Figure 7C** shows the classification results of the third test for different topologies. The best obtained mean value of classification accuracy is 97.70 % which is obtained by using the 2*nd* topology while the maximum value of classification accuracy is 98.14 % which associates with the 3*rd* topology.

In order to gain better results while using genetic algorithm we use the topology with the maximum classification accuracy. Therefore, the 3*rd* topology of the third test is considered as the initial individual. Now, based on the locations of neurons, dendrites, and axons in this topology, the next generation of individuals will be generated. Each individual is evaluated using the fitness function represented in section 4 and in each generation, the individual with the highest fitness value is used to generate the next population of individuals. The best classification accuracy (best individual) of each generation is depicted in **Figure 8**. As can be seen, 30 different populations of individuals are generated. As mentioned above, the first population of individuals is generated from the 3*rd* topology of the third test. It also can be seen that the best solution is obtained in the 26*th* generation which holds the maximum value of classification accuracy which is 98.38 %.

The results obtained from comparing the proposed model with other state-of-the-art spiking neural networks, while classifying the N-MNIST dataset are shown in **Table 2**. As can be seen, all the networks used to classify this dataset are in the form of multi-layer perceptron (MLP) and or convolutional neural network (CNN). Different methods are used for training these networks. The highest accuracy of classifying this dataset obtained so far is 98.78 % (Wu et al., 2018) in which MLP structure is trained with spatiotemporal backpropagation algorithm (STBP). Network structures are also

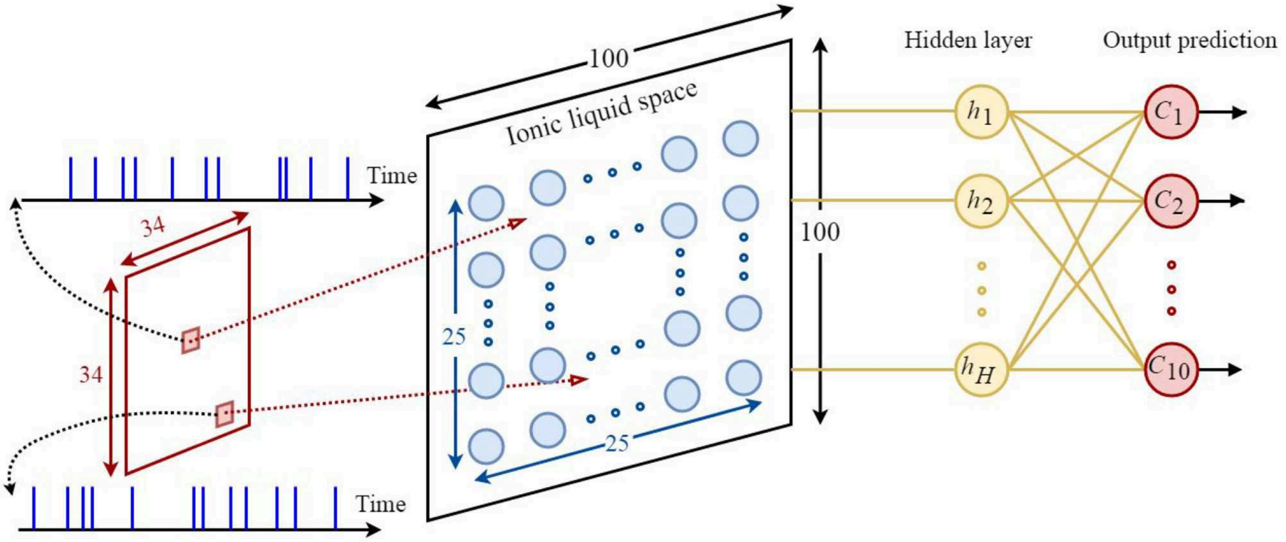

FIGURE 6 | The architecture of the proposed neural network based on the ILS model used to classify the N-MNIST dataset. Each sample consists of 34 × 34 = 1156 pixels. Each pixel is represented by a 300 *ms* event stream injected into an ILS bin. To show the states of the ILS, 25 × 25 = 625 neurons were considered in a 100 × 100 ILS. In this network the readout layer is a two-layered fully-connected structure consisting of hidden layer neurons and output prediction neurons. There are 10 neurons for predicting the class of each sample from the dataset. Each neuron represents a specific class. In this paper, the number of hidden layer neurons is considered to be 120.

TABLE 1 | Setting parameter values of the proposed model for different tests.

Test number	Number of dendrites	Dendrite length	Axon length	Number of topologies
First	4	Rand ∈ [0.5, 2]	Rand ∈ [3, 5]	20
Second	1 to 10	1.5	4	10
Third	2	Rand ∈ [0.5, 2.5]	Rand ∈ [3, 6]	10

presented in **Table 2**. It can be seen that the number of neurons of the proposed model is less than that of other available networks. In other words, using ionic liquid structure leads to obtaining a reasonable classification accuracy with a lesser number of neurons. Without applying the genetic algorithm for optimizing the structure of neurons in the ILS and only by applying the SCG training method in order to train the readout layer of the proposed model, a classification accuracy of 97.69 % is obtained. Optimizing the structure of neurons using the genetic algorithm, however, leads to obtaining a classification accuracy of 98.38 %.

5.2. Comparison With Other Reservoir Models

In order to show the performance of the proposed model in the N-MNIST classification problem compared to the original LSM architecture, we used CSIM (Natschläger et al., 2003) to implement a spike-based reservoir. We used the StaticSpikingSynapse model for synapses, and the SpikingInputNeuron and LifNeuron for the input and reservoir neurons, respectively. **Table 4** contains more details about the LSM parameters in this experiment. The mean separation of the proposed liquid in comparison with the original LSM is then computed. Our model with and without topological optimization provides 18 % and 11 % increase in liquid separabilities, respectively compared to the original LSM with static synapses. **Table 3** contains the mean separation of the original LSM and the proposed model on N-MNIST dataset. Since the number of neurons affects the separation value, the same number of neurons (25 × 25) for both the original LSM and the proposed model is considered in this experiment. To obtain the accuracy of classification, the readout layer consisting of multiple perceptrons each for a class, is considered and then trained using p-delta-rule. **Table 3** also contains the mean classification accuracy of the original LSM and the proposed model on N-MNIST test dataset. It is clearly seen that the proposed liquid outperforms the original LSM. Moreover, using a genetic algorithm to optimize the network topology in ionic liquid not only leads to increasing the accuracy, but also increases the separation capability.

As it is mentioned in the introduction, there are a lot of studies that focus on improving the liquid performance. In this section, we tend to compare the performance of our proposed model with that of other works including the random generated liquid (traditional LSM) and the improved liquid using separation

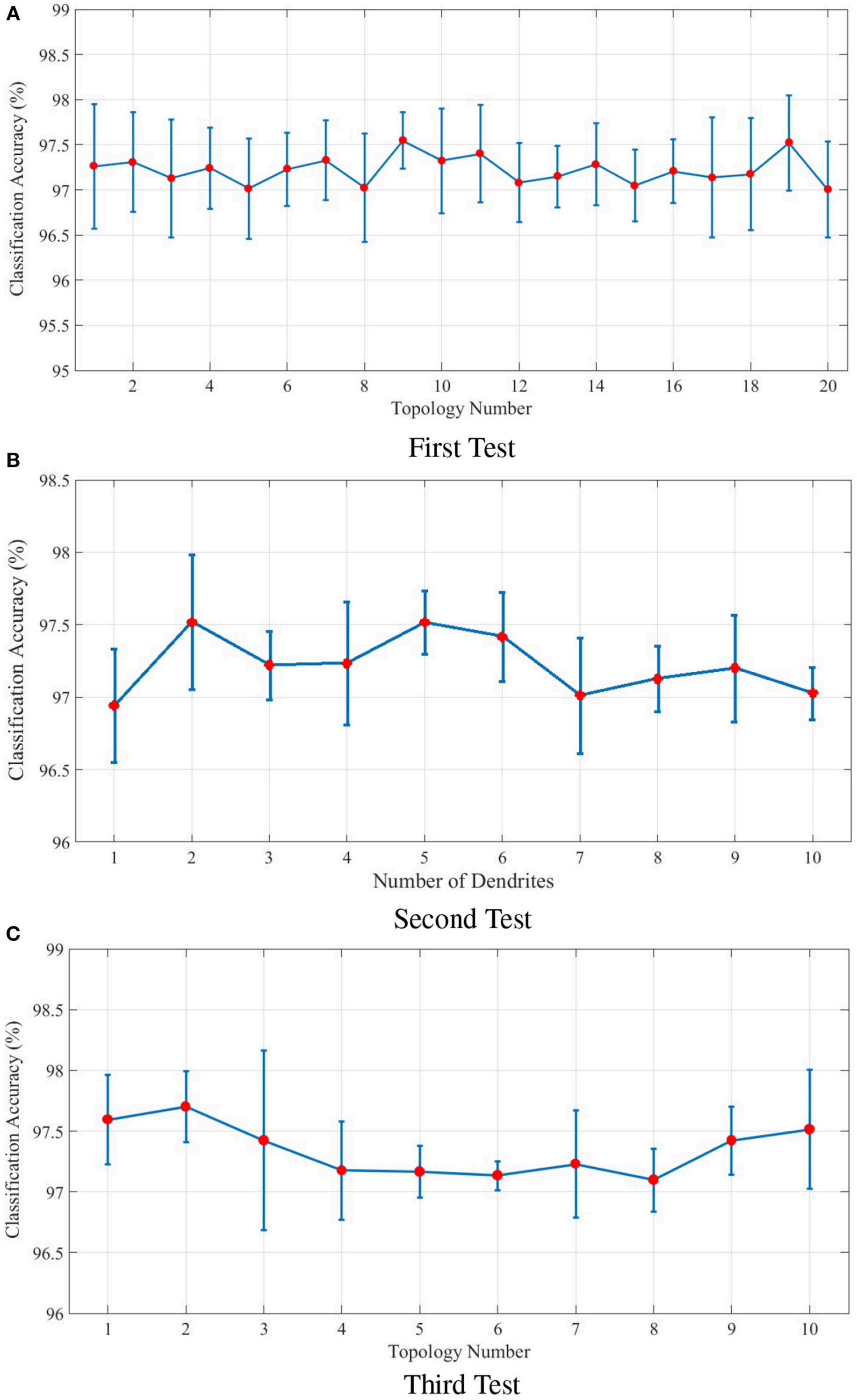

FIGURE 7 | (A) Classification results for 20 different topologies generated using the parameters of the first test mentioned in **Table 1**. In this test, each neuron in the ILS has 4 dendrites. The lengths of dendrites and axons vary for different topologies. The mean and the maximum value of classification accuracy of the topology with the best results are 97.52 and 98.04 %, respectively. **(B)** Classification results of the second test considering different number of dendrites for each neuron in the ILS. The best classification results are obtained by considering 2 dendrites per neuron. The mean and the maximum value of classification accuracy by assuming 2 dendrites for each neuron inside ILS are 97.51 and 97.98 %, respectively. **(C)** Classification results for ten different topologies generated using the parameters of the third test mentioned in **Table 1** and considering two dendrites per neuron. The best mean value for classification accuracy is 97.70 % which is obtained by using the second topology and the maximum value of classification accuracy is 98.14 % which is associated with the 3*rd* topology. The structure of the proposed ionic liquid space in the 3*rd* topology of this test is considered as the initial individual of the genetic algorithm.

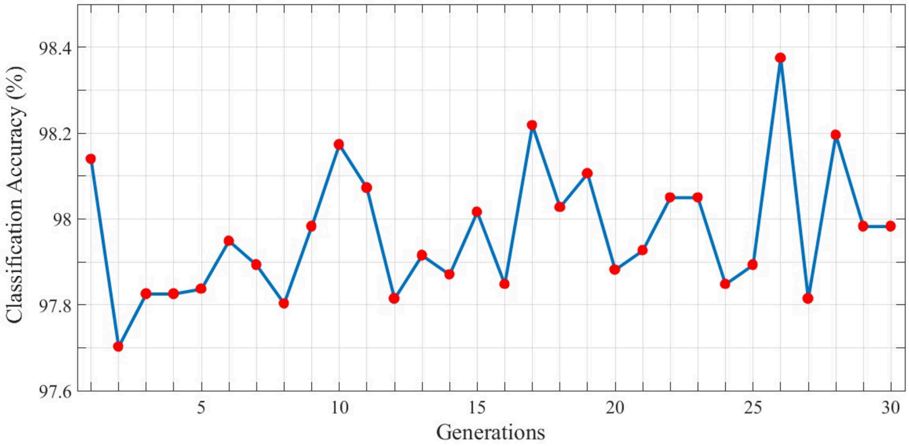

FIGURE 8 | Classification results associated with the best individual in each generation for the N-MNIST test dataset. 30 different generations, each consisting of 10 individuals are generated. The maximum value of classification accuracy is obtained for the 26*th* generation and is equal to 98.38 %.

TABLE 2 | Comparing the proposed model with other state-of-the-art spiking neural networks while classifying the N-MNIST dataset.

Model	Network structure	Learning-rule	CA (%)
MLP (Wu et al., 2018)	34 × 34 × 2 – 800 – 10	STBP	98.78
MLP (Lee et al., 2016)	34 × 34 × 2 – 800 – 10	BackProp	98.66
CNN (Stromatias et al., 2017)	18 × 22 × 22 – 2178 – 10	SGD	97.77
CNN (Neil and Liu, 2016)	–	–	95.72
MLP (Cohen et al., 2016)	34 × 34 × 2 – 10000 – 10	OPIUM (van Schaik and Tapson, 2015)	92.87
The proposed model	25 × 25 – 120 – 10	SCG	97.69
The proposed model	25 × 25 – 120 – 10	SCG & GA	98.38

TABLE 3 | The N-MNIST classification results of proposed liquid compared to the LSM.

Model	Liquid structure	Learning rule of readout	Learning rule of liquid	Mean separation	Mean accuracy (%)
LSM	25 × 25	p-delta-rule	–	0.55	90.87
Ionic liquid			–	0.62	91.48
Ionic liquid			GA	0.67	92.56

driven synaptic modification (SDSM) (Norton and Ventura, 2010) for two classification problems derived from TIMIT dataset (Garofolo et al., 1993). To compare our results, the total number of neurons in the proposed model is considered the same as the number of neurons in Table 2 of Norton and Ventura (2010). Also, we use perceptron for training the readout layer similar to Norton and Ventura (2010). The mean accuracy and mean separation of the proposed model when using TIMIT dataset (Garofolo et al., 1993) are shown in **Figure 9**. TIMIT dataset consists of 6300 spoken sentences, sampled at 16 *kHz*. This dataset consists of 52 phonemes which makes it difficult to correctly identify all of them. Hence, Norton and Ventura (2010) reduces the problem to two simpler problems: The first is identifying phonemes as either "vowels" or "consonants," and the other is identifying one of four "vowel" phonemes. In this experiment, we use the same method for converting phoneme WAV to spike trains as used in Norton and Ventura (2010). It is clearly seen that the proposed liquid performs better than the initial LSM in cases of either mean separation or mean accuracy for the testing patterns. Also, it is clearly seen that the separation and the accuracy of the proposed model optimized by genetic algorithm are better than that of without topologic optimization. Moreover, it can be seen that the performance of the optimized proposed model and the LSM trained with SDSM learning rule are very close.

5.3. Discussion

So far, the efficiency of the proposed model through separation, approximation, and generalization properties were studied. Also, the mean separation and mean accuracy for several classification problems confirmed the better performance of the proposed liquid compared to the random generated LSM. This subsection discusses some of the other capabilities of the proposed model such as scalability, stability and robustness to noisy input. Scalability of a neural network model is an attribute that describes its ability to grow and manage other demands. Hence, in order to show the scalability of the model, the proposed model was used to classify different types of datasets consisting of FSDD, TIMIT, and N-MNIST. To classify each dataset, different numbers of neurons in liquid were considered depending on the difficulty of the problem. It is also clear from the results that the proposed model can function properly with its liquid growth depending on the difficulty of the problem.

TABLE 4 | The specifications of the LSM and the proposed liquid.

LSM specification parameters	
Neuron type	LIF neuron[a] (lifNeuron)
Number of neurons	Different for each problem ($N = L \times L$)
Synaptic type	Static Synapses (StaticSpikingSynapses)
Percentage of excitatory neurons	80
Percentage of inhibitory neuron	20
Internal connectivity (λ)	2.5
Proposed liquid specification parameters	
Neuron type	Adaptive-threshold LIF neuron
Number of neurons	Different for each problem ($N = L \times L$)
Size of liquid space	($4L \times 4L$)
Intrinsic plasticity rate (η)	About 0.2
Transmitted flow coefficient α	0.1
Leakage coefficient (β)	0.05
Fraction of active neurons at any time ($\frac{k}{N}$)[b]	Small about 0.05

[a] *The parameters of LIF neuron are set to the values similar to those in Table 1 of Roy and Basu (2016).*

[b] *Typically we choose $k \ll N$ which ensures population sparseness, i.e., only a small fraction of neurons are active in the liquid at any given time.*

In order to explain how the inclusion of intrinsic plasticity helps in stabilizing the learning in ILS, the separation property of the proposed model should be carefully studied. As can be seen in Equation (10), separation is directly proportional to inter-class distance (c_d) while it is inversely proportional to intra-class variance (c_v). High separation results in better effectiveness of the model. Therefore, high inter-class distance and low intra-class variance are desirable. With low inter-class distance, the liquid behaves in a particular way regardless of the input class. That is because particular ionic trajectories are formed and crystalized in ILS which may result in activating particular neurons. The crystalized ionic trajectories act like strong synapses, leading to the liquid behaving in a particular way without considering the input class. It is such like that the input does not affect the liquid anymore. To rectify, it is necessary to increase the chaotic characteristic of the liquid. Intrinsic plasticity adjusts the neurons' thresholds to ensure homeostasis of their activity. Neurons which receive high levels of input driving them to activation, are more likely to raise their thresholds while neurons receiving lower levels of input lower theirs. This results in the increase of the chaotic level of the proposed model. As a result, the power to distinguish between different classes grows. Also, it makes the intra-class variance increase. Therefore, in order to achieve high inter-class distance and low intra-class variance, a trade-off between diffusion and IP has to be established.

In addition, robustness against noisy or shifted input is one of the advantages of the proposed model due to the effect of diffusion in ionic liquid. What influences the performance of the proposed model is the overall effect of the input data over time. Although the current input data causes the ionic density of the liquid to change which results in changing the internal liquid state, the preceding inputs have affected the ionic density. It means ionic density of ILS depends on both the current and the preceding inputs. This property works

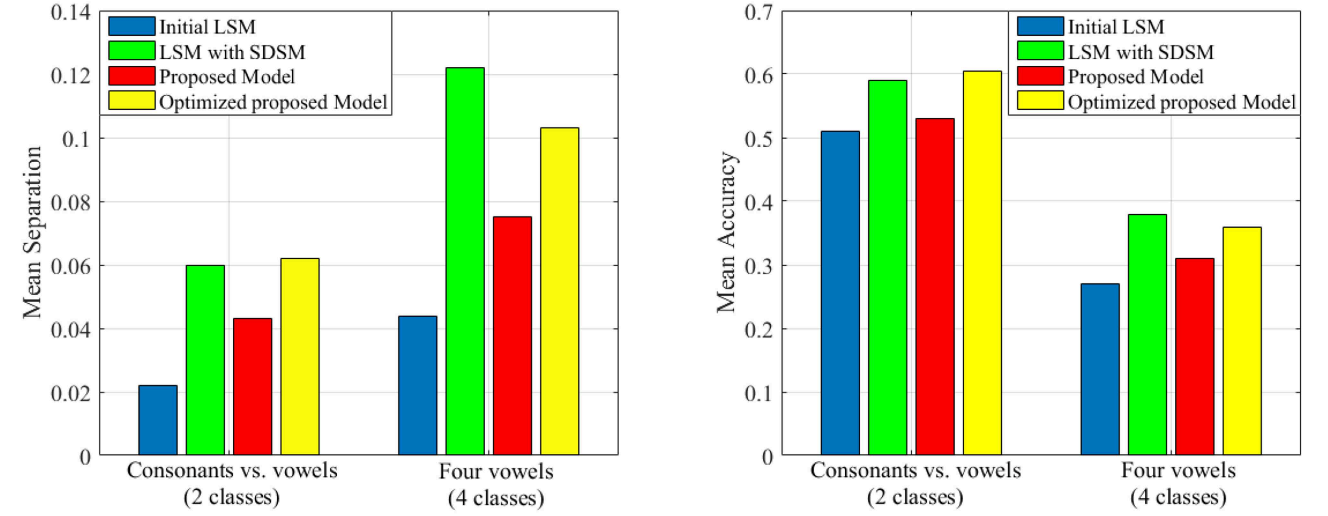

FIGURE 9 | The separation and accuracy of the proposed model in comparison with traditional LSM and SDSM across two problems derived from TIMIT dataset. The mean value of separation and accuracy are obtained for 50 liquids.

like low pass filter which results in less sensitivity to noisy input. For example, assuming the spike train as the input, if the spike occurs sooner than the predefined time, this spike affects ionic density. However, the effect of this spike diminishes. Conversely, if the spike occurs later, its effect on the ionic density increases. Robustness against noisy or shifted input was shown in section 3 while talking about generalizability of the model.

6. CONCLUSION

In this paper, a new model of reservoir networks has been introduced. The synaptic plasticity of this model leads to better learning capability. In this model, neurons are located in an ionic liquid space in which the changes of ionic density result in the neurons being fired. What makes this model different from traditional neural networks is the connection of neurons in ionic liquid space. In fact, the natural diffusion in ionic liquid space results in the creation of ionic trajectories which represent the connection of neurons. In this paper, we first showed the separation, approximation and generalization capabilities of the proposed ionic liquid by performing some experiments consisting of classifying free-spoken-digit-dataset which shows better performance of the proposed model than that of the randomly generated LSM. Then we put our focus on optimizing the topology of the proposed model or the structure of a spiking neural network in ILS. For a specific problem, the topology of the proposed model should be optimized in a way that the performance of the network improves. Since the proposed model is suitable for classifying neuromorphic and or spatiotemporal data, it has been applied for classification of the N-MNIST dataset. In order to optimize the topology of the proposed model for classifying this dataset, first, several tests have been conducted to fix some of the parameters. Then, considering these parameters and by applying the genetic algorithm, the best possible topology has been obtained. Without optimizing the topology of the proposed model and only by applying a training algorithm to the readout layer, the maximum value of classification accuracy (97.69 %) for N-MNIST dataset has been obtained. However, optimizing the topology of the proposed model has remarkably increased the classification accuracy on this dataset to 98.38 %. Based on the results obtained from topological optimization, it can be concluded that optimizing the structure of neurons in ionic liquid space results in better classification. In addition, the comparison results via classifying the two problems derived from the TIMIT dataset showed that the optimized proposed model outperforms the original LSM and it is comparable with LSM trained with some learning rules.

DATA AVAILABILITY STATEMENT

All datasets used are publicly available.

AUTHOR CONTRIBUTIONS

EI designed the proposed model, carried out the implementation of the proposed model, performed some experiments, analyzed the results, and wrote the manuscript. SS supervised the project and contributed to the final version of the manuscript. MF contributed to the design and implementation of the proposed model. NB helped to write the manuscript. BL-B helped supervise the project and provided helpful suggestions to evaluate the proposed model.

FUNDING

This work was partly funded by EU H2020 grant 824164 HERMES.

REFERENCES

Abbott, L. F. (1999). Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Res. Bull.* 50, 303–304. doi: 10.1016/S0361-9230(99)00161-6

Auer, P., Burgsteiner, H., and Maass, W. (2008). A learning rule for very simple universal approximators consisting of a single layer of perceptrons. *Neural Netw.* 21, 786–795. doi: 10.1016/j.neunet.2007.12.036

Bi, G., and ming Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu. Rev. Neurosci.* 24, 139–166. doi: 10.1146/annurev.neuro.24.1.139

Bienenstock, E., Cooper, L., and Munro, P. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* 2, 32–48. doi: 10.1523/JNEUROSCI.02-01-00032.1982

Butts, D. A., Weng, C., Jin, J., Yeh, C.-I., Lesica, N. A., Alonso, J.-M., et al. (2007). Temporal precision in the neural code and the timescales of natural vision. *Nature* 449, 92–95. doi: 10.1038/nature06105

Cohen, G. K., Orchard, G., Leng, S. H., Tapson, J., Benosman, R. B., and van Schaik, A. (2016). Skimming digits: neuromorphic classification of spike-encoded images. *Front. Neurosci.* 10:184. doi: 10.3389/fnins.2016.00184

Daoudal, G., and Debanne, D. (2003). Long-term plasticity of intrinsic excitability: learning rules and mechanisms. *Learn. Mem.* 10, 456–465. doi: 10.1101/lm.64103

Desai, N., Rutherford, L., and Turrigiano, G. (1999). Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nat. Neurosci.* 2, 515–520. doi: 10.1038/9165

Diehl, P., and Cook, M. (2016). Learning and inferring relations in cortical networks. arXiv:1608.08267.

Faraji, M. M., Bagheri Shouraki, S., and Iranmehr, E. (2017). "Unsupervised feature selection for phoneme sound classification using genetic algorithm," in *3rd Conference on Electrical and Computer Engineering Technology* (Tehran: Payame Noor University). Available online at: https://www.civilica.com/Paper-ETECH03-ETECH03_019.html

Gao, S., Zhou, M., Wang, Y., Cheng, J., Yachi, H., and Wang, J. (2019). Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 601–614. doi: 10.1109/TNNLS.2018.2846646

Gardner, B., Sporea, I., and Grüning, A. (2015). Learning spatiotemporally encoded pattern transformations in structured spiking neural networks. *Neural Comput.* 27, 2548–2586. doi: 10.1162/NECO_a_00790

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., Dahlgren, N. L., et al. (1993). *Timit Acoustic-Phonetic Continuous Speech Corpus*. Technical report.

George, R., Diehl, P., Cook, M., Mayr, C., and Indiveri, G. (2015a). Modeling the interplay between structural plasticity and spike-timing-dependent plasticity. *BMC Neurosci.* 16:P107. doi: 10.1186/1471-2202-16-S1-P107

George, R., Mayr, C., Indiveri, G., and Vassanelli, S. (2015b). "Event-based softcore processor in a biohybrid setup applied to structural plasticity," in *2015 International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)* (Krakow), 1–4.

Gerstner, W., and Kistler, W. (2002). *Spiking Neuron Models: An Introduction*. New York, NY: Cambridge University Press.

Goodman, E., and Ventura, D. (2006). "Spatiotemporal pattern recognition via liquid state machines," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (Vancouver, BC), 3848–3853.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation, 2nd Edn*. Upper Saddle River, NJ: Prentice Hall PTR.

Hazan, H., and Manevitz, L. M. (2012). Topological constraints and robustness in liquid state machines. *Expert Syst. Appl.* 39, 1597–1606. doi: 10.1016/j.eswa.2011.06.052

Hebb, D. (1949). *The Organization of Behavior*. New York, NY: Wiley.

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. doi: 10.1113/jphysiol.1952.sp004764

Hourdakis, E., and Trahanias, P. E. (2013). Use of the separation property to derive liquid state machines with enhanced classification performance. *Neurocomputing* 107, 40–48. doi: 10.1016/j.neucom.2012.07.032

Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., et al. (2004). Synfire chains and cortical songs: temporal modules of cortical activity. *Science* 304, 559–564. doi: 10.1126/science.1093173

Iranmehr, E., Shouraki, S. B., and Faraji, M. M. (2017). "Unsupervised feature selection for phoneme sound classification using particle swarm optimization," in *2017 5th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)* (Qazvin), 86–90.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *Trans. Neur. Netw.* 14, 1569–1572. doi: 10.1109/TNN.2003.820440

Izhikevich, E. M., and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proc. Natl. Acad. Sci. U.S.A.* 105, 3593–3598. doi: 10.1073/pnas.0712231105

Jackson, Z. (2016). *Free Spoken Digit Dataset (fsdd)*. Technical report.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. *GMD Report 148.* GMD - German National Research Institute for Computer Science. Available online at: http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf

Ju, H., Xu, J.-X., Chong, E., and VanDongen, A. M. (2013). Effects of synaptic connectivity on liquid state machine performance. *Neural Netw.* 38, 39–51. doi: 10.1016/j.neunet.2012.11.003

Kuhlmann, L., Hauser-Raspe, M., Manton, H. J., Grayden, D., Tapson, J., and van Schaik, A. (2013). Approximate, computationally efficient online learning in bayesian spiking neurons. *Neural Comput.* 26, 472–496. doi: 10.1162/NECO_a_00560

Lazar, A., Pipa, G., and Triesch, J. (2007). Fading memory and time series prediction in recurrent networks with different forms of plasticity. *Neural Netw.* 20, 312–322. doi: 10.1016/j.neunet.2007.04.020

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Lee, J. H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10:508. doi: 10.3389/fnins.2016.00508

Maass, W. (2000). On the computational power of winner-take-all. *Neural Comput.* 12, 2519–2535. doi: 10.1162/089976600300014827

Maass, W., Legenstein, R. A., and Bertschinger, N. (2005). "Methods for estimating the computational power and generalization capability of neural microcircuits," in *Advances in Neural Information Processing Systems* (MIT Press), 865–872.

Maass, W., Natschlager, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560. doi: 10.1162/089976602760407955

Markram, H. (2006). The blue brain project. *Nat. Rev. Neurosci.* 7, 153–160. doi: 10.1038/nrn1848

Mohemmed, A., Schliebs, S., Matsuda, S., and Kasabov, N. (2012). Span: spike pattern association neuron for learning spatio-temporal spike patterns. *Int. J. Neural Syst.* 22:1250012. doi: 10.1142/S0129065712500128

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* 6, 525–533. doi: 10.1016/S0893-6080(05)80056-5

Murakami, M., and Honda, N. (2007). A study on the modeling ability of the ids method: a soft computing technique using pattern-based information processing. *Int. J. Approx. Reason.* 45, 470–487. doi: 10.1016/j.ijar.2006.06.022

Natschläger, T., Maass, W., and Markram, H. (2002). The "liquid computer": a novel strategy for real-time computing on time series. *Special Issue Found. Inform. Process. TELEMATIK* 8, 39–43.

Natschläger, T., Markram, H., and Maass, W. (2003). "Computer models and analysis tools for neural microcircuits," in *Neuroscience Databases: A Practical Guide*, ed R. Kötter (Boston, MA: Springer), 123–138. doi: 10.1007/978-1-4615-1079-6_9

Neil, D., and Liu, S. (2016). "Effective sensor fusion with event-based sensors and deep network architectures," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)* (Montreal, QC), 2282–2285.

Norton, D., and Ventura, D. (2010). Improving liquid state machines through iterative refinement of the reservoir. *Neurocomputing* 73, 2893–2904. doi: 10.1016/j.neucom.2010.08.005

Notley, S., and Gruning, A. (2012). "Improved spike-timed mappings using a tri-phasic spike timing-dependent plasticity rule," in *The 2012 International Joint Conference on Neural Networks (IJCNN)* (Brisbane, QLD), 1–6.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *J. Math. Biol.* 15, 267–273. doi: 10.1007/BF00275687

Orchard, G., Jayawan, A., Cohen, G. K., and Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* 9:437. doi: 10.3389/fnins.2015.00437

Panchev, C., and Wermter, S. (2004). Spike-timing-dependent synaptic plasticity: from single spikes to spike trains. *Neurocomputing* 58–60, 365–371. doi: 10.1016/j.neucom.2004.01.068

Panda, P., and Roy, K. (2017). Learning to generate sequences with combination of hebbian and non-hebbian plasticity in recurrent spiking neural networks. *Front. Neurosci.* 11:693. doi: 10.3389/fnins.2017.00693

Ponulak, F. (2005). *Resume-New Supervised Learning Method for Spiking Neural Networks.* Technical Report, Poznan, Poland.

Ponulak, F., and Kasiński, A. (2010). Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neural Comput.* 22, 467–510. doi: 10.1162/neco.2009.11-08-901

Rhéaume, F., Grenier, D., and Bossé, E. (2011). Multistate combination approaches for liquid state machine in supervised spatiotemporal pattern classification. *Neurocomputing* 74, 2842–2851. doi: 10.1016/j.neucom.2011.03.033

Roberts, T., Tschida, K., Klein, M., and Mooney, R. (2010). Rapid spine stabilization and synaptic enhancement at the onset of behavioural learning. *Nature* 463, 948–952. doi: 10.1038/nature08759

Rolls, E. T., and Tovee, M. J. (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *J. Neurophysiol.* 73, 713–726. doi: 10.1152/jn.1995.73.2.713

Roy, S., and Basu, A. (2016). An online structural plasticity rule for generating better reservoirs. *Neural Comput.* 28, 2557–2584. doi: 10.1162/NECO_a_00886

Roy, S., and Basu, A. (2017). An online unsupervised structural plasticity algorithm for spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 900–910. doi: 10.1109/TNNLS.2016.2582517

Roy, S., San, P. P., Hussain, S., Wei, L. W., and Basu, A. (2016). Learning spike time codes through morphological learning with binary synapses. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 1572–1577. doi: 10.1109/TNNLS.2015.2447011

Schrauwen, B., Verstraeten, D., and Van Campenhout, J. (2007). "An overview of reservoir computing: theory, applications and implementations," in *Proceedings of the 15th European Symposium on Artificial Neural Networks* (Bruges), 471–482.

Shouraki, S. B., and Honda, N. (1997). "A new method for establishing and saving fuzzy membership function," in *13th Symposium of FUZZY Systems* (Toyama), 91–94.

Shouraki, S. B., and Honda, N. (1999). Recursive fuzzy modeling based on fuzzy interpolation. *JACIII* 3, 114–125. doi: 10.20965/jaciii.1999.p0114

Sillin, H. O., Aguilera, R., Shieh, H.-H., Avizienis, A. V., Aono, M., Stieg, A. Z., et al. (2013). A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology* 24:384004. doi: 10.1088/0957-4484/24/38/384004

Simon, D. (2013). *Evolutionary Optimization Algorithms*. New York, NY: John Wiley & Sons Inc.

Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Annu. Rev. Neurosci.* 3, 919–926. doi: 10.1038/78829

Sporea, I., and Grüning, A. (2013). Supervised learning in multilayer spiking neural networks. *Neural Comput.* 25, 473–509. doi: 10.1162/NECO_a_00396

Stein, R. B. (1965). A theoretical analysis of neuronal variability. *Biophys. J.* 5, 173–194. doi: 10.1016/S0006-3495(65)86709-1

Stromatias, E., Soto, M., Serrano-Gotarredona, T., and Linares-Barranco, B. (2017). An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. *Front. Neurosci.* 11:350. doi: 10.3389/fnins.2017.00350

Toga, A., Thompson, P., Mori, S., Amunts, K., and Zilles, K. (2006). Towards multimodal atlases of the human brain. *Nat. Rev. Neurosci.* 7, 952–966. doi: 10.1038/nrn2012

Tschida, K. A., and Mooney, R. (2012). Deafening drives cell-type-specific changes to dendritic spines in a sensorimotor nucleus important to learned vocalizations. *Neuron* 73, 1028–1039. doi: 10.1016/j.neuron.2011.12.038

van Schaik, A., and Tapson, J. (2015). Online and adaptive pseudoinverse solutions for elm weights. *Neurocomputing* 149, 233–238. doi: 10.1016/j.neucom.2014.01.071

Wang, W., Zhu, J. Z., Chang, K. T., and Min, K. T. (2012). Dscr1 interacts with fmrp and is required for spine morphogenesis and local protein synthesis. *EMBO J.* 31, 3655–3666. doi: 10.1038/emboj.2012.190

Wojcik, G. M. (2012). Electrical parameters influence on the dynamics of the hodgkin-huxley liquid state machine. *Neurocomputing* 79, 68–74. doi: 10.1016/j.neucom.2011.10.007

Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* 12:331. doi: 10.3389/fnins.2018.00331

Xue, F., Hou, Z., and Li, X. (2013). Computational capability of liquid state machines with spike-timing-dependent plasticity. *Neurocomputing* 122, 324–329. doi: 10.1016/j.neucom.2013.06.019

Zuo, X.-N., He, Y., Betzel, R. F., Colcombe, S., Sporns, O., and Milham, M. P. (2017). Human connectomics across the life span. *Trends Cogn. Sci.* 21, 32–45. doi: 10.1016/j.tics.2016.10.005

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.