Dendritic Neural Network: A Novel Extension of Dendritic Neuron Model

Cheng Tang, *Member, IEEE*, Junkai Ji, *Member, IEEE*, Yuki Todo, *Member, IEEE*, Atsushi Shimada, *Member, IEEE*, Weiping Ding, *Senior Member, IEEE*, and Akimasa Hirata, *Fellow, IEEE*

***Abstract*—The conventional dendritic neuron model (DNM) is a single-neuron model inspired by biological dendritic neurons that has been applied successfully in various fields. However, an increasing number of input features results in inefficient learning and gradient vanishing problems in the DNM. Thus, the DNM struggles to handle more complex tasks, including multiclass classification and multivariate time-series forecasting problems. In this study, we extended the conventional DNM to overcome these limitations. In the proposed dendritic neural network (DNN), the flexibility of both synapses and dendritic branches is considered and formulated, which can improve the model's nonlinear capabilities on high-dimensional problems. Then, multiple output layers are stacked to accommodate the various loss functions of complex tasks, and a dropout mechanism is implemented to realize a better balance between the underfitting and overfitting problems, which enhances the network's generalizability. The performance and computational efficiency of the proposed DNN compared to state-of-the-art machine learning algorithms were verified on 10 multiclass classification and 2 high-dimensional binary classification datasets. The experimental results demonstrate that the proposed DNN is a promising and practical neural network architecture.**

***Index Terms*—Dendritic neuron model, dendritic neural network, dropout mechanism, multiclass classification.**

I. Introduction

ARTIFICIAL neural networks (ANN) are a subset of machine learning (ML) models inspired by the structure and function of biological neurons [1]. ANNs comprise interconnected nodes or artificial neuron models organized in layers, and they can mimic the human brain in processing complex data inputs to produce output predictions [2]. The development of ANNs has a long and rich history that spans several decades of research and innovation in computer science, mathematics, and cognitive neuroscience [3].

In 1943, McCulloch and Pitts pioneered the first computational neuron model inspired by biological neurons and based on binary logic and threshold functions [4]. This model, which is referred to as the McCulloch-Pitts neuron, formed the basis for many early ANNs and laid the foundation for the computational neuroscience field. In 1958, Rosenblatt developed an early ANN, which is well-known as a perceptron that could perform basic pattern recognition tasks [5]. Despite its early success in binary classification tasks, the perceptron has several limitations that hinder its performance in more complex applications [6]. For example, the perceptron cannot handle linearly indivisible data. In other words, the perceptron can only classify input data in the input space that can be separated by linear boundaries or hyperplanes [7]. However, the perceptron paved the way for more sophisticated ANNs, e.g., multilayer neural networks and deep learning techniques that can handle complex and high-dimensional data [8]. Currently, ANNs are used extensively in many applications, including speech recognition [9], [10], image processing [11], [12], natural language processing [13], [14], and robotics [15], [16].

There are more than 10^4 neurons per cubic millimeter in the human brain. Neurons primarily comprise dendrites, one axon, and one soma body [17]. In total, approximately 10^{11} neurons and 10^{15} connections are integrated into complex neural networks that perform various brain functions. Note that dendrites are more than simple passive information conduits. They are active computational units that can process and convert signals from other neurons or sensory cells [18], and produce local spikes that propagate back to the cell body and modulate the output response of the neuron [19]. The dendritic branches of neurons are very complex and variable, each receiving inputs from various sources and locations. The inputs received by dendrites can be excitatory, which means that they depolarize dendrites and make them more likely to excite action potentials, or inhibitory, implying that they hyperpolarize dendrites and make them less likely to excite action potentials [20]. In addition to computational functions, dendrites play a critical role in plasticity, i.e., the nervous system's ability to adapt based on learning. Dendritic plasticity occurs at multiple levels, including changes in dendritic morphology, dendritic excitability, and

Manuscript received 31 August 2023; accepted 20 December 2023. This work was supported by JST CREST under Grant JPMJCR22D1, Grant JSPS KAKENHI, and Grant JP22H00551, Japan. *(Corresponding authors: Cheng Tang; Akimasa Hirata.)*

Cheng Tang and Atsushi Shimada are with the Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan (e-mail: tang@ait.kyushu-u.ac.jp; atsushi@limu.ait.kyushu-u.ac.jp).

Junkai Ji is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: jijunkai@szu.edu.cn).

Yuki Todo is with the Faculty of Electrical and Computer Engineering, Kanazawa University, Kanazawa 920-1192, Japan (e-mail: yktodo@se.kanazawau.ac.jp).

Weiping Ding is with the School of Information and Technology, Nantong University, Nantong 226019, China (e-mail: dwp9988@163.com).

Akimasa Hirata is with the Department of Electrical and Mechanical Engineering, Nagoya Institute of Technology, Nagoya 466-0061, Japan (e-mail: ahirata@nitech.ac.jp).

This article has supplementary downloadable material available at https://doi.org/10.1109/TETCI.2024.3367819, provided by the authors.

Recommended for acceptance by Prof. M. Gong.

Digital Object Identifier 10.1109/TETCI.2024.3367819

synaptic strength [21], and these changes can be induced by various forms of synaptic activity and can strengthen or weaken specific synaptic connections [22]. The changes in synaptic strength are believed to be the cellular basis of learning and memory in the brain, and they are critical for forming new neural connections and reorganizing existing connections [23]. Dendritic plasticity is regulated by complex interplay between intrinsic dendritic properties and extrinsic factors, e.g., neuromodulators and growth factors [24]. With their branching and spines, the unique structure of dendrites allows them to integrate information from multiple synaptic inputs, which enables them to detect and respond to patterns of activity that can induce plasticity [25]. Overall, dendritic plasticity is essential for the brain's ability to adapt to changing environments and to form and modify neural circuits [26].

Inspired by biological dendritic neurons, we have previously proposed a simple neuronal model featuring a dendritic structure [27]. The synaptic layer, dendritic layer, membrane layer, and soma body collectively form the main structures of the dendritic neuron model (DNM). Consistent with neurobiological observations in the brain, the structure of the DNM exhibits sufficient plasticity to discard useless synapses and redundant dendrites after training; thus, it can produce a unique dendritic structure for each task [28]. The DNM has been used to solve several vision problems, e.g., motion recognition [29], orientation detection, and depth rotation [30]. Subsequently, the model was improved by introducing simple and powerful multiplication and summation operations rather than soft-minimum and maximum functions in the dendritic and membrane layers, respectively [31]. The neuronal architecture of the DNM was verified to be completely hardwareized by logic circuits consisting of comparators and logic gates (e.g., AND, OR, and NOT gates), which is a breakthrough for the DNM in the pattern recognition field [32]. Compared with other ML methods that require floating-point computation, logic circuits perform computation in binary, which results in extremely fast processing time at minimal computational costs [33]. In addition, the DNM serves as a support for the hypothesis that logical computation can realize synaptic interactions on dendrites. Consequently, research on the neuronal architecture and learning algorithms of DNM has gained momentum and been successfully applied to solve problems in various fields [34], e.g., medical diagnosis [35], credit assessment [36], and other classification problems [37]. The DNM has also been applied to solve various time-series forecasting problems, e.g., epidemic transmission propensity [38], wind speed prediction [39], and stock price movement [40].

However, the single-neuron structure limits DNM performance significantly and makes it difficult to handle more complex problems. Combining multiple DNMs has been successful in the object motion direction detection task; however, it is still of limited help in terms of unleashing the capabilities of the DNM [41]. With the advancements of computational resources, especially advances in graphics processing units (GPUs) and data processing units (DPUs), it is necessary to improve the DNM as a deep neural network rather than a single-neuron model. Thus, in this paper, we propose a DNM-based dendritic neural network (DNN). Differing from the conventional DNM,

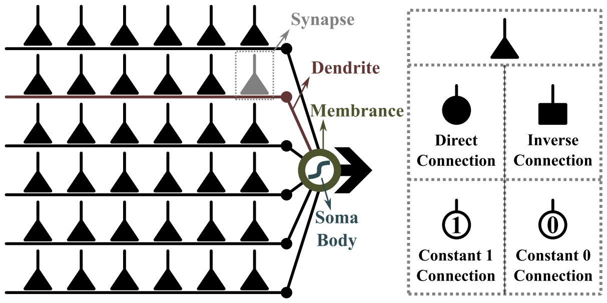

Fig. 1. Structure of dendritic neuron model and four synaptic states. The trained synapse can evolve to the four states in the right half of the figure.

the proposed DNN includes multiple membrane layers and soma bodies; thus, it can produce multiple outputs to better handle complex problems. In addition, the flexible synaptic structure enriches the nonlinear capability of the DNN, thereby making it more proficient in handling increasingly complex problems. The inherent properties of the DNM ensure that the gradient vanishing problem intensifies as the number of features increases; therefore, a dropout mechanism is designed and implemented to mitigate gradient vanishing as an optional strategy in the proposed DNN. The proposed DNN is applied to 10 multiclass classification and 2 high-dimensional binary classification problems to examine its effectiveness. In summary, the primary contributions of this study are summarized as follows.

1) The DNN model is proposed to extend the DNM from a single-neuron model to a neural network to handle more complex problems effectively.
2) The flexible synaptic structure enhances the ability of the proposed DNN to handle nonlinear tasks. Flexible synapses consider each feature sufficiently and allow the proposed DNN to realize an effective balance between underfitting and overfitting.
3) A dropout mechanism designed specifically for the proposed DNN provides an effective adjustable strategy to mitigate the gradient vanishing problem caused by cumulative multiplication.
4) The dendritic structure with multiple soma bodies enables sufficient adaptation to more diverse loss functions. In addition, it is feasible to concatenate multiple outputs into a fully-connected network or input them into a new DNN to construct a deep neural network.

The remainder of this paper is organized as follows. Section II reviews the structure of the DNM and its unique properties. The proposed DNN is described in detail in Section III. Section IV describes the datasets, experimental setup, and performance metrics used to evaluate the DNN. An analysis of the experimental results is summarized in Section IV. Finally, conclusions and future work are discussed in Section V.

II. Related Works

A. *Dendritic Neuron Model*

The DNM is a feedforward neuron model that mimics biological dendritic neurons, primarily comprising synapses, dendrites, a membrane, and a soma body. As shown in the left-hand panel of

Fig. 1, multiple synapses are attached to the same dendrite, and multiple dendrites are summarized into a membrane layer and a cell body. To be specific, synapses are responsible for processing the input information as signal-receiving units and can evolve into four synaptic states after training. Dendrites collect and process information from all synapses connected to them and transmit this information to the membrane. The dendritic signals are accumulated in the membrane and are carried into the soma body, which is responsible for producing the final output. The conventional DNM can be expressed mathematically as follows:

1) Synaptic Layer:

$$S_{i,j} = \frac{1}{1+e^{-d(w_{i,j}x_i - b_{i,j})}}, \tag{1}$$

2) Dendritic Layer:

$$D_j = \prod_{i=1}^{I} S_{i,j}, \tag{2}$$

3) Membrane Layer:

$$M = \sum_{j=1}^{J} D_j, \tag{3}$$

4) Soma Body:

$$O = \frac{1}{1+e^{-\lambda(M-\theta_{soma})}}, \tag{4}$$

where $i \in [1, 2, \ldots, I]$ and $j \in [1, 2, \ldots, J]$. Here, I and J represent the number of synapses on each dendrite and the number of dendrites, respectively, and $w_{i,j}$ and $b_{i,j}$ refer to the connection weights and biases of each synapse, which are optimized according to the learning algorithm. d is a distance factor, which is set to a constant value in the conventional DNM. λ and θ_{soma} denote the steepness factor and activation threshold of the soma body, respectively, which are also predefined constants in the conventional DNM. The activation threshold of each synapse can be formulated as follows:

$$\theta_{i,j} = \frac{w_{i,j}}{b_{i,j}}. \tag{5}$$

B. Synaptic Evolution

$w_{i,j}$ and $b_{i,j}$ can be modified during training and solidified into four synaptic states after the learning process is completed. The four synaptic states are presented in the right-hand side of Fig. 1, which can be expressed as follows:

1) Direct Connection:

$$\{(w_{i,j}, b_{i,j})|w_{i,j} > b_{i,j} > 0\}, \tag{6}$$

2) Inverse Connection:

$$\{(w_{i,j}, b_{i,j})|w_{i,j} < b_{i,j} < 0\}, \tag{7}$$

3) Constant 1 Connection:

$$\{(w_{i,j}, b_{i,j})|w_{i,j} > 0 > b_{i,j} \cup 0 > w_{i,j} > b_{i,j}\}, \tag{8}$$

4) Constant 0 Connection:

$$\{(w_{i,j}, b_{i,j})|w_{i,j} < 0 < b_{i,j} \cup 0 < w_{i,j} < b_{i,j}\}. \tag{9}$$

In the synaptic state of the direct connection, the output approaches 1 when the input x_i reaches the corresponding threshold $\theta_{i,j}$ and approximates 0 otherwise. In contrast, in the synaptic state of the inverse connections, when the input x_i exceeds the corresponding threshold $\theta_{i,j}$, its output is 0; otherwise, it is 1. In the synaptic states with constant 1 and constant 0 connections, regardless of the input x_i, the outputs are always 1 and 0. The DNM is consistent with biological dendritic neurons, with outputs 0 and 1 corresponding to biological excitatory and inhibitory signals, respectively.

C. Dendritic Neuron Model Analysis

The conventional DNM is a feedforward multiple-input single-output neuron model that has been widely utilized to solve classification and prediction problems [42], [43], [44]. When handling binary classification problems, benefiting from cumulative multiplication in dendritic layers, the evolved structure can be simplified according to a neuronal pruning strategy, which corresponds to plasticity in biology [45]. Specifically, synapses with a constant 1 connection can be eliminated, and all dendrites containing synapses with constant 0 connections can be removed completely.

However, cumulative multiplication exacerbates the gradient vanishing problem, which makes it difficult for the DNM to solve tasks with many features. Any value multiplied by 0 is equal to 0; thus, even if there is only a single synapse with a constant 0 connection on a dendrite, all other synapses on that dendrite will become meaningless. The dendrite will degenerate, and its output is constantly 0. As the number of features in the dataset increases, so does the number of synapses on the dendrites. Simultaneously, the risk of dendrites containing synapses that fall into saddle points increases, which makes it difficult for the DNM to solve problems containing too many features [46]. Previous studies have investigated ways to mitigate synapses falling into saddle points; however, the main focus of these studies was learning algorithms [47], [48], [49]. Although the DNM performs well when handling traditional multi-input single-output tasks, it is difficult for a single DNM to handle more complex tasks, e.g., multi-output problems. In addition, the solidified structure prevents DNMs from being developed as deep learning techniques because it is difficult to stack DNMs or connect them to other ANN models.

The proposed DNN is designed to improve the conventional DNM in terms of its structure. The flexibility of the synapses allows the proposed DNN to have a stronger nonlinear ability to handle more features, and the dropout mechanism enhances the ability of synapses to jump out of the saddle point during the learning process. In addition, expansion of the output layer (i.e., the soma body) enables the proposed DNN to solve multi-output problems efficiently and provides the ability to stack and connect multiple DNNs to other ANNs.

D. Ensemble Dendritic Neuron Models

The DNM is a neuron model designed for binary classification problems; thus, it is practical to collaborate with multiple DNMs when handling multiclass classifications. Generally, one of the

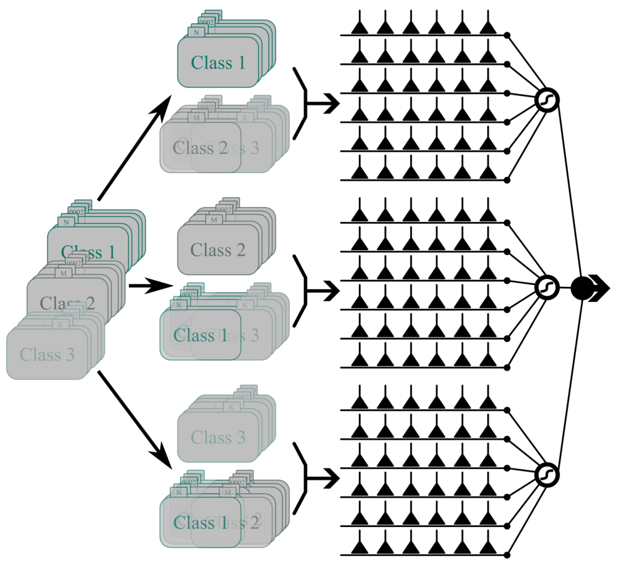

Fig. 2. EDNMs with the One-vs-Rest decomposition scheme. The dataset is binary divided multiple times based on the classes to train individual DNM.

basic concepts involved in solving a multiclass classification task is to decompose the problem into multiple binary classifications. There are three classical decomposition schemes, i.e., the One-vs-One, One-vs-Rest, and Many-vs-Many schemes [50]. Here, the One-vs-One scheme involves one-to-one pairing of N classes in the dataset, resulting in $N(N-1)/2$ binary classifications, and ultimately the most predicted class as the decision outcome based on the $N(N-1)/2$ classification results [51]. The One-vs-Rest scheme involves training N classifiers by alternating one of the N classes as the positive side and the samples of all other classes as the negative side. Here, if only a single classifier is predicted as positive, the corresponding class label is adopted as the final result, and if multiple classifiers are labeled as positive, the confidence interval is considered, and the one with the highest confidence level is selected as the result [52]. The Many-vs-Many scheme involves taking turns with several classes as positive and several classes as negative according to a specific encoding scheme, e.g., error correcting output codes, and both the One-vs-One and One-vs-Rest schemes are special cases of the Many-vs-Many scheme [53]. Inevitably, the number of classifiers to be utilized in the One-vs-One scheme is greater than that of the One-vs-Rest scheme; however, each classifier in the One-vs-One scheme requires samples from only two classes, whereas each classifier in the One-vs-Rest scheme requires all samples. As a result, the storage and computational costs of the One-vs-Rest scheme are higher. The performance depends primarily on the particular data features and distribution; but the two schemes generally exhibit comparable performance in the majority of cases [54].

Thus, the proposal for an ensemble of dendritic neuron models (EDNM) based on the One-vs-Rest scheme is natural, and each DNM in the EDNMs is responsible for identifying one of the categories in the dataset [55]. As shown in Fig. 2, the dataset is repeated three times according to the three classes, and one of the classes is selected to distinguish it from the other two classes each time. The final solid black circle represents the loss function. Note that the cross-entropy function is utilized in this study. As discussed previously, significant effort has been dedicated to EDNMs; however, this method suffers from the following limitations and challenges.

1) It is necessary for each individual DNM to handle all samples from all classes, which is a significant waste of data storage and computational resources.
2) The dendrites of the DNM have a significant impact on the final output, whereas the dendrites of each DNM contribute very little to the other DNMs.
3) A new DNM must be added for each additional class in the dataset. This structure lacks flexibility, which increases the redundancy and inefficiency of dendrites.
4) EDNMs are an extension of the conventional DNM specifically designed to solve multiclass classification problems, which limits the generalizability and evolvability of the model.

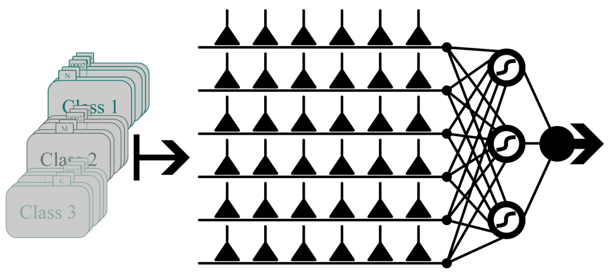

Fig. 3. Structure of proposed DNN for multiclass classification. The dataset is fed directly into the DNN, which optimizes the model structure and conserves computational resources.

III. Material

A. Dendritic Neural Network

As a neural network-based extension of the conventional DNM, the proposed DNN primarily includes synapses, dendrites, membranes, and soma bodies, as shown in Fig. 3. In addition, the flexibility of the proposed DNN allows it to handle more complex tasks. The mathematical details of the proposed DNN are expressed as follows:

1) Synaptic Layer:

$$S_{i,j} = \frac{1}{1+e^{-d_{i,j}(w_{i,j}x_i - b_{i,j})}}, \tag{10}$$

2) Dendritic Layer: refer to (2).
3) Membrane Layer:

$$M_n = \sum_{j=1}^{J} v_{j,n} D_j, \tag{11}$$

4) Soma Body:

$$O_n = \frac{1}{1+e^{-\lambda(M_n - \theta_{soma})}}, \tag{12}$$

where $d_{i,j}$ denotes the distance parameter of the synapses, which enhances the flexibility of the synaptic connections. Here, each synapse has its own distance parameter, and the flexible switch controls whether it participates in the optimization process, where "True" means that the $d_{i,j}$ is considered a parameter that requires optimization. In addition, $v_{j,n}$ represents the connection strength between the j-th dendrite and n-th membrane, which increases the flexibility of the dendrites. Introducing synaptic and dendritic flexibility allows the proposed DNN to acquire stronger nonlinear capabilities than the conventional DNM when handling complex problems.

B. Dropout Mechanism

In ML, when a model involves too many parameters and too few samples to learn, the trained model is prone to overfitting [56]. As a common problem of ML, overfitting will likely result in unusable models. An ensemble of models is generally employed, which means that multiple models are combined for training to solve this problem. However, training and testing multiple models is time consuming. The dropout mechanism effectively mitigates overfitting by reducing the interactions of the feature detection units, thereby improving the neural network's performance [57], [58]. In the hidden layer of other feedforward neural networks, the local neuron is essentially a weighted summation of the neurons in the preceding hidden layer. The standard dropout mechanism randomly discards different neurons in the hidden layer, which is similar to shaping different networks. The dropout operation enables the neural network to perform a weighted summation of many different neural networks. Different overfitting is generated in these networks, and the "opposite fitting" can counteract each other to reduce the overall overfitting. In addition, the retained weights must be rescaled during training to reduce the impact of neuron dropout on the input of the next hidden layer.

Differing from the overfitting problem in traditional ML techniques, gradient vanishing is an important factor that limits the performance of the proposed DNN. As discussed in Section II-B, the trained synapses can evolve into four synaptic states, in which direct and inverse connections are considered valid, whereas constant 1 and constant 0 connections are considered invalid. Due to the cumulative multiplicative function in the dendritic layer, even a single synapse with a constant 0 connection on the dendrite can result in the degeneration of the connected dendrite. In addition, the sigmoid function in the synapse limits the output of the synaptic layer to between 0 and 1, which further exacerbates the gradient vanishing as the number of features increases. Thus, the proposed DNN includes a dropout mechanism, which is described as follows:

$$\widetilde{S}_{i,j} = \begin{cases} 1, & rand < p \\ S_{i,j}, & otherwise \end{cases}, \tag{13}$$

where p indicates the dropout rate.

As shown in Fig. 4, the dropout mechanism is utilized as an adjustable mechanism to train the proposed DNN. Specifically, the gradient vanishing problem can be reduced significantly by ignoring some of the synapses during training (i.e., setting the value of some synapses to 1). Due to the cumulative multiplication in dendrites, the dropout mechanism in the proposed DNN does not require the rescaling of weights, and the preserved synapses directly inherit the original values. The dropout mechanism causes synapses to not always work in the DNN; thus, parameter updates are no longer dependent on the interaction between synapses with fixed relationships, which prevents some features from only being effective under certain conditions. In addition, eliminating synapses with a constant 0 connection state makes it possible for dendrites to avoid degeneration, which allows them to contribute more to the DNN. The dropout mechanism enables the proposed DNN's sensitivity to some particular cue fragments and to extract information from other cues even if a particular cue is lost. As a result, the proposed DNN is forced to learn more robust features.

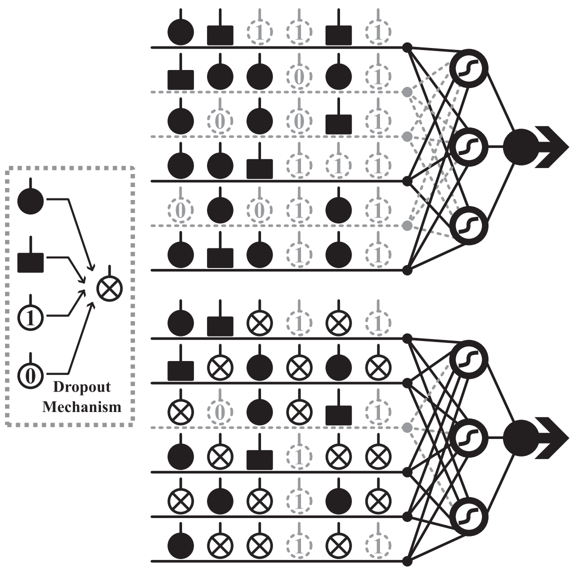

Fig. 4. Dropout mechanism in proposed DNN. The introduction of the Dropout mechanism enables more dendrites to escape degradation.

C. Learning Algorithm

In this study, the proposed DNN is employed as a multiclass classifier; thus, the cross-entropy function is adopted as the loss function, which can be formulated as follows:

$$E = -\sum_{n=1}^{N} T_n \cdot \log\left(\frac{e^{O_n}}{\sum_{n=1}^{N} e^{O_n}}\right), \tag{14}$$

where T_n and O_n denote the target and actual outputs of the n-th soma body, respectively. According to the error, the synaptic and dendritic parameters can be optimized according to the Widrow-Hoff delta learning rule, which is expressed as follows:

$$d_{i,j}(t+1) = d_{i,j}(t) - \eta \triangle \frac{\partial E}{\partial d_{i,j}}, \tag{15}$$

$$w_{i,j}(t+1) = w_{i,j}(t) - \eta \triangle \frac{\partial E}{\partial w_{i,j}}, \tag{16}$$

$$b_{i,j}(t+1) = b_{i,j}(t) - \eta \triangle \frac{\partial E}{\partial b_{i,j}}, \tag{17}$$

$$v_{j,n}(t+1) = v_{j,n}(t) - \eta \triangle \frac{\partial E}{\partial v_{j,n}}, \tag{18}$$

where η indicates the learning rate parameter. Following the chain rule in calculus [59], the partial differential derivatives in the above equations can be decomposed as follows:

$$\frac{\partial E}{\partial d_{i,j}} = \frac{\partial E}{\partial O_n}\frac{\partial O_n}{\partial M_n}\frac{\partial M_n}{\partial D_j}\frac{\partial D_j}{\partial S_{i,j}}\frac{\partial S_{i,j}}{\partial d_{i,j}}, \tag{19}$$

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial O_n}\frac{\partial O_n}{\partial M_n}\frac{\partial M_n}{\partial D_j}\frac{\partial D_j}{\partial S_{i,j}}\frac{\partial S_{i,j}}{\partial w_{i,j}}, \tag{20}$$

$$\frac{\partial E}{\partial b_{i,j}} = \frac{\partial E}{\partial O_n}\frac{\partial O_n}{\partial M_n}\frac{\partial M_n}{\partial D_j}\frac{\partial D_j}{\partial S_{i,j}}\frac{\partial S_{i,j}}{\partial b_{i,m}}, \tag{21}$$

$$\frac{\partial E}{\partial v_{j,n}} = \frac{\partial E}{\partial O_n}\frac{\partial O_n}{\partial M_n}\frac{\partial M_n}{\partial v_{j,n}}, \tag{22}$$

$$\frac{\partial E}{\partial O_n} = O_n - T_n, \tag{23}$$

$$\frac{\partial O_n}{\partial M_n} = \lambda \cdot O_n \cdot (1 - O_n), \tag{24}$$

$$\frac{\partial M_n}{\partial D_j} = v_{j,n}, \tag{25}$$

$$\frac{\partial M_n}{\partial v_{j,n}} = D_j, \tag{26}$$

$$\frac{\partial D_j}{\partial S_{i,j}} = \frac{D_j}{S_{i,j}}, \tag{27}$$

$$\frac{\partial S_{i,j}}{\partial w_{i,j}} = x_i \cdot S_{i,j} \cdot (1 - S_{i,j}), \tag{28}$$

$$\frac{\partial S_{i,j}}{\partial b_{i,j}} = -S_{i,j} \cdot (1 - S_{i,j}). \tag{29}$$

In addition, as an enhanced optimization strategy of adaptive moment estimation (i.e., the Adam optimizer) [60], the AdamW optimizer is implemented to reduce intrinsic fluctuations in the learning process and accelerate the convergence speed [61], which can be described as follows:

$$g_\xi(t+1) = \lambda\xi(t) + \nabla F\left(\xi(t)\right), \tag{30}$$

where ξ is the parameter to be optimized. Here, ∇F and λ denote the corresponding partial differential derivatives and weight decays, respectively. The new gradient update formulas of the AdamW optimizer depending on the momentum m_ξ and velocity v_ξ are expressed as follows:

$$m_\xi(t+1) = \beta_1 \cdot m_\xi(t) + (1-\beta_1) \cdot g_\xi(t+1), \tag{31}$$

$$v_\xi(t+1) = \beta_2 \cdot v_\xi(t) + (1-\beta_2) \cdot g_\xi^2(t+1), \tag{32}$$

$$\widehat{m}_\xi(t+1) = \frac{m_\xi(t+1)}{1-\beta_1^{t+1}}, \tag{33}$$

$$\widehat{v}_\xi(t+1) = \frac{v_\xi(t+1)}{1-\beta_2^{t+1}}. \tag{34}$$

$$\xi(t+1) = \xi(t) - \eta_{t+1}\left(\frac{\alpha\widehat{m}_\xi(t+1)}{\sqrt{\widehat{v}_\xi(t+1)}+\varepsilon} + \lambda\xi(t)\right), \tag{35}$$

where β_1 and β_2 are two constants that are generally set to 0.9 and 0.999, respectively. Note that the initial values of $m_\xi(0)$ and $v_\xi(0)$ are set to 0. Here, ε represents an infinitesimally small positive quantity.

TABLE I
DESCRIPTION OF THE DATASETS

Dataset	Samples	Features (I)	Classes (N)
Balance	625	4	3
CHD	297	13	5
CMC	1473	9	3
DB	13611	16	7
Firewall	65532	11	4
HCV	589	12	5
Iris	150	4	3
Seed	210	7	3
Thyroid	215	5	3
Wine	178	14	3

IV. EXPERIMENT

A. Experimental Setup

To evaluate the performance of the proposed DNN comprehensively, 10 multiclass classification and 2 high-dimensional binary classification datasets from the UCI Machine Learning Repository (https://archive.ics.uci.edu/) were used in our experiments, including the Balance Scale (Balance), Cleveland Heart Disease (CHD), Contraceptive Method Choice (CMC), Dry Bean (DB), Internet Firewall (Firewall), Hepatitis C Virus (HCV), Iris, Seed, Thyroid Disease (Thyroid), and Wine datasets [62]. The details of these datasets are given in Table I. In these experiments, 70% of the samples from each dataset were used for training, and the remaining samples were used to test the performance of the compared models.

Note that there are four adjustable user-predefined parameters in the proposed DNN, i.e., the number of dendrites, dropout rate, flexible switch, and batch size. Thus, sixteen DNNs with different parameter settings were employed for parameter sensitivity analysis, and EDNMs with the One-vs-Rest scheme were adopted to measure the improvement realized by the proposed DNN. In addition, the proposed DNN was compared to 10 existing ML methods to demonstrate its effectiveness. Here, each experiment was run 20 times independently, and the results are presented as the mean and standard deviation to emphasize the confidence of the results.

B. Performance Metrics

For common classification metrics were used to measure the performance of the classifiers: accuracy, precision, F1 score, and Cohen's kappa (κ). The corresponding metrics for each label were calculated, and they were weighted and summed according to the proportion of their sample size in the total sample size, which is expressed as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}, \tag{36}$$

$$\begin{aligned}\text{Precision} = {} & \text{Precision}_1 \cdot w_1 + \text{Precision}_2 \cdot w_2 + \cdots \\ & + \text{Precision}_N \cdot w_N,\end{aligned} \tag{37}$$

$$\text{F1} = \text{F1}_1 \cdot w_1 + \text{F1}_2 \cdot w_2 + \cdots + \text{F1}_N \cdot w_N, \tag{38}$$

$$\kappa = \{2 \cdot (TP \cdot TN - FN \cdot FP)\}/\{(TP + FP) \cdot (FP + TN) + (TP + FN) \cdot (FN + TN)\}, \quad (39)$$

where the corresponding performance metrics for each label are calculated as follows:

$$\text{Precision}_N = \frac{TP_N}{TP_N + FP_N}, \quad (40)$$

$$\text{F1}_N = \frac{2 \cdot \text{Precision}_\text{N} \cdot \text{Recall}_\text{N}}{\text{Precision}_\text{N} + \text{Recall}_\text{N}}, \quad (41)$$

$$\text{Recall} = \text{Recall}_1 \cdot w_1 + \text{Recall}_2 \cdot w_2 + \cdots + \text{Recall}_N \cdot w_N, \quad (42)$$

$$\text{Recall}_N = \frac{TP_N}{TP_N + FN_N}. \quad (43)$$

Here, TP and TN represent true positive and true negative, and FP and FN denote false positive and false negative, respectively.

In addition, nonparametric statistical tests were conducted to verify the differences between the methods. Here, the Friedman test was applied to compare and rank DNNs with different parameters, where the significance level α was set to 0.05 [63]. In addition, the Bonferroni-Dunn procedure was employed as a post hoc test to characterize the statistical results, which can compensate for the lack of controlling family-wise error rates for unadjusted p values [64]. The Wilcoxon signed-rank test was utilized to determine whether significant differences could be observed between the proposed DNN and the compared ML methods, where the significance level α was also set to 0.05 [63].

C. Comparison of DNNs

Based on previous studies, the number of dendrites in the DNM is typically set close to the number of input features (I) [65], and the number of dendrites in EDNMs is $I \times N$. To measure the performance of DNNs systematically, the number of dendrites in DNNs was set to increment from I to $I \times N$. Considering that the DNNs in our experiments are not deep DNNs with a large number of dendrites, the dropout rate was set to two levels, i.e., 0 and 0.05. In addition, the flexible switch also has two states, i.e., true and false. In these experiments, batch size was adaptive, the parameters were modified 200 times in each iteration, and the batch size was set to 1 when the number of samples from the dataset was less than 200. The number of epochs was set to 1000 for all methods. Additional details about the parameter settings are given in Table II.

The accuracy, precision, F1 score, and κ results of the compared DNNs with different parameter settings obtained the 10 datasets are shown in Tables SI and SII, where the best results for each metric are shown in bold. As can be seen, the DNNs achieved competitive results in most cases, and DNNs with more dendrites obtained better performance. The increased flexibility generally improves the performance of DNNs, which allowed them to outperform the original DNNs on most datasets. The performance of the DNNs incorporating only the dropout mechanism was enhanced with an increasing number of dendrites, which is in line with expectations. The DNNs with both high flexibility and the dropout mechanism have a more complex neural network structure; thus, their performance was slightly unsatisfactory with fewer dendrites. Theoretically, the performance of DNNs can be improved further by adjusting parameter settings, e.g., increasing the number of dendrites and epochs to realize sufficient learning. From these results, we conclude that the proposed DNN can solve multiclass classification tasks. In addition, introducing flexibility effectively improves the nonlinear capability of the proposed DNN, thereby enabling it to solve complex tasks efficiently.

TABLE II
PARAMETER SETTINGS OF DNNS

Model	No. of Dendrites	Dropout	Flexible
DNN-1	I	0	False
DNN-2	$I \cdot (N+2)/3$	0	False
DNN-3	$I \cdot (2N+1)/3$	0	False
DNN-4	$I \cdot N$	0	False
DDNN-1	I	0.05	False
DDNN-2	$I \cdot (N+2)/3$	0.05	False
DDNN-3	$I \cdot (2N+1)/3$	0.05	False
DDNN-4	$I \cdot N$	0.05	False
FDNN-1	I	0	True
FDNN-2	$I \cdot (N+2)/3$	0	True
FDNN-3	$I \cdot (2N+1)/3$	0	True
FDNN-4	$I \cdot N$	0	True
DFDNN-1	I	0.05	True
DFDNN-2	$I \cdot (N+2)/3$	0.05	True
DFDNN-3	$I \cdot (2N+1)/3$	0.05	True
DFDNN-4	$I \cdot N$	0.05	True

In addition, the statistical results of DNNs in terms of each performance metric are summarized in Table III . As can be seen, the FDNN-4 model obtained the best performance in terms of accuracy and precision, and the FDNN-2 model achieved the best results in terms of the F1 score and κ metrics. For the accuracy and precision metrics, the FDNN-4 model was significantly superior to the DDNN-1, DDNN-2, and DFDNN-1 models, and the FDNN-2 model outperformed the DDNN-1, DDNN-2, and DFDNN-1 models significantly in terms of F1 score and κ. There was no significant difference between the DNNs with other parameter settings and those with the best performance. The relatively optimal parameter settings for each dataset are listed in Table IV . The dropout mechanism can lead to the DNN underfitting in some simple tasks; however, it can provide a solution to prevent overfitting when solving more complex tasks. Theoretically, the underfitting problem can be solved by setting the number of epochs appropriately, while increasing the number of dendrites and setting suitable dropout parameters can prevent overfitting. Determining appropriate parameter settings for different tasks can improve the performance of the proposed DNN effectively. In summary, the competitive results obtained on most datasets suggest that the DNNs can solve numerous multiclass classification tasks without relying on hyperparameter settings, and the fine hyperparameter design provides the potential to further improve the performance of the proposed DNN.

TABLE III
STATISTICAL RESULTS OF DNNs

Model	Accuracy				Precision			
	Rank	$z\text{-}value$	unadjusted p	p_{Bonf}	Rank	$z\text{-}value$	unadjusted p	p_{Bonf}
DNN-1	9	1.737772	0.082251	1.233764	9.4	1.972607	**0.04854**	0.728106
DNN-2	7.7	1.127204	0.259656	3.894845	8.3	1.455971	0.145401	2.181008
DNN-3	7.1	0.845403	0.397886	5.96829	7.3	0.986303	0.323984	4.859765
DNN-4	6.4	0.516635	0.605411	9.081164	6.4	0.563602	0.573025	8.595377
DDNN-1	12.3	3.287678	**0.00101**	**0.015153**	12.2	3.287678	**0.00101**	**0.015153**
DDNN-2	11.9	3.09981	**0.001936**	**0.029047**	11.6	3.005877	**0.002648**	**0.039722**
DDNN-3	7.8	1.174171	0.240327	3.604901	7.3	0.986303	0.323984	4.859765
DDNN-4	6.8	0.704502	0.48112	7.2168	6.9	0.798436	0.424618	6.369263
FDNN-1	9.4	1.92564	0.054149	0.812241	9.2	1.878673	0.060289	0.904338
FDNN-2	5.7	0.187867	0.850981	12.76471	5.7	0.234834	0.814337	12.215062
FDNN-3	5.6	0.1409	0.887949	13.319229	5.6	0.187867	0.850981	12.76471
FDNN-4	**5.3**	-	-	-	**5.2**	-	-	-
DFDNN-1	12.3	3.287678	**0.00101**	**0.015153**	12.4	3.381611	**0.000721**	**0.010809**
DFDNN-2	11.5	2.911943	**0.003592**	0.053878	10.9	2.677109	**0.007426**	0.111391
DFDNN-3	8.6	1.549905	0.121164	1.817464	8.7	1.643839	0.10021	1.503143
DFDNN-4	8.6	1.549905	0.121164	1.817464	8.9	1.737772	0.082251	1.233764

Model	F1				κ			
	Rank	$z\text{-}value$	unadjusted p	p_{Bonf}	Rank	$z\text{-}value$	unadjusted p	p_{Bonf}
DNN-1	9.1	2.06654	**0.038778**	0.581663	9.1	2.06654	**0.038778**	0.581663
DNN-2	8	1.549905	0.121164	1.817464	8	1.549905	0.121164	1.817464
DNN-3	7.9	1.502938	0.132855	1.992824	7.9	1.502938	0.132855	1.992824
DNN-4	6	0.610569	0.541485	8.122277	6	0.610569	0.541485	8.122277
DDNN-1	12.2	3.522512	**0.000427**	**0.006412**	12.2	3.522512	**0.000427**	**0.006412**
DDNN-2	11.7	3.287678	**0.00101**	**0.015153**	11.7	3.287678	**0.00101**	**0.015153**
DDNN-3	7.2	1.174171	0.240327	3.604901	7.2	1.174171	0.240327	3.604901
DDNN-4	7.2	1.174171	0.240327	3.604901	7.2	1.174171	0.240327	3.604901
FDNN-1	9.4	2.207441	**0.027283**	0.409249	9.4	2.207441	**0.027283**	0.409249
FDNN-2	**4.7**	-	-	-	**4.7**	-	-	-
FDNN-3	6.1	0.657536	0.510837	7.66255	6.1	0.657536	0.510837	7.66255
FDNN-4	5.6	0.422701	0.672513	10.087697	5.6	0.422701	0.672513	10.087697
DFDNN-1	12.5	3.663412	**0.000249**	**0.003733**	12.5	3.663412	**0.000249**	**0.003733**
DFDNN-2	10.7	2.818009	**0.004832**	0.072484	10.7	2.818009	**0.004832**	0.072484
DFDNN-3	8.9	1.972607	**0.04854**	0.728106	8.9	1.972607	**0.04854**	0.728106
DFDNN-4	8.8	1.92564	0.054149	0.812241	8.8	1.92564	0.054149	0.812241

TABLE IV
OPTIMAL PARAMETER SETTINGS OF DNNs

Dataset	Dendrites	Dropout	Flexible
Balance	10	0	True
CHD	48	0.05	True
CMC	27	0.05	True
DB	81	0	True
Firewall	44	0	False
HCV	45	0	False
Iris	4	0	False
Seed	16	0	True
Thyroid	12	0	True
Wine	23	0	True

TABLE V
PARAMETER INITIALIZATION RANGES OF THE ML METHODS

Method	Parameters	Range
EDTs	Number of DTs	N
	Min Leaf Size	auto
	Max Tree Depth	auto
EeSVMs	Type of SVMs	epsilon SVM
	Number of SVMs	N
	Kernel	linear, polynomial, sigmoid, radial basis function
EnSVMs	Type of SVMs	nu SVM
	Number of SVMs	N
	Kernel	linear, polynomial, sigmoid, radial basis function
MLP	Hidden Layer	I
	Learning Rate	0.01
	epochs	1000
EDNMs	Number of DNMs	N
	Learning Rate	0.01
	epochs	1000

D. Comparison of ML Methods

To further validate the classification performance of the proposed DNN, EDNMs, and 10 ML methods were compared experimentally, including ensemble decision trees using the One-vs-Rest scheme, ensemble epsilon support vector machines (EeSVMs), and ensemble nu support vector machines (EnSVMs) with different kernel functions using the One-vs-Rest scheme, i.e., EDTs, EeSVMs-L (with a linear kernel), EeSVMs-P (with a polynomial kernel), EeSVMs-R (with a radial basis function kernel), EeSVMs-S (with a sigmoid kernel), EnSVMs-L (with a linear kernel), EnSVMs-P (with a polynomial kernel), EnSVMs-R (with a radial basis function kernel), EnSVMs-S (with a sigmoid kernel), and the multilayer perceptron (MLP). The hyperparameter settings of all methods are listed in Table V.

Tables SIII and SIV compare the accuracy, precision, F1 score, and κ results of the DNN, EDNMs, and ML methods obtained on the 10 experimental datasets. As can be seen, the proposed DNN achieved the best results in nearly all performance metrics on all datasets, with the exception of the accuracy metric on the CHD dataset and all metrics on the Firewall dataset. On

TABLE VI
STATISTICAL RESULTS OF THE COMPARED ML METHODS

Model	Accuracy				Precision			
	Rank	z-$value$	unadjusted p	p_{Bonf}	Rank	z-$value$	unadjusted p	p_{Bonf}
EDTs	6.3	3.162886	**0.001562**	**0.017183**	5.6	2.790782	**0.005258**	**0.057839**
EeSVMs-L	6.2	3.100868	**0.00193**	**0.021225**	6.3	3.224903	**0.00126**	**0.013862**
EeSVMs-P	9.4	5.085424	**0**	**0.000004**	9.5	5.209459	**0**	**0.000002**
EeSVMs-R	5.8	2.852799	**0.004334**	**0.04767**	5.9	2.976834	**0.002912**	**0.032037**
EeSVMs-S	8.3	4.403233	**0.000011**	**0.000117**	8.7	4.71332	**0.000002**	**0.000027**
EnSVMs-L	7.2	3.721042	**0.000198**	**0.002182**	8.2	4.403233	**0.000011**	**0.000117**
EnSVMs-P	10.1	5.519546	**0**	**0**	10.2	5.64358	**0**	**0**
EnSVMs-R	6.75	3.441964	**0.000578**	**0.006353**	7.6	4.031129	**0.000056**	**0.000611**
EnSVMs-S	8.75	4.682311	**0.000003**	**0.000031**	8.7	4.71332	**0.000002**	**0.000027**
MLP	4.9	2.294643	**0.021754**	0.23929	3.8	1.674469	0.094038	1.034423
EDNMs	3.1	1.17833	0.238665	2.625316	2.4	0.806226	0.420113	4.621239
DNN	**1.2**	-	-	-	**1.1**	-	-	-
Model	**F1**				**κ**			
	Rank	z-$value$	unadjusted p	p_{Bonf}	Rank	z-$value$	unadjusted p	p_{Bonf}
EDTs	5.7	2.852799	**0.004334**	**0.04767**	5.7	2.852799	**0.004334**	**0.04767**
EeSVMs-L	6.4	3.28692	**0.001013**	**0.011142**	6.5	3.348938	**0.000811**	**0.008923**
EeSVMs-P	10.1	5.581563	**0**	**0**	10.1	5.581563	**0**	**0**
EeSVMs-R	6	3.038851	**0.002375**	**0.026123**	6.1	3.100868	**0.00193**	**0.021225**
EeSVMs-S	8.4	4.527268	**0.000006**	**0.000066**	8.5	4.589285	**0.000004**	**0.000049**
EnSVMs-L	7.6	4.031129	**0.000056**	**0.000611**	7.7	4.093146	**0.000043**	**0.000468**
EnSVMs-P	10.3	5.705598	**0**	**0**	10.2	5.64358	**0**	**0**
EnSVMs-R	7.3	3.845077	**0.000121**	**0.001326**	7.1	3.721042	**0.000198**	**0.002182**
EnSVMs-S	8.9	4.837355	**0.000001**	**0.000014**	8.7	4.71332	**0.000002**	**0.000027**
MLP	3.8	1.674469	0.094038	1.034423	3.8	1.674469	0.094038	1.034423
EDNMs	2.4	0.806226	0.420113	4.621239	2.5	0.868243	0.385261	4.237874
DNN	**1.1**	-	-	-	**1.1**	-	-	-

the Firewall dataset, the EDTs achieved the best results for all performance metrics, with DNNs ranking second for all metrics. In addition, the statistical results of the Wilcoxon signed-rank test are shown in Tables SIII and SIV, where "-" indicates that the DNN cannot be compared with itself. As shown, the proposed DNN significantly outperformed the vast majority of the compared ML methods in terms of nearly all metrics on all datasets. Specifically, the proposed DNN significantly outperformed all other ML methods in terms of accuracy, precision, F1 score, and κ on the Balance, CMC, DB, and Seed datasets. On the CHD dataset, the proposed DNN and EeSVMs-S achieved comparable results, and the DNN was significantly better than all other ML methods in terms of the other performance metrics. On the Firewall dataset, the proposed DNN significantly outperformed all other ML methods (except the EDTs) in terms of all performance metrics. In addition, we found that the EDNMs achieved similar results to the proposed DNN on the HCV, Iris, Thyroid, and Wine datasets, and EnSVMs-S obtained results that were comparable to those of the proposed DNN on the Wine dataset. No significant difference was observed between the proposed DNN and EnSVMs-L in terms of the precision and κ metrics on the Wine dataset. Thus, we conclude that the proposed DNN has a more appropriate neural network structure that can utilize the dendritic information more effectively than a simple ensemble of DNMs. Introducing flexibility and the dropout mechanism enables the proposed DNN to solve complex tasks more effectively. Compared with many other multiclass classifiers based on ML methods, the excellent multiclass classification capabilities of the proposed DNN demonstrate that it is a promising multiclass classifier.

The statistical results of the Friedman test are summarized in Table VI, and the results show that the proposed DNN ranked first among the compared ML methods. The proposed DNN is significantly superior to the EDTs, EeSVMs-L, EeSVMs-P, EeSVMs-R, EeSVMs-S, EnSVMs-L, EnSVMs-P, EnSVMs-R, EnSVMs-S, and MLP models in terms of the accuracy, precision, F1 score, and κ metrics. Benefiting from having more dendrites, the EDNMs achieved performance that is comparable to that of the proposed DNN, and no significant difference was observed between the EDNMs and DNN. The MLP model also obtained competitive performance because it has more hidden layers. Based on these results, we conclude that, compared to existing ML methods, the proposed DNN is an excellent neural network. Ensemble models that combine binary classification models as units significantly limit the performance of numerous ML methods in terms of solving multiclass classification problems. Although EDNMs can provide competitive performance on some datasets, the flexible and concise architectural design of the proposed DNN provides better computational efficiency.

E. Extension

In this section, we investigate the performance of the proposed DNN when applied to binary classification datasets with more features. The Breast dataset contains 569 samples with 30 features, and the Parkinson dataset consists of 195 samples, each of which contains 22 features. A comparison of the accuracy, precision, F1 score, and κ results obtained by the proposed DNN, EDNMs, and nine ML methods on 2 datasets is shown in Table VII. In this evaluation, the number of output layers was set to two for the MLP. In addition, two individual units were utilized to collaborate in the ensemble models. As shown in Table VII, the proposed DNN obtained better results than the compared ML methods in terms of accuracy, precision, and F1

TABLE VII
COMPARISON OF RESULTS ON BREAST AND PARKINSON DATASETS

	Model	Accuracy (%)	*p-value*	Precision (%)	*p-value*	F1 (%)	*p-value*	κ (%)	*p-value*
Breast	EDTs	91.72±2.35	**1.31e-06**	91.83±2.35	**1.01e-06**	91.72±2.35	**1.01e-06**	91.72±2.36	**1.17e-03**
	EeSVMs-L	94.11±1.59	**2.88e-06**	94.63±1.32	**1.67e-06**	94.11±1.59	**1.67e-06**	93.99±1.64	4.75e-01
	EeSVMs-P	68.95±3.41	**8.85e-07**	79.31±1.34	**9.13e-07**	68.95±3.41	**9.12e-07**	60.73±4.84	**9.13e-07**
	EeSVMs-R	94.35±1.69	**2.53e-06**	94.85±1.40	**1.85e-06**	94.35±1.69	**1.85e-06**	94.23±1.76	4.19e-01
	EeSVMs-S	94.27±1.72	**8.78e-07**	94.80±1.41	**9.13e-07**	94.27±1.72	**9.13e-07**	94.15±1.78	5.49e-01
	EnSVMs-L	94.78±1.31	**1.87e-06**	95.14±1.13	**1.01e-06**	94.78±1.31	**1.24e-06**	94.68±1.35	8.53e-01
	EnSVMs-P	88.91±2.54	**8.91e-07**	90.56±1.90	**9.13e-07**	88.91±2.54	**9.13e-07**	88.39±2.74	**4.03e-06**
	EnSVMs-R	95.30±1.39	**1.29e-05**	95.62±1.23	**1.03e-05**	95.30±1.39	**1.13e-05**	**95.23±1.44**	9.66e-01
	EnSVMs-S	93.96±1.76	**1.92e-06**	94.52±1.45	**1.24e-06**	93.96±1.76	**1.24e-06**	93.82±1.84	3.56e-01
	MLP	63.02±2.82	**9.03e-07**	39.79±3.53	**9.13e-07**	48.76±3.50	**9.13e-07**	0.00±0.00	**9.12e-07**
	EDNMs	96.57±1.40	**1.40e-02**	96.62±1.35	**6.79e-03**	96.55±1.41	**1.18e-02**	92.65±2.90	**1.06e-02**
	DNN	**97.35±1.15**	-	**97.42±1.09**	-	**97.33±1.17**	-	94.19±2.61	-
Parkinson	EDTs	85.08±4.59	**1.97e-05**	86.61±4.34	**7.09e-05**	85.08±4.59	**2.12e-05**	**85.30±4.55**	1.00
	EeSVMs-L	84.12±3.63	**8.93e-06**	85.44±4.47	**2.33e-05**	84.12±3.63	**1.36e-05**	81.63±4.52	9.97e-01
	EeSVMs-P	75.31±4.97	**8.52e-07**	56.96±7.43	**9.12e-07**	75.31±4.97	**9.12e-07**	64.79±6.68	**2.02e-04**
	EeSVMs-R	81.53±6.32	**2.81e-06**	81.08±13.42	**3.66e-06**	81.53±6.32	**2.25e-06**	76.69±9.65	8.71e-01
	EeSVMs-S	76.33±4.59	**8.65e-07**	61.47±12.18	**9.12e-07**	76.33±4.59	**9.12e-07**	66.52±6.80	**2.75e-04**
	EnSVMs-L	85.82±3.49	**4.85e-05**	87.07±3.13	**3.30e-05**	85.82±3.49	**1.16e-04**	83.78±4.54	1.00
	EnSVMs-P	78.76±7.67	**3.54e-06**	66.27±17.36	**1.67e-06**	78.76±7.67	**2.74e-06**	70.73±12.18	1.01e-01
	EnSVMs-R	85.65±4.15	**1.38e-05**	86.24±7.01	**1.95e-05**	85.65±4.15	**3.03e-05**	83.37±5.93	1.00
	EnSVMs-S	80.17±8.08	**2.05e-05**	72.24±18.55	**3.30e-05**	80.17±8.08	**3.30e-05**	73.76±12.99	4.11e-01
	MLP	85.93±7.99	**7.72e-03**	83.29±14.90	**1.32e-02**	83.69±11.61	**1.12e-02**	56.26±30.63	**7.61e-03**
	EDNMs	88.31±4.39	**1.46e-02**	89.16±4.10	**1.01e-02**	87.22±5.07	**3.75e-03**	64.41±11.55	**6.21e-04**
	DNN	**90.79±3.66**	-	**91.53±3.44**	-	**90.65±3.90**	-	74.63±10.02	-

score. On the Breast dataset, the EnSVMs-R model achieved the best results, and the proposed DNN was slightly inferior in terms of κ. These statistical results suggest that the proposed DNN significantly outperformed the EDTs, EeSVMs-L, EeSVMs-P, EeSVMs-R, EeSVMs-S, EnSVMs-L, EnSVMs-P, EnSVMs-R, EnSVMs-S, MLP, and EDNMs models on the Breast and Parkinson datasets in terms of accuracy, precision, and F1 score. Thus, we conclude that introducing flexibility and the dropout mechanism enhances the proposed DNN's ability to solve multiclass classification problems and improves its performance on binary classification datasets with more features.

V. CONCLUSION

In this paper, we have proposed the DNN architecture, which extends the single-neuron model of the conventional DNM to a feedforward neural network structure that can process multiple inputs and produce multiple outputs. The added flexibility enhances the adaptability of the model to different loss functions and enables the construction of deep neural networks. The proposed DNN architecture represents a more appropriate neural network structure than a simple ensemble of DNMs because it enables more effective utilization of dendritic information. In addition, the synaptic flexibility enhances its nonlinear capability, thereby making it more efficient when solving complex tasks. Although introducing the dropout mechanism may result in underfitting on some simpler tasks, it allows the proposed DNN to prevent overfitting in more complex tasks. To address underfitting, it is theoretically possible to increase the amount of learning. In contrast, overfitting can be prevented by increasing the number of dendrites and setting appropriate dropout parameters. Thus, designing appropriate parameter settings for different tasks is crucial in terms of improving the performance of the proposed DNN. To evaluate the effectiveness of the proposed DNN architecture, we applied it to 10 multiclass classification and 2 high-dimensional binary classification problems. Compared with representative multiclass classification methods, the proposed DNN exhibited competitive classification performance and satisfactory computational efficiency. These results suggest that the proposed DNN is a promising ANN with practical application potential in various classification tasks. In future research, we plan to investigate the application of the proposed DNN to other more complex problems and design strategies to further optimize its performance. In addition, the effects of various neural network parameter settings on the performance of the proposed DNN in various tasks will be verified to analyze the architectural differences between it and other neural networks.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] M. A. Nielsen, *Neural Networks and Deep Learning*, vol. 25. San Francisco, CA, USA: Determination Press, 2015.

[3] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.

[4] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.

[5] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, 1958, Art. no. 386.

[6] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Circuits Syst.*, vol. 8, no. 7, pp. 579–588, 2009.

[7] M. Minsky and S. Papert, "An introduction to computational geometry," *Cambridge Tiass., HIT*, vol. 479, 1969, Art. no. 480.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] R. Mira, K. Vougioukas, P. Ma, S. Petridis, B. W. Schuller, and M. Pantic, "End-to-end video-to-speech synthesis using generative adversarial networks," *IEEE Trans. Cybern.*, vol. 53, no. 6, pp. 3454–3466, Jun. 2023.

[10] Q. Song, B. Sun, and S. Li, "Multimodal sparse transformer network for audio-visual speech recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10028–10038, Dec. 2023.

[11] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing CNN architectures using the genetic algorithm for image classification," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3840–3854, Sep. 2020.
[12] N. K. Tomar et al., "FANet: A feedback attention network for improved biomedical image segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9375–9388, Nov. 2023.
[13] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 604–624, Feb. 2021.
[14] H. Sakai and H. Iiduka, "Riemannian adaptive optimization algorithm and its application to natural language processing," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7328–7339, Aug. 2021.
[15] H. Qiao, J. Chen, and X. Huang, "A survey of brain-inspired intelligent robots: Integration of vision, decision, motion control, and musculoskeletal systems," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 11267–11280, Oct. 2022.
[16] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas, "Reinforcement learning for mobile robotics exploration: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 3796–3810, Aug. 2023.
[17] L. Squire, D. Berg, F. E. Bloom, S. Du Lac, A. Ghosh, and N. C. Spitzer, *Fundamental Neuroscience*. Cambridge, MA, USA: Academic Press, 2012.
[18] G. Major, M. E. Larkum, and J. Schiller, "Active properties of neocortical pyramidal neuron dendrites," *Annu. Rev. Neurosci.*, vol. 36, pp. 1–24, 2013.
[19] P. Poirazi and B. W. Mel, "Impact of active dendrites and structural plasticity on the memory capacity of neural tissue," *Neuron*, vol. 29, no. 3, pp. 779–796, 2001.
[20] G. Stuart, N. Spruston, and M. Häusser, *Dendrites*. Oxford, U.K.: Oxford Univ. Press, 2016.
[21] N. Spruston, "Pyramidal neurons: Dendritic structure and synaptic integration," *Nature Rev. Neurosci.*, vol. 9, no. 3, pp. 206–221, 2008.
[22] R. C. Malenka and M. F. Bear, "LTP and LTD: An embarrassment of riches," *Neuron*, vol. 44, no. 1, pp. 5–21, 2004.
[23] T. V. Bliss and G. L. Collingridge, "A synaptic model of memory: Long-term potentiation in the hippocampus," *Nature*, vol. 361, no. 6407, pp. 31–39, 1993.
[24] R. Yuste, "Dendritic spines and distributed circuits," *Neuron*, vol. 71, no. 5, pp. 772–781, 2011.
[25] D. Johnston and R. Narayanan, "Active dendrites: Colorful wings of the mysterious butterflies," *Trends Neurosci.*, vol. 31, no. 6, pp. 309–316, 2008.
[26] A. Holtmaat and K. Svoboda, "Experience-dependent structural synaptic plasticity in the mammalian brain," *Nature Rev. Neurosci.*, vol. 10, no. 9, pp. 647–658, 2009.
[27] Z. Tang, H. Tamura, O. Ishizuka, and K. Tanno, "A neuron model with interaction among synapses," *IEEJ Trans. Electron., Inf. Syst.*, vol. 120, no. 7, pp. 1012–1019, 2000.
[28] W. R. Taylor, S. He, W. R. Levick, and D. I. Vaney, "Dendritic computation of direction selectivity by retinal ganglion cells," *Science*, vol. 289, no. 5488, pp. 2347–2350, 2000.
[29] Y. Todo, H. Tamura, K. Yamashita, and Z. Tang, "Unsupervised learnable neuron model with nonlinear interaction on dendrites," *Neural Netw.*, vol. 60, pp. 96–103, 2014.
[30] H. TAMURA, Z. TANG, and M. ISHII, "The neuron model considering difference of time of inputs and its movement direction selection function," *IEEJ Trans. Electron., Inf. Syst.*, vol. 122, no. 7, pp. 1094–1103, 2002.
[31] Y. Todo, Z. Tang, H. Todo, J. Ji, and K. Yamashita, "Neurons with multiplicative interactions of nonlinear synapses," *Int. J. Neural Syst.*, vol. 29, no. 08, 2019, Art. no. 1950012.
[32] J. Ji, S. Gao, J. Cheng, Z. Tang, and Y. Todo, "An approximate logic neuron model with a dendritic structure," *Neurocomputing*, vol. 173, pp. 1775–1783, 2016.
[33] X. Luo, X. Wen, M. Zhou, A. Abusorrah, and L. Huang, "Decision-tree-initialized dendritic neuron model for fast and accurate data classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4173–4183, Sep. 2022.
[34] A. Yilmaz and U. Yolcu, "A robust training of dendritic neuron model neural network for time series prediction," *Neural Comput. Appl.*, pp. 1–20, 2023.
[35] C. Tang, J. Ji, Y. Tang, S. Gao, Z. Tang, and Y. Todo, "A novel machine learning technique for computer-aided diagnosis," *Eng. Appl. Artif. Intell.*, vol. 92, 2020, Art. no. 103627.
[36] Y. Tang, J. Ji, S. Gao, H. Dai, Y. Yu, and Y. Todo, "A pruning neural network model in credit classification analysis," *Comput. Intell. Neurosci.*, vol. 2018, 2018.
[37] J. Ji, S. Song, Y. Tang, S. Gao, Z. Tang, and Y. Todo, "Approximate logic neuron model trained by states of matter search algorithm," *Knowl.-Based Syst.*, vol. 163, pp. 120–130, 2019.
[38] M. Dong, C. Tang, J. Ji, Q. Lin, and K.-C. Wong, "Transmission trend of the COVID-19 pandemic predicted by dendritic neural regression," *Appl. Soft Comput.*, vol. 111, 2021, Art. no. 107683.
[39] Z. Song, Y. Tang, J. Ji, and Y. Todo, "Evaluating a dendritic neuron model for wind speed forecasting," *Knowl.-Based Syst.*, vol. 201, 2020, Art. no. 106052.
[40] Y. Tang, Z. Song, Y. Zhu, M. Hou, C. Tang, and J. Ji, "Adopting a dendritic neural model for predicting stock price index movement," *Expert Syst. With Appl.*, vol. 205, 2022, Art. no. 117637.
[41] C. Tang, Y. Todo, J. Ji, and Z. Tang, "A novel motion direction detection mechanism based on dendritic computation of direction-selective ganglion cells," *Knowl.-Based Syst.*, vol. 241, 2022, Art. no. 108205.
[42] H. He, S. Gao, T. Jin, S. Sato, and X. Zhang, "A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction," *Appl. Soft Comput.*, vol. 108, 2021, Art. no. 107488.
[43] S. Gao et al., "Fully complex-valued dendritic neuron model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2105–2118, Apr. 2023.
[44] X. Luo, L. Ye, X. Liu, X. Wen, M. Zhou, and Q. Zhang, "Interpretability diversity for decision-tree-initialized dendritic neuron model ensemble," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 06, 2023, doi: 10.1109/TNNLS.2023.3290203.
[45] J. Ji, J. Zhao, Q. Lin, and K. C. Tan, "Competitive decomposition-based multiobjective architecture search for the dendritic neural model," *IEEE Trans. Cybern.*, vol. 53, no. 11, pp. 6829–6842, Nov. 2023.
[46] C. Tang, Y. Todo, J. Ji, Q. Lin, and Z. Tang, "Artificial immune system training algorithm for a dendritic neuron model," *Knowl.-Based Syst.*, vol. 233, 2021, Art. no. 107509.
[47] J. Ji, M. Dong, C. Tang, J. Zhao, and S. Song, "A novel plastic neural model with dendritic computation for classification problems," in *Proc. 16th Intell. Comput. Theories Appl.: Int. Conf.*, 2020, pp. 471–483.
[48] C. Tang, Z. Song, Y. Tang, H. Tang, Y. Wang, and J. Ji, "An evolutionary neuron model with dendritic computation for classification and prediction," in *Proc. Intell. Comput. Theories Appl.: 17th Int. Conf.*, 2021, pp. 18–36.
[49] Z. Xu, Z. Wang, J. Li, T. Jin, X. Meng, and S. Gao, "Dendritic neuron model trained by information feedback-enhanced differential evolution algorithm for classification," *Knowl.-Based Syst.*, vol. 233, 2021, Art. no. 107536.
[50] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognit.*, vol. 44, no. 8, pp. 1761–1776, 2011.
[51] Z.-L. Zhang, X.-G. Luo, S. González, S. García, and F. Herrera, "DRCW-ASEG: One-versus-one distance-based relative competence weighting with adaptive synthetic example generation for multi-class imbalanced datasets," *Neurocomputing*, vol. 285, pp. 176–187, 2018.
[52] X. Gao et al., "A multiclass classification using one-versus-all approach with the differential partition sampling ensemble," *Eng. Appl. Artif. Intell.*, vol. 97, 2021, Art. no. 104034.
[53] J. Ji, H. Jia, Y. Ren, and M. Lei, "Supervised contrastive learning with structure inference for graph classification," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1684–1695, May/Jun. 2023.
[54] Z.-H. Zhou, *Machine Learning*. Berlin, Germany: Springer, 2021.
[55] Q. Peng et al., "An extension network of dendritic neurons," *Comput. Intell. Neurosci.*, vol. 2023, 2023.
[56] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*.
[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[58] N. Srivastava, "Improving neural networks with dropout," *Univ. Toronto*, vol. 182, no. 566, 2013, Art. no. 7.
[59] O. H. Rodríguez and J. M. Lopez Fernandez, "A semiotic reflection on the didactics of the chain rule," *Math. Enthusiast*, vol. 7, no. 2, pp. 321–332, 2010.
[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[61] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[62] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.
[63] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.
[64] A. Gasparri, S. Panzieri, and F. Pascucci, "A spatially structured genetic algorithm for multi-robot localization," *Intell. Service Robot.*, vol. 2, pp. 31–40, 2009.
[65] J. Ji, C. Tang, J. Zhao, Z. Tang, and Y. Todo, "A survey on dendritic neuron model: Mechanisms, algorithms and practical applications," *Neurocomputing*, vol. 489, pp. 390–406, 2022.

Cheng Tang (Member, IEEE) received the M.E. and Ph.D. degrees in engineering from the University of Toyama, Toyama, Japan, in 2020 and 2022, respectively. From 2022 to 2023, he was an Assistant Professor with the Nagoya Institute of Technology, Nagoya, Japan. In 2023, he joined Kyushu University, Fukuoka, Japan, where he is currently an Assistant Professor with the Faculty of Information Science and Electrical Engineering, Department of Advanced Information Technology. His research interests include neural networks, evolutionary computation, bioinformatics, and artificial intelligence in medical Big Data.

Junkai Ji (Member, IEEE) received the B.S. degree from the Hefei University of technology, Anhui, China, in 2013, and the M.S. and D.E. degrees from the University of Toyama, Toyama, Japan, in 2016 and 2018, respectively. In 2019, he joined Shenzhen University, Shenzhen, China, where he is currently an Assistant Professor with the National Engineering Laboratory for Big Data System Computing Technology. His research interests include neural networks, evolutionary computation, and computer-aided drug design.

Yuki Todo (Member, IEEE) received the B.S. degree from Zhejiang University, Zhejiang, China, the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, and the D.E. degree from Kanazawa University, Kanazawa, Japan, in 1983, 1986, and 2005, respectively. From 1987 to 1989, she was an Assistant Professor with the Institute of Microelectronics, Shanghai Jiao Tong University, Shanghai, China. From 1989 to 1990, she was a Research Student with Nagoya University, Nagoya, Japan. From 1990 to 2000, she was a Senior Engineer with Sanwa Newtech Inc., Miyazaki, Japan. From 2000 to 2011, she was with Tateyama Systems Institute, Toyama, Japan. In 2012, she joined Kanazawa University, Kanazawa, Japan, where she is currently an Associate Professor with the School of Electrical and Computer Engineering. Her research interests include multiple-valued logic, neural networks, and optimization.

Atsushi Shimada (Member, IEEE) received the D.E. degree from Kyushu University, Fukuoka, Japan, in 2007. He is currently a Professor with the Faculty of Information Science and Electrical Engineering, Kyushu University. From 2015 to 2019, he was a JST-PRESTO Researcher. His research interests include learning analytics, pattern recognition, media processing, and image processing. He was the recipient of the MIRU Interactive Presentation Award (2011, 2017), MIRU Demonstration Award (2015), Background Models Challenge 2012 The First Place (2012), PRMU research award (2013), SBM-RGBD Challenge The First Place (2017), ITS Symposium Best Poster Award (2018), IPSJ/IEEE-Computer Society Young Computer Researcher Award (2019), CELDA Best Paper Award (2019), and MEXT Young Scientist Award (2020).

Weiping Ding (Senior Member, IEEE) received the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013. From 2014 to 2015, he was a Postdoctoral Researcher with the Brain Research Center, National Chiao Tung University, Hsinchu, Taiwan. In 2016, he was a Visiting Scholar with the National University of Singapore, Singapore. From 2017 to 2018, he was a Visiting Professor with the University of Technology Sydney, Ultimo, NSW, Australia. He is currently a Professor with the School of Information Science and Technology, Nantong University, Nantong, China, and also the supervisor of Ph.D. postgraduate by the Faculty of Data Science, City University of Macau, China. His ninthteen authored/coauthored papers have been selected as ESI Highly Cited Papers. He has coauthored four books. He holds 32 approved invention patents, including two U.S. patents and one Australian patent. He has authored or coauthored more than 200 articles in international journals, including more than 110 IEEE journal papers, such as IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. His research interests include deep neural networks, multimodal machine learning, granular data mining, and medical images analysis. He is/was the Editorial Board Member of *Knowledge-Based Systems*, *Engineering Applications of Artificial Intelligence*, and *Applied Soft Computing*. He is an Area Editor of *Information Fusion*. He is an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, *Information Sciences*, and *Neurocomputing*. He is the Leading Guest Editor of Special Issues in several prestigious journals, including IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON FUZZY SYSTEMS, and *Information Fusion*.

Akimasa Hirata (Fellow, IEEE) received the B.E. and M.E. degrees in communications engineering from Osaka University, Suita, Japan, in 1996, 1998, and 2000, respectively. From 1999 to 2001, he was a Research Fellow of the Japan Society for the Promotion of Science, and a Visiting Research Scientist with the University of Victoria, Victoria, BC, Canada, in 2000. In 2001, he joined the Department of Communications Engineering, Osaka University, Osaka, Japan, as an Assistant Professor. In 2004, he was an Associate Professor with the Department of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya, Japan, where he is currently a Full Professor. His research interests include electromagnetic safety, risk management system for heat-related illness, methods in neuroscience, antennas, filters, and related computational techniques. He is an editorial board Member of *Physics in Medicine and Biology*, a Member of the main commission, and the Chair of Project Group of International Commission on Non-Ionizing Radiation Protection, and administrative committee, and a Subcommittee (EMF Dosimetry Modeling) Chair of IEEE International Committee on Electromagnetic Safety, and an expert of World Health Organization. From 2006 to 2012, he was an Associate Editor for IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING. He was the recipient of several awards including Young scientists' Prize (2006) and Prizes for Science and Technology (Research Category in 2011, Public Understanding Promotion Category in 2014, 2020) by the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science, and Technology, Japan, and IEEE EMC-S Technical Achievement Award (2015), Japan Academy Medal and JSPS Prize (2018), and Japan Open Innovation Prize (President of the Science Council of Japan Prize in 2022) from the Cabinet Office. He is a Fellow of Institute of Physics and a Member of IEICE, IEE Japan, and Bioelectromagnetics Society.