An evidential classifier based on Dempster-Shafer theory and deep learning

Zheng Tong [a,*], Philippe Xu [a], Thierry Denœux [a,b,c]

[a] *Université de technologie de Compiègne, CNRS, Heudiasyc, Compiègne, France*
[b] *Shanghai University, UTSEUS, Shanghai, China*
[c] *Institut universitaire de France, Paris, France*

ARTICLE INFO

ABSTRACT

We propose a new classifier based on Dempster-Shafer (DS) theory and a convolutional neural network (CNN) architecture for set-valued classification. In this classifier, called the evidential deep-learning classifier, convolutional and pooling layers first extract high-dimensional features from input data. The features are then converted into mass functions and aggregated by Dempster's rule in a DS layer. Finally, an expected utility layer performs set-valued classification based on mass functions. We propose an end-to-end learning strategy for jointly updating the network parameters. Additionally, an approach for selecting partial multi-class acts is proposed. Experiments on image recognition, signal processing, and semantic-relationship classification tasks demonstrate that the proposed combination of deep CNN, DS layer, and expected utility layer makes it possible to improve classification accuracy and to make cautious decisions by assigning confusing patterns to multi-class sets.

1. Introduction

In machine learning, classification refers to the task of predicting the class of a new sample based on a learning set of labeled instances. The most common classification problem is *precise classification*, in which a sample is classified into one and only one of the possible classes. Unfortunately, such a hard assignment often leads to misclassification in case of high uncertainty. For example, ambiguity occurs when the feature vector does not contain sufficient information to identify a precise class, and multiple classes have similar probabilities. Also, a classifier with only precise classification may fail to identify outliers belonging to class that is not represented in the learning set.

Set-valued classification [20,45,38] is a potential way to solve this problem; it is defined as the assignment of a new observation into a non-empty subset of classes when the uncertainty is too high to make a precise classification. For instance, given a class set $\Omega = \{\omega_1, \omega_2, \omega_3\}$, we may not be able to reliably classify a sample $\boldsymbol{x}$ into a single class, but it may be almost sure that it does not belong to ω_3. In this case, it is more cautious to assign $\boldsymbol{x}$ to the set $\{\omega_1, \omega_2\}$. Classification with a reject option in [4,62] can be

regarded as a special case of set-valued classification, rejection being equivalent to assigning a sample to the entire set of possible classes. A related problem concerns the treatment of outliers, which cannot be classified into any of the known classes, a problem referred to as "novelty detection" or "distance rejection" [16]. Depending on the method, such samples may be assigned to the empty set, or to the whole set Ω, reflecting maximum uncertainty [9]. Set-valued classification makes it possible to better reflect classification uncertainty, increase the cautiousness of classifiers and ultimately reduce the error rate. Precise classification can be considered as a special case of set-valued classification, in which only the sets with one class are considered.

In this study, we propose a new classifier based on Dempster-Shafer (DS) theory and deep convolutional neural networks (CNN) for set-valued classification, called the *evidential deep-learning classifier*.[1] In this classifier, a deep CNN is used to extract high-order features from raw data. Then, the features are imported into a distance-based DS layer [9] for constructing mass functions. Finally, mass functions are used to compute the utilities of acts assigning to a set of classes for set-valued classification. The whole network is trained using an end-to-end learning procedure. Additionally, we provide a strategy for considering only some subsets

* Corresponding author.
E-mail addresses: zheng.tong@hds.utc.fr (Z. Tong), philippe.xu@hds.utc.fr (P. Xu), thierry.denoeux@hds.utc.fr (T. Denœux).

[1] A short preliminary version of this paper was presented at the SUM 2019 conference [62].

of classes instead of considering all of them. The effectiveness of the classifier and its decision strategy are demonstrated and discussed using three types of datasets (image, signal, and semantic relationship). The main contribution of this study is the demonstration that CNNs can be enhanced with set-valued classification and novelty detection capabilities thanks to the addition of an additional DS layer, while maintaining their very good performance in precise classification tasks.

1.1. Related work

In recent years, with the explosive development of deep learning [29], several models have been developed for precise classification, such as convolutional neural networks (CNNs) [25,31,65], recurrent neural networks [33,39,40], graph neural networks [53,54], and deep autoencoders [63,64]. Deep learning is a class of machine learning methods that uses multiple layers to progressively extract higher-level features from raw data as object representation. For example, when processing images using a CNN, lower layers may identify edges, while higher layers may identify more abstract concepts relevant to humans such as digits, letters or faces. Object representation based on deep learning is generally robust and reliable. In particular, the representation has a strong tolerance to translation and distortion of raw data. However, despite the power of the deep learning-based models in precise classification, we still face the problem of making them more cautious by allowing them to assign highly uncertain samples to sets of classes.

The Dempster-Shafer (DS) theory of belief functions [7,55], also referred to as *evidence theory*, can be harnessed to provide a solution to the problem. DS theory is a well-established formalism for reasoning and making decisions with uncertainty [70]. It is based on representing independent pieces of evidence by completely monotone capacities and combining them using a generic operator called Dempster's rule [55]. In the last two decades, DS theory has been increasingly applied to pattern recognition and supervised classification, following three main directions. The first one is *classifier fusion*, in which the outputs of several classifiers are transformed into belief functions and aggregated by a suitable combination rule (e.g., [1,35,48,74]). Another direction is *evidential calibration*: the decisions of classifiers are converted into mass functions with some frequency calibration property (e.g., [37,41,42,67,72]). The last approach is to design *evidential classifiers* (e.g., [9,13]), which break down the evidence of input features into elementary mass functions and combine them by Dempster's rule. The outputs of an evidential classifier can be used for decision-making [3,18]. Thanks to the generality and expressiveness of the DS formalism, the outputs of an evidential classifier provide more information than those of conventional classifiers (e.g., a neural network with a softmax layer) that convert an input feature vector into a probability distribution or any other distribution. For example, the expressiveness of an evidential classifier can be used for uncertainty quantification and ambiguity rejection [8,38]. Over the years, two main principles for designing an evidential classifier have been proposed: the model-based and distance-based approaches. The former uses estimated class-conditional distributions [57], while the latter constructs mass functions based on distances to prototypes [9,13]. In practice, the performance of an evidential classifier mainly depends on two factors: the training data set and the reliability of object representation.

In the last twenty years we have seen an increase in the size of benchmark datasets for supervised learning at an unprecedented rate from 10^2 to 10^5 [27] and even 10^9 instances [49]. However, little has been done to hybridize recent techniques for object representation, such as deep learning, with evidential classifiers for

decision-making. Some studies combining DS theory and deep learning has been reported, but most of these studies address the problem of deep-learning classifier fusion, where the outputs of several deep-learning models are regarded as pieces of evidence and aggregated by Dempster's rule of combination. For example, Soua et al. [58] use deep belief networks to independently predict traffic flow using streams of data and event-based data, and then update the beliefs from the networks by Dempster's conditional rule to achieve enhanced prediction. Tian et al. [61] also use Dempster's rule to fuse the beliefs from several deep-learning models with different types of data to detect anomalous network behavior patterns. Das et al. [5] use CNNs to perform superpixel semantic segmentation with three levels; DS theory is then utilized to combine the segmentation results of the three levels into reliable ones. Besides, Guo et al. [19] propose an "iFusion" framework, which uses Dempster's rule to combine different deep-learning discrimination models taking real-time or heterogeneous data as input. Similar works using DS theory for deep-learning classifier fusion can also be found in the field of posture recognition [32], remote-sensing images processing [15], and emotion classification [68]. In [11], the author shows that the operations performed in a multilayer perceptron classifier can be analyzed from the point of view of DS theory as the application of Dempster's rule; however, he does not propose a new model. Though Yuan et al. [71] propose a method using DS theory to measure the uncertainty of outputs from deep neural networks for decision-making, it still appears that little has been done to use features from a deep-learning model as inputs of an evidential classifier to generate informative mass-function outputs for decision-making allowing set-valued classification, a gap that we aim to fill in this work.

The rest of the paper is organized as follows. Section 2 starts with a brief reminder of DS theory, the DS layer for constructing mass functions, and feature representation via deep CNN. The new classifier is then introduced in Section 3. Section 4 reports numerical experiments, which demonstrate the advantages of the proposed classifier. Finally, we conclude the paper in Section 5.

2. Background

This section first recalls some necessary definitions regarding DS theory (Section 2.1) and the evidential neural network introduced in [9] (Section 2.2). Then, a brief description of feature representation via deep CNN is provided in Section 2.3.

2.1. Dempster-Shafer theory

The main concepts regarding DS theory are briefly presented in this section, and some basic notations are introduced. Detailed information can be found in Shafer's original work [55] and in the recent review [12].

Let $\Omega = \{\omega_1, \ldots, \omega_M\}$ be a finite set of states, called the *frame of discernment*. A *mass function* on Ω is a mapping m from 2^Ω to [0,1] such that $m(\varnothing) = 0$ and

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{1}$$

For any $A \subseteq \Omega$, each mass $m(A)$ is interpreted as a share of a unit mass of belief allocated to the hypothesis that the truth is in A, and which cannot be allocated to any strict subset of A based on the available evidence. Set A is called a *focal element* of m if $m(A) > 0$.

Two mass functions m_1 and m_2 representing independent items of evidence can be combined conjunctively by Dempster's rule $\oplus$ [55] as
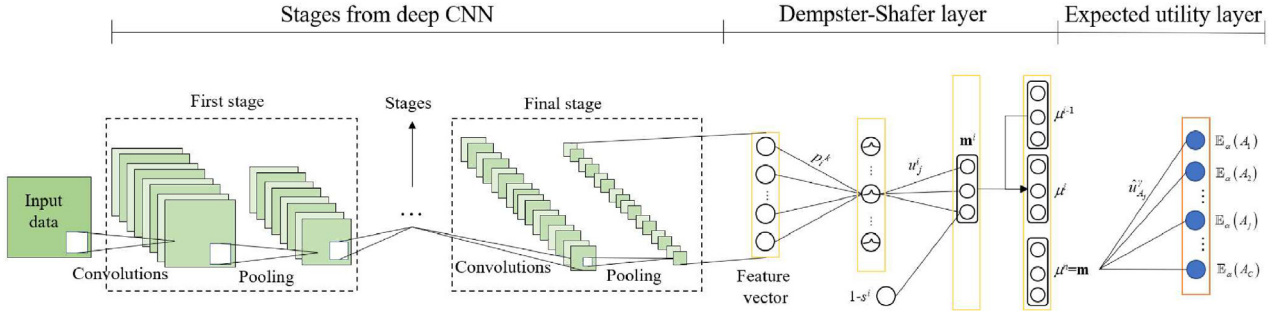
Fig. 1. Architecture of an evidential deep-learning classifier.

$$(m_1 \oplus m_2)(A) = \frac{(m_1 \cap m_2)(A)}{1 - (m_1 \cap m_2)(\varnothing)} \quad (2a)$$

for all $A \neq \varnothing$, with

$$(m_1 \cap m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C) \quad (2b)$$

and

$$(m_1 \cap m_2)(\varnothing) = \sum_{B \cap C = \varnothing} m_1(B) m_2(C). \quad (2c)$$

Mass functions m_1 and m_2 can be combined if and only if the denominator on the right-hand side of Eq. (2a) is strictly positive. The operator $\oplus$ is commutative and associative.

For decision-making with belief functions, we define the *lower and upper expected utilities* [10] of selecting ω_i as, respectively,

$$\underline{\mathbb{E}}_m\left(f_{\omega_i}\right) = \sum_{B \subseteq \Omega} m(B) \min_{\omega_j \in B} u_{ij}, \quad (3a)$$

and

$$\overline{\mathbb{E}}_m\left(f_{\omega_i}\right) = \sum_{B \subseteq \Omega} m(B) \max_{\omega_j \in B} u_{ij}, \quad (3b)$$

where $u_{ij} \in [0, 1]$ is the utility of selecting ω_i when the true state is ω_j, and f_{ω_i} denotes the act of selecting ω_i. A pessimistic decision-maker (DM) typically selects the act with the largest lower expected utility, while an optimistic DM maximizes the upper expected utility. The generalized Hurwicz decision criterion [23,24,60,10] models the DM's attitude to ambiguity by a *pessimism index* v and defines the expected utility of act f_{ω_i} as the weighted sum

$$\mathbb{E}_{m,v}\left(f_{\omega_i}\right) = v\underline{\mathbb{E}}\left(f_{\omega_i}\right) + (1 - v)\overline{\mathbb{E}}\left(f_{\omega_i}\right). \quad (4)$$

It is clear that the pessimistic and optimistic attitudes correspond, respectively, to $v = 1$ and $v = 0$.

2.2. Evidential neural network

Based on DS theory, Denœux [9] proposed a distance-based neural-network layer for constructing mass functions, also known as the *evidential neural network (ENN) classifier*. In the ENN classifier, the proximity of an input vector to prototypes is considered as evidence about its class. This evidence is converted into mass functions and combined using Dempster's rule. This section provides a short description of the ENN classifier.

We consider a training set $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^P$ of N examples represented with P-dimensional feature vectors, and an ENN classifier composed of n prototypes $\{\mathbf{p}^1, \ldots, \mathbf{p}^n\}$ in $\mathbb{R}^P$. For a test sample $\mathbf{x}$, the ENN classifier constructs mass functions that quantify the uncertainty about its class in $\Omega = \{\omega_1, \ldots, \omega_M\}$, using a three-step procedure. This procedure can be implemented in a

Table 1
Utility matrix extended by an OWA operator with $\gamma = 0.8$.

	Classes		
	ω_1	ω_2	ω_3
$f_{\{\omega_1\}}$	1	0	0
$f_{\{\omega_2\}}$	0	1	0
$f_{\{\omega_3\}}$	0	0	1
$f_{\{\omega_1, \omega_2\}}$	0.8	0.8	0
$f_{\{\omega_1, \omega_3\}}$	0.8	0	0.8
$f_{\{\omega_2, \omega_3\}}$	0	0.8	0.8
$f_{\{\Omega\}}$	0.6819	0.6819	0.6819

neural-network layer, which will be plugged into a deep CNN in Section 3.1. The three-step procedure is defined as follows.

Step 1: The distance-based support between $\mathbf{x}$ and each reference pattern $\mathbf{p}^i$ is computed as

$$s^i = \alpha^i \exp\left(-\left(\eta^i d^i\right)^2\right) \quad i = 1, \ldots, n, \quad (5)$$

where $d^i = \|\mathbf{x} - \mathbf{p}^i\|$ is the Euclidean distance between $\mathbf{x}$ and prototype $\mathbf{p}^i$, and $\alpha^i \in (0, 1)$ and $\eta^i \in \mathbb{R}$ are parameters associated with prototype $\mathbf{p}^i$. Prototype vectors $\mathbf{p}^1, \ldots, \mathbf{p}^n$ can be considered as vectors of connection weights between the input layer and a hidden layer of n Radial basis Function (RBF) units.

Step 2: The mass function m^i associated to reference pattern $\mathbf{p}^i$ is computed as

$$m^i(\{\omega_j\}) = h_j^i s^i, \quad j = 1, \ldots, M \quad (6a)$$
$$m^i(\Omega) = 1 - s^i, \quad (6b)$$

where h_j^i is the degree of membership of prototype $\mathbf{p}^i$ to class ω_j with $\sum_{j=1}^M h_j^i = 1$. We denote the vector of masses induced by prototype $\mathbf{p}^i$ as

$$\mathbf{m}^i = \left(m^i(\{\omega_1\}), \ldots, m^i(\{\omega_M\}), m^i(\Omega)\right)^T.$$

Eq. (6) can be regarded as computing the activation of units in a second hidden layer of the ENN classifier, composed of n modules of $M + 1$ units each. The result of module i corresponds to the belief masses assigned by m^i.

Step 3: The n mass functions $\mathbf{m}^i, i = 1, \ldots, n$, are aggregated by Dempster's rule (2). The combined mass function is computed iteratively as $\mu^1 = m^1$ and $\mu^i = \mu^{i-1} \cap m^i$ for $i = 2, \ldots, n$. We have

$$\mu^i(\{\omega_j\}) = \mu^{i-1}(\{\omega_j\}) m^i(\{\omega_j\}) \\ + \mu^{i-1}(\{\omega_j\}) m^i(\{\Omega\}) + \mu^{i-1}(\Omega) m^i(\{\omega_j\}) \quad (7a)$$
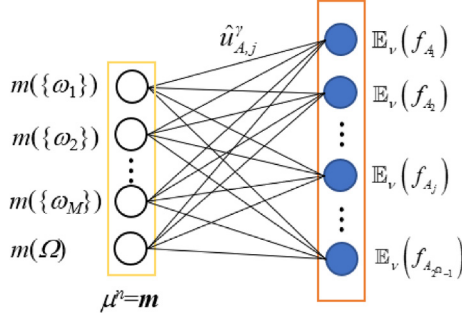
Fig. 2. Architecture of the expected utility layer.

for $i = 2, \ldots, n$ and $j = 1, \ldots, M$, and

$$\mu^i(\Omega) = \mu^{i-1}(\Omega)m^i(\Omega) \quad i = 2, \ldots, n. \tag{7b}$$

The vector of outputs from the ENN classifier $\boldsymbol{m} = (m(\{\omega_1\}), \ldots, m(\{\omega_M\}), m(\Omega))^T$ is finally obtained as

$$m(\{\omega_j\}) = \frac{\mu^n(\{\omega_j\})}{\sum_{j'=1}^M \mu^n(\{\omega_{j'}\}) + \mu^n(\Omega)}$$

and

$$m(\Omega) = \frac{\mu^n(\Omega)}{\sum_{j'=1}^M \mu^n(\{\omega_{j'}\}) + \mu^n(\Omega)}.$$

2.3. Feature representation via deep CNN

In practice, the effectiveness of an ENN classifier heavily depends on the information contained in its input features. Feature representation, an essential part of the machine learning workflow, consists in discovering the predictors needed for classification from raw data. In recent years, deep learning models [29] have become very popular because of their ability to construct rich deep feature representations, allowing them to achieve exceptional performance in such tasks as pattern recognition and segmentation [17,36,73], signal processing [47,50], and even material discovery [44,59].

Deep CNNs, one of the most widely used deep learning architectures, are a special type of multi-layered neural network and the main focus of this paper. The most common CNNs consist of convolutional layers, pooling layers, and fully connected layers. Convolutional and pooling layers are defined as stages. A stage converts its input data into an intermediate representation, working as a feature extractor. In general, a deep CNN is composed of several stacked stages that process raw data and repeatedly converts them into higher-level feature maps. Then, fully connected layers, serving as a decision maker, assign the input to one of the classes based on the feature maps. Therefore, the final output of the stacked stages in a deep CNN can be considered as a feature representation of the input data. In the study, these high-level features are used as input to a DS layer capable of set-valued classification, as will be shown in Section 3.1.

To understand the feature representation of deep CNNs, we briefly recall the processes of convolutional and pooling layers. Consider a stage with input $\boldsymbol{z} = (\boldsymbol{z^1}, \ldots, \boldsymbol{z^D})$ consisting of D input maps or *input channels* z^i ($i = 1, \ldots, D$) with size $H \times W$. A convolutional layer consists of several convolution kernels that extract feature maps from $\boldsymbol{z}$. A convolution kernel is a small matrix used to apply a convolution operation to each input map by sliding over the map, performing an element-wise multiplication with the part of the input map where the kernel is currently on, summing up the

multiplied results into a single value, and then adding the bias of the kernel to the summed value. Thus, the processes in a convolutional layer, consisting of e convolution kernels with size $a \times b$, are expressed as

$$c^j = f\left(b^j + \sum_i w^{i,j} * z^i\right), \tag{8}$$

where $w^{i,j}$ is the convolution kernel between the i-th input map and the j-th output map; b^j is the bias of kernel $w^{i,j}$; $*$ denotes the convolution operation; z^i is the i-th input map with size $H \times W, i = 1, \ldots, D; c^j$ is the j-th output feature map, with size $\frac{H-a+1}{r} \times \frac{W-b+1}{r}, j = 1, \ldots, e; r$ is the stride with which the kernel slides over input map $z^i; f$ is the activation function, such as the rectified linear unit $\mathrm{ReLU}(x) = \max(0, x)$ [28]. A pooling operation with an $s \times s$ non-overlapping local region is formulated as

$$po_{a,b}^k = (\beta^1, \ldots, \beta^{s \times s})^T \cdot \mathrm{Or}\left(c_{as,bs}^k, \ldots, c_{as+s,bs}^k, \ldots, c_{as+s,bs+s}^k\right), \tag{9}$$

where $po_{a,b}^k$ is the element (a, b) from the k-th output map, which is in the a-th row and the b-th column; Or is a sort function from maximum to minimum; $\cdot$ denotes dot product; $(\beta^1, \ldots, \beta^{s \times s})$ is the pooling weight vector, such as max pooling $(\beta^1, \beta^2, \ldots, \beta^{s \times s}) = (1, 0, \ldots, 0)$ and mean pooling

$$(\beta^1, \ldots, \beta^{s \times s}) = \left(\frac{1}{s \times s}, \ldots, \frac{1}{s \times s}\right).$$

3. Proposed classifier

In this section, we describe the proposed classifier. Section 3.1 presents the overall architecture composed of several stages from a deep CNN for feature representation, a DS layer to construct mass functions, and an expected utility layer for decision-making. The details of the expected utility layer are described in Section 3.2, and the learning strategy for the proposed classifier is exposed in Section 3.3. Finally, an approach for selecting partial multi-class acts is introduced in Section 3.4.

3.1. Network architecture

The main idea of this work is to hybridize the ENN classifier presented in Section 2.2 and the CNN architecture recalled in Section 2.3 by "plugging" a DS layer followed by a utility layer at

Table 3
Example of utilities and losses

Examples	Expected utility			Loss ($\omega_* = \omega_1$)
	$\mathbb{E}_1(\{\omega_1\})$	$\mathbb{E}_1(\{\omega_2\})$	$\mathbb{E}_1(\{\omega_3\})$	
#1	0.70	0.10	0.10	0.303
#2	0.97	0.01	0.01	0.026
#3	0.50	0.50	0	0.602
#4	0.40	0.40	0	0.796

Table 2
Examples of DS layer outputs

Examples	Outputs of a DS layer			
	$m(\{\omega_1\})$	$m(\{\omega_2\})$	$m(\{\omega_3\})$	$m(\Omega)$
#1	0.70	0.10	0.10	0.10
#2	0.97	0.01	0.01	0.01
#3	0.50	0.50	0	0
#4	0.40	0.40	0	0.2

the output of a CNN. The architecture of the proposed method, called the *evidential deep-learning classifier*, is illustrated in Fig. 1. An evidential deep-learning classifier has the ability to perform set-valued classification and quantify the uncertainty about the class of the sample on $\Omega = \{\omega_1, \ldots, \omega_M\}$ by a belief function. Propagation of information through this network can be described by the following three-step procedure:

Step 1: An input sample is propagated into several stages of a CNN architecture to extract latent features relevant for classification, as done in a probabilistic CNN. In the final stage, the *P*-dimensional output vector is a feature representation of the sample, ready to be fed as input to the DS layer. This architecture provides a robust and reliable representation of the input sample. Thanks to this representation, the evidential deep-learning classifier yields similar or even better performance for precise classification than does a probabilistic classifier with the same stages. This superiority will be demonstrated by performance comparisons between the evidential and probabilistic deep-learning classifiers in precise classification tasks (Section 4).

Step 2: The feature vector computed in Step 1 is fed into the DS layer, in which it is converted into mass functions aggregated by Dempster's rule, as explained in Section 2.2. The output of the DS layer is an $(M + 1)$ mass vector

$$\boldsymbol{m} = (m(\{\omega_1\}), \ldots, m(\{\omega_M\}), m(\Omega))^T,$$

which characterizes the classifier's belief about the probability of the sample class and quantifies the uncertainty in the object representation. The mass $m(\{\omega_i\})$ is a degree of belief that the sample belongs to class ω_i. The DS layer tends to allocate masses uniformly across classes when the feature representation contains confusing and conflicting information. The additional degree of freedom

$m(\Omega)$ makes it possible to quantify the lack of evidence and verify whether the model is well trained [62]. The advantages of the DS layer will be verified in the performance evaluation of set-valued classification using evidential deep-learning classifiers reported in Section 4.

Step 3: The output mass vector $\boldsymbol{m}$ is fed into an expected utility layer for decision-making, where it is used to compute the expected utilities of acts. Each act is defined as the assignment of the sample to a non-empty subset A of Ω. Thus, the output of the expected utility layer is an expected-utility vector of size at most equal to $2^M - 1$ if all of the possible acts are considered. The expected utility layer allows the proposed classifier to perform set-valued classification. This capability will be demonstrated by the performance comparison between the evidential and probabilistic deep-learning classifiers in set-valued classification and novelty detection tasks reported in Section 4. The details of the expected utility layer for set-valued classification are introduced in the next section.

### 3.2. Expected utility layer

Let $\Omega = \{\omega_1, \ldots, \omega_M\}$ be the set of classes. For classification problems with only precise prediction, an act is defined as the assignment of an example to one and only one of the M classes. The set of acts is $\mathcal{F} = \left\{ f_{\omega_1}, \ldots, f_{\omega_M} \right\}$, where f_{ω_i} denotes assignment to class ω_i. To make decisions, we define a utility matrix $\mathbb{U}_{M \times M}$, whose general term $u_{ij} \in [0, 1]$ is the utility of assigning an example to class ω_i when the true class is ω_j. Here, $\mathbb{U}_{M \times M}$ is called the *original utility matrix*. For decision-making with belief functions, each act f_{ω_i} induces expected utilities, such as the lower and upper expected utilities defined by (3).

		Labels			
		ω_1	ω_2	ω_3	ω_4
Acts	$f\omega_1$	557	115	24	13
	$f\omega_2$	107	679	32	14
	$f\omega_3$	13	16	663	128
	$f\omega_4$	25	32	145	627

(a)

		Labels			
		ω_1	ω_2	ω_3	ω_4
Acts	$f\omega_1$	0.793	0.136	0.027	0.017
	$f\omega_2$	0.152	0.806	0.037	0.018
	$f\omega_3$	0.018	0.019	0.767	0.167
	$f\omega_4$	0.035	0.038	0.168	0.802

(b)
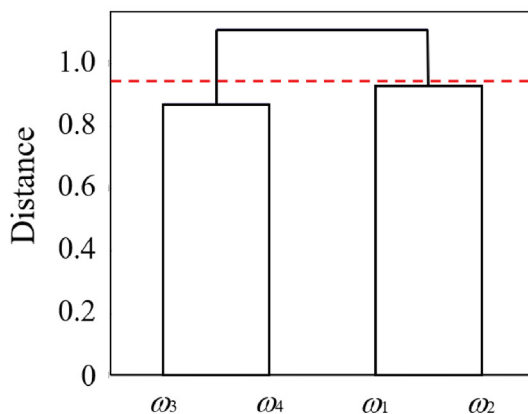
(c)
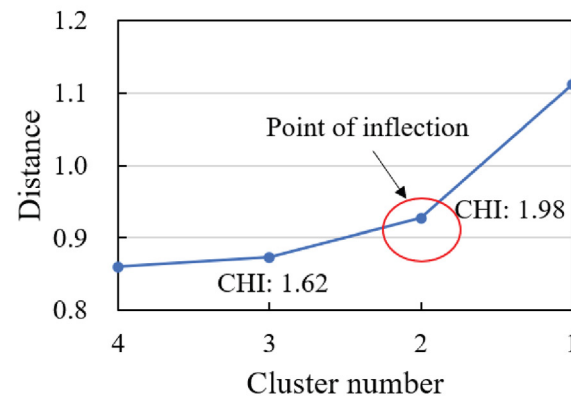
(d)

Fig. 3. An example of act selection: confusion matrix (a), normalized confusion matrix (b), dendrogram (c), and a curve of distance vs. cluster number (d).

For classification problems with imprecise prediction, Ma and Denœux [38] proposed an approach to conduct set-valued classification under uncertainty by generalizing the set of acts as partially assigning a sample to a non-empty subset A of Ω. Thus, the set of acts becomes $\mathcal{F} = \left\{ f_A, A \in 2^{\Omega} \setminus \varnothing \right\}$, in which 2^{Ω} is the power set of Ω and f_A denotes the assignment to a subset A. In this study, subset A is defined as a *multi-class set* if $|A| \geqslant 2$. For decision-making with $\mathcal{F}$, the original utility matrix $\mathbb{U}_{M \times M}$ is extended to $\mathbb{U}_{(2^{\Omega}-1) \times M}$, where each element $\hat{u}_{A,j}$ denotes the utility of assigning a sample to set A of classes when the true label is ω_j.

When the true class is ω_j, the utility of assigning a sample to set A is defined as an Ordered Weighted Average (OWA) aggregation [69] of the utilities of each precise assignment in A as

$$\hat{u}_{A,j} = \sum_{k=1}^{|A|} g_k \cdot u^A_{(k)j}, \tag{10}$$

where $u^A_{(k)j}$ is the k-th largest element in the set $\left\{ u^A_{ij}, \omega_i \in A \right\}$ made up of the elements in the original utility matrix $\mathbb{U}_{M \times M}$, and weights $\boldsymbol{g} = \left(\boldsymbol{g_1}, \ldots, \boldsymbol{g_{|A|}} \right)$ represent the preference to choose $u^A_{(k)j}$ when a classifier has to make a precise decision among a set of possible choices. The elements in weight vector $\boldsymbol{g}$ represent the DM's *tolerance to imprecision*. For example, full tolerance to imprecision is achieved when the assignment act f_A has utility 1 once set A contains the true label, no matter how imprecise A is. In the case, only the maximum utility of elements in set $\left\{ u^A_{ij}, \omega_i \in A \right\}$ is considered: $\left(g_1, g_2, \ldots, g_{|A|} \right) = (1, 0, \ldots, 0)$. At the other extreme, a DM attaching no value to imprecision would consider the act f_A as equivalent to selecting one class uniformly at random from A: this is achieved when

$$\left(g_1, g_2, \ldots, g_{|A|} \right) = \left(\frac{1}{|A|}, \frac{1}{|A|}, \ldots, \frac{1}{|A|} \right).$$

In this study, following [38], we determine the weight vector $\boldsymbol{g}$ of the OWA operators by adapting O'Hagan's method [46]. We define the *imprecision tolerance degree* as

$$TDI(\boldsymbol{g}) = \sum_{k=1}^{|A|} \frac{|A| - k}{|A| - 1} g_k = \gamma, \tag{11}$$

which equals to 1 for the maximum, 0 for the minimum, and 0.5 for the average. In practice, we only need to consider values of γ between 0.5 and 1 as a precise assignment is preferable to an imprecise one when $\gamma < 0.5$ [38]. Given a value of γ, we can compute the weights of the OWA operator by maximizing the entropy

$$ENT(\boldsymbol{g}) = -\sum_{k=1}^{|A|} g_k \log g_k \tag{12}$$

subject to the constraints $TDI(\boldsymbol{g}) = \gamma$, $\sum_{k=1}^{|A|} g_k = 1$, and $g_k \geqslant 0$.

Example 1. Table 1 shows an example of the extended utility matrix generated by an OWA operator with $\gamma = 0.8$ for a classification problem. The first three rows constitute the original utility matrix, indicating that the utility equals 1 when assigning a sample to its true class, otherwise it equals 0. The remaining rows are the matrix of the aggregated utilities. For example, we get a utility of 0.8 when assigning a sample to set $\{\omega_1, \omega_2\}$ if the true label is ω_1.

Based on an extended utility matrix $\mathbb{U}_{(2^{\Omega}-1) \times M}$ and the outputs of a DS layer $\boldsymbol{m}$, we can compute the expected utility of an act assigning a sample to set A using the generalized Hurwicz criterion (4) as

Table 4
The three baseline stages used on CIFAR-10 data.

NIN [34]	FitNet-4 [43]	ViT-L/16 [14]
Input: 32 × 32 × 3		
		16 × 16 × 3 × 4 patches with positional encoding
5 × 5 Conv. NIN 64 ReLU	3 × 3 Conv. 32 ReLU	3 × 3 Conv. 32 ReLU
	3 × 3 Conv. 32 ReLU	3 × 3 Conv. 32 ReLU
	3 × 3 Conv. 32 ReLU	3 × 3 Conv. 32 ReLU
	3 × 3 Conv. 48 ReLU	3 × 3 Conv. 48 ReLU
	3 × 3 Conv. 48 ReLU	3 × 3 Conv. 48 ReLU
2 × 2 max-pooling with 2 strides		
5 × 5 Conv. NIN 64 ReLU 2 × 2 mean-pooling with 2 strides	3 × 3 Conv. 80 ReLU	3 × 3 Conv. 80 ReLU
	3 × 3 Conv. 80 ReLU	3 × 3 Conv. 80 ReLU
	3 × 3 Conv. 80 ReLU	3 × 3 Conv. 80 ReLU
	3 × 3 Conv. 80 ReLU	3 × 3 Conv. 80 ReLU
	3 × 3 Conv. 80 ReLU	3 × 3 Conv. 80 ReLU
2 × 2 max-pooling with 2 strides		
5 × 5 Conv. NIN 128 ReLU 2 × 2 mean-pooling with 2 strides	3 × 3 Conv. 128 ReLU	3 × 3 Conv. 128 ReLU
	3 × 3 Conv. 128 ReLU	3 × 3 Conv. 128 ReLU
	3 × 3 Conv. 128 ReLU	3 × 3 Conv. 128 ReLU
	3 × 3 Conv. 128 ReLU	3 × 3 Conv. 128 ReLU
	3 × 3 Conv. 128 ReLU	3 × 3 Conv. 128 ReLU
	8 × 8 max-pooling with 2 strides	4 × 4 max-pooling with 2 strides + positional encoding
Average global pooling		Transformer decoder

$$\mathbb{E}_{m,v}(f_A) = v\underline{\mathbb{E}}_m(f_A) + (1 - v)\overline{\mathbb{E}}_m(f_A), \tag{13a}$$

where $\underline{\mathbb{E}}_m(f_A)$ and $\overline{\mathbb{E}}_m(f_A)$ are, respectively, the lower and upper expected utilities

$$\underline{\mathbb{E}}_m(f_A) = \sum_{B \subseteq \Omega} m(B)\min_{\omega_k \in B} \hat{u}_{A,j}, \tag{13b}$$

$$\overline{\mathbb{E}}_m(f_A) = \sum_{B \subseteq \Omega} m(B)\max_{\omega_k \in B} \hat{u}_{A,j}, \tag{13c}$$

and v is the pessimism index, which is considered as a hyperparameter of the proposed classifier. The sample is finally assigned to set A such that

$$A = \arg \max_{\varnothing \neq B \subseteq \Omega} \mathbb{E}_{m,v}(f_B). \tag{14}$$

Similar to the DS layer, the procedure of assigning a sample to a set in $\mathcal{F}$ using utility theory can be summarized as a layer of the neural network, called an *expected utility layer*, as shown in Fig. 2. In this layer, the inputs and outputs are, respectively, the mass vector $\boldsymbol{m}$ from the preceding DS layer and the expected utilities of all acts in $\mathcal{F}$. The connection weight between unit j of the DS layer and output unit $A \subseteq \Omega$ corresponding to the assignment to set A is the utility

Table 5
Test average utilities in precise classification on CIFAR-10 data.

Models	NIN [34]		FitNet-4 [43]		ViT-L/16 [14]	
	Probabilistic classifier	Evidential classifier	Probabilistic classifier	Evidential classifier	Probabilistic classifier	Evidential classifier
Utility	0.8959	0.8978	0.9353	0.9361	0.9921	0.9908
p-value (McNemar's test)	0.0489		0.0492		0.0452	

Table 6
Test average utilities for precise classification of the CIFAR-100 data after transfer learning.

Models	NIN [34]		FitNet-4 [43]		ViT-L/16 [14]	
	CNN classifier	Evidential classifier	Probabilistic classifier	Evidential classifier	Probabilistic classifier	Evidential classifier
Utility	0.3442	0.3461	0.6688	0.6714	0.8251	0.8217
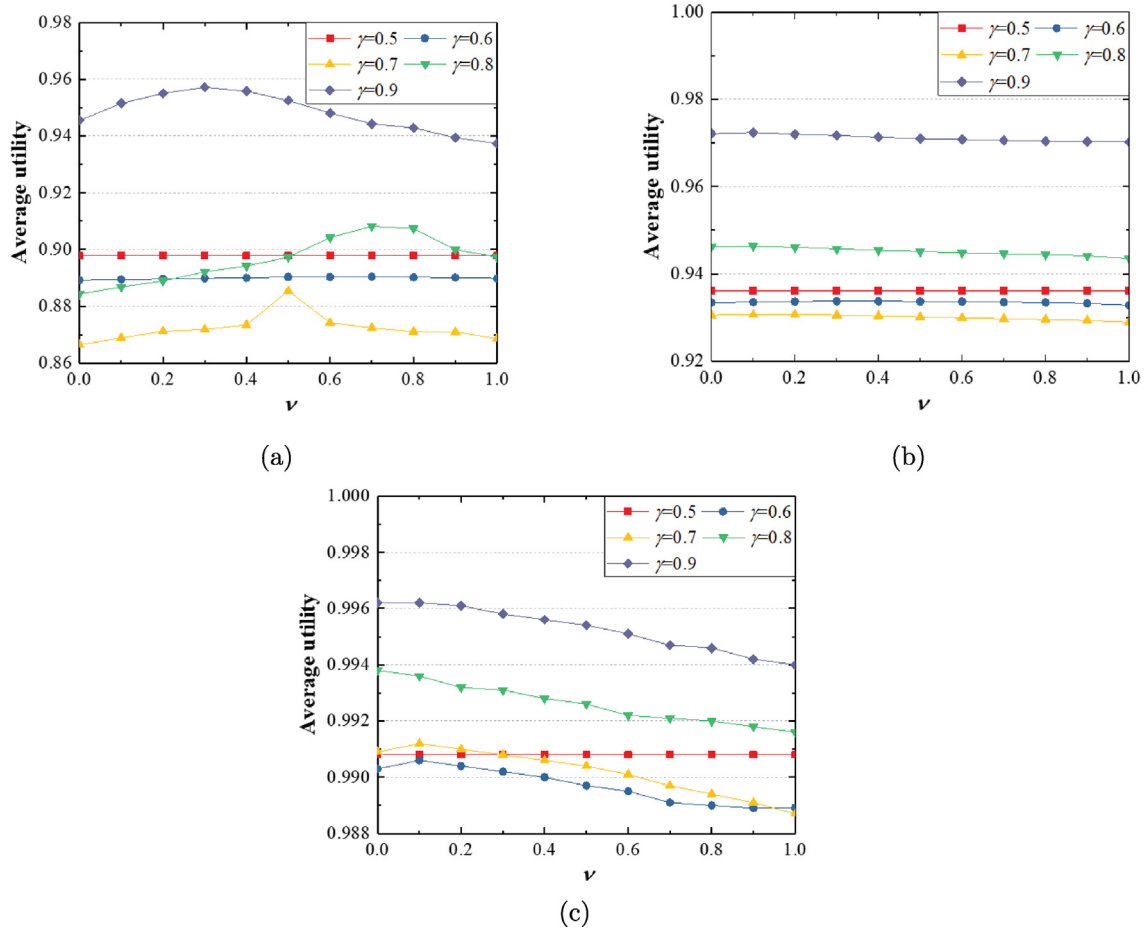
Fig. 4. Average utility vs. v for the proposed classifiers on the CIFAR-10 dataset: NIN (a), FitNet-4 (b), and ViT-L/16 (c).

value $\widehat{u}_{A_j}$. As coefficient γ describing the imprecision tolerance degree is fixed, the connection weights of the expected utility layer do not need to be updated during training.

3.3. Learning

The evidential deep-learning classifier can be trained by a stochastic gradient descent algorithm. Given a sample $\boldsymbol{x}$ with class label ω_*, we define the prediction loss as

$$\mathcal{L}_v(\boldsymbol{x}) = -\sum_{k=1}^{M} y_k \log \mathbb{E}_v\left(f_{\omega_k}\right) + (1 - y_k) \log\left(1 - \mathbb{E}_v\left(f_{\omega_k}\right)\right) \quad (15a)$$

with

$$y_k = \begin{cases} 1 & \text{if } \omega_k = \omega_* \\ 0 & \text{if } \omega_k \neq \omega_* \end{cases}. \quad (15b)$$

The loss $\mathcal{L}_v(\boldsymbol{x})$ is minimized when $\mathbb{E}_v\left(f_{\omega_k}\right) = 1$ for $\omega_k = \omega_*$ and $\mathbb{E}_v\left(f_{\omega_l}\right) = 0$ for $\omega_l \neq \omega_*$.

Example 2. Table 2 shows several examples, whose utilities of single-valued assignments and losses are shown in Table 3. The extended utility matrix is shown in Table 1, and v equals 1. We assume that $\Omega = \{\omega_1, \omega_2, \omega_3\}$ and $\omega_* = \omega_1$. Eq. (15) yields different losses given a set of DS layer outputs.

The derivatives of $\mathcal{L}_v(\boldsymbol{x})$ of the error w.r.t $\boldsymbol{m}$ in the expected utility layer are

$$\frac{\partial \mathcal{L}_v(\boldsymbol{x})}{\partial m(\{\omega_k\})} = -\frac{y_k}{\mathbb{E}_v\left(f_{\omega_k}\right)}\left[\widehat{u}_{\{\omega_k\},k} + (1-v)\max_{i=1,\ldots,M}\widehat{u}_{\{\omega_k\},i}\right], \tag{16a}$$

$$\frac{\partial \mathcal{L}_v(\boldsymbol{x})}{\partial m(\Omega)} = -\sum_{k=1}^{M}\frac{y_k}{\mathbb{E}_v\left(\{f_{\omega_k}\}\right)}(1-v)\max_{i=1,\ldots,M}\widehat{u}_{\{\omega_k\},i}. \tag{16b}$$

The derivatives of $\mathcal{L}_v(\boldsymbol{x})$ w.r.t p_k^i, η^i, and ξ^i in a DS layer are the same as the original work of Denœux [9]:

$$\frac{\partial \mathcal{L}_v(\boldsymbol{x})}{\partial p_k^i} = \frac{\partial \mathcal{L}_v(\boldsymbol{x})}{\partial s^i}2\left(\eta^i\right)^2 s^i(x_k - p_k^i), \quad k = 1,\ldots,P, \ i$$
$$= 1,\ldots,n, \tag{17}$$

$$\frac{\partial \mathcal{L}_v(\boldsymbol{x})}{\partial \eta^i} = \frac{\mathcal{L}_v(\boldsymbol{x})}{\partial s^i}\left(-2\eta^i\left(d^i\right)^2 s^i\right), \quad i = 1,\ldots,n, \tag{18}$$

and

$$\frac{\partial \mathcal{L}_v(\boldsymbol{x})}{\partial \xi^i} = \frac{\mathcal{L}_v(\boldsymbol{x})}{\partial s^i}\exp\left(-\left(\eta^i d^i\right)^2\right)(1-\alpha^i)\alpha^i, \quad i = 1,\ldots,n, \tag{19}$$

where P is the dimension of the reference patterns and the input feature vector and n is the number of prototypes.

In the proposed classifier, the DS layer is connected to the pooling layer of the last convolutional stage, as shown in Fig. 1. Thus, we can compute the derivatives of the error w.r.t. x_k and po^k as

$$\frac{\partial \mathcal{L}_v(\boldsymbol{x})}{\partial x_k} = \frac{\mathcal{L}_v(\boldsymbol{x})}{\partial po^k} = -2\frac{\mathcal{L}_v(\boldsymbol{x})}{\partial s^i}\left(\eta^i\right)^2 s^i\sum_{i=1}^{n}(x_k - p_k^i), \quad k$$
$$= 1,\ldots,P, \tag{20}$$

where po^k is the k-th output map in the final pooling layer, which is a 1×1 tensor. Error propagation in the remaining stages is performed as in a probabilistic CNN.

3.4. Act selection

As explained in Section 3.2, the set of acts when considering multi-class assignment is $\mathcal{F} = \left\{f_A, A \in 2^{\Omega} \setminus \varnothing\right\}$, as instances can be assigned to any non-empty subset A of Ω. However, the cardinality of $\mathcal{F}$ increases exponentially with the number of classes, which could preclude the application of this approach when the number M of classes is large.

In [62], we showed that a neural network with convolutional layers and a DS layer tends to assign samples to multi-class sets when candidate classes are similar, such as, e.g., "cat" and "dog", or "horse" and "deer". Thus, it may be advantageous to only consider partial multi-class acts assigning samples to subsets containing two or more similar classes.

In this study, we propose a strategy to determine similar classes in the frame of discernment and select partial multi-class acts from $\mathcal{F}$ based on class similarity. Using the selected partial multi-class acts, rather than all acts in $\mathcal{F}$, we can reduce the compute cost in set-valued assignments. This strategy can be described as follows.

Step 1: A confusion matrix with only precise assignments is generated by a trained evidential deep-learning classifier using the training set. In the confusion matrix, each column represents the predicted sample distribution in one class.
Step 2: Each column in the confusion matrix is normalized using its total number. Each normalized column is regarded as the feature of its corresponding class.

Step 3: The Euclidean distance between every two features is computed, and a dendrogram is generated by a hierarchical agglomerative clustering (HAC) algorithm [6,56]. The distance between every two features represents the similarity of the two classes. The distance is close to 0 if two classes are similar.
Step 4: The distance can be drawn versus the number of clusters based on the dendrogram, as shown in Fig. 3d. A point of inflection in the curve can then be used to determine the threshold for cutting the dendrogram. In this study, we used the Calinski-Harabasz index (CHI) [2] to determine this point. The point of inflection is the one in the curve with the maximum CHI, as illustrated in Fig. 3d of Example 3. The right of the point has a small number of highly similar classes. This can be explained by the nature of the HAC algorithm [6]. Very similar classes are consolidated first as the algorithm proceeds. Toward the end of the HAC run, we reach a stage where dissimilar classes are left to be merged but the distance between them is large; these classes are not similar and do not need to be clustered in the act-selection strategy.
Step 5: The distance corresponding to the inflection point is used as the threshold to cut the dendrogram. Similar patterns are the classes in the clustered groups with the distance lower than the threshold. Finally, we select the multi-class acts corresponding to similar classes.

Example 3. Fig. 3 shows an example of act selection, in which a HAC algorithm with Ward linkage is used to generate a dendrogram. Fig. 3d display a point of inflection whose *CHI* is 1.91 and corresponding distance is 0.927. The distance is used as the threshold of the Euclidean distance to cut the dendrogram. There are two pairs of similar patterns: $\{\omega_1, \omega_2\}$ and $\{\omega_3, \omega_4\}$. Thus, the selected partial multi-class acts are $f_{\{\omega_1,\omega_2\}}$ and $f_{\{\omega_3,\omega_4\}}$.

4. Experiments

In this section, we present numerical experiments demonstrating the advantages of the proposed classifier. In Section 4.1, we provide three metrics for performance evaluation. Experimental results on image recognition, signal processing and semantic-relationship classification tasks are then reported and discussed, respectively, in Sections 4.2, 4.3 and 4.4.

4.1. Evaluation of set-valued classification

In the applications of evidential deep-learning classifiers, we use the extended utility matrix $\mathbb{U}_{(2^{\Omega}-1)\times M}$ for performance evaluation. For a dataset T, the classification performance is evaluated by the *averaged utility* as

$$AU(T) = \frac{1}{|T|}\sum_{i=1}^{|T|}\widehat{u}_{A(i),y_i}, \tag{21}$$

where y_i is the true class of learning example $i, A(i)$ is the selected subset for example i using (14) and, using the notation introduced in Section 3.2, $\widehat{u}_{A,y_i}$ is the utility of assigning sample i to subset $A \subseteq \Omega$ when its true class is y_i. When only considering precise acts, the *AU* criterion defined by (21) boils down to classification accuracy. *The averaged cardinality* is also computed as
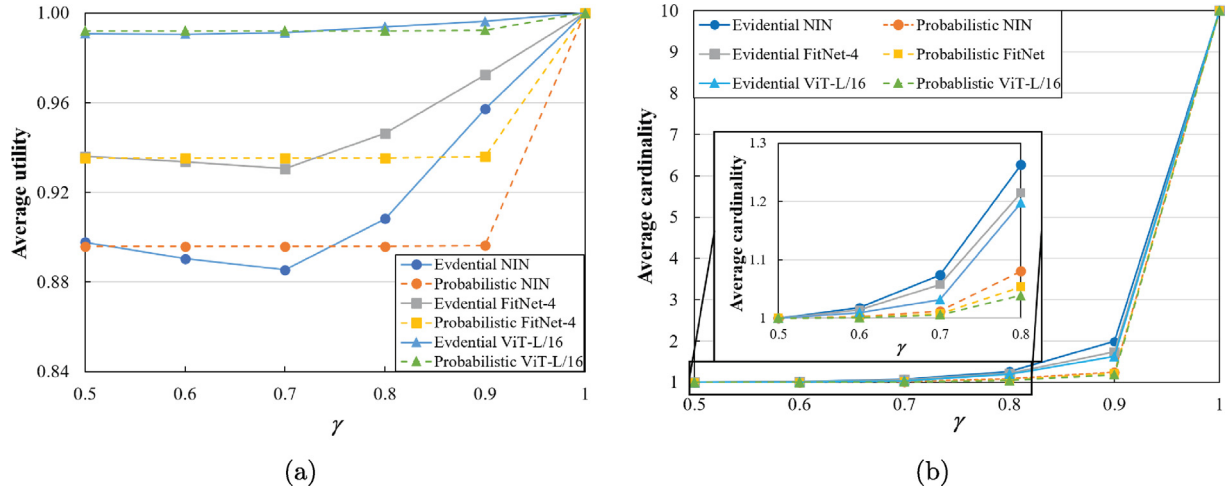
$$AC(T) = \frac{1}{|T|}\sum_{i=1}^{|T|}|A(i)|. \tag{22}$$

Fig. 5. Average utility (a) and average cardinality (b) vs. γ for the evidential and probabilistic deep-learning classifiers on the CIFAR-10 dataset.

Table 7
Label classification/utilities with different γ.

	#1(ω^*=cat)	#2(ω^*=dog)	#3(ω^*=deer)	#4(ω^*=automobile)
γ=0.5	{dog}/0	{dog}/1	{deer}/1	{airplane}/0
γ=0.6	{cat,dog}/0.6	{cat,dog}/0.6	{deer}/1	{airplane}/0
γ=0.7	{cat,dog}/0.7	{cat,dog}/0.7	{deer,horse}/0.7	{airplane}/0
γ=0.8	{cat,dog}/0.8	{cat,dog}/0.8	{deer,horse}/0.8	{airplane}/0
γ=0.9	{cat,dog}/0.9	{cat,dog}/0.9	{cat,deer,dog,horse}/0.7104	{cat,deer,dog,horse}/0
γ=1.0	Ω/1.0	Ω/1.0	Ω/1.0	Ω/1.0

Additionally, we also consider the case where a dataset $T' = \{T'_O, T'_I\}$ is composed of a subset T'_O of outliers whose class does not belong to the frame of discernment Ω, and a subset T'_I of inliers whose class belongs to Ω. We compare the rate of f_Ω in T'_I and T'_O to evaluate the capacity of a classifier to reject outliers together with ambiguous samples. This capacity is called *novelty detection* in [9]. Generally, a well-trained classifier is expected to have a low rate of f_Ω in T'_I but a high rate in T'_O.

In this study, we compare the proposed classifiers with probabilistic CNNs. To ensure a fair comparison, we adopt the following strategy for probability-based set-valued classification in CNNs: $f_A \succeq_s f_{A'}$ if and only if $\mathbb{E}(f_A) \leqslant \mathbb{E}(f_{A'})$, with $\mathbb{E}(f_A) = \sum_{\omega_k \in A} p(\omega_k) \cdot \hat{u}_{A,k}$.

4.2. Image classification experiment

We used the CIFAR-10 dataset to evaluate the performance of the proposed classifier in image classification. The CIFAR-10 dataset [26] consists of 60,000 RGB images of size 32×32 partitioned in 10 classes. There are 50,000 training examples and 10,000 testing examples. During training, we randomly selected 10,000 images as validation data. We used two datasets (CIFAR-100 [26] and MNIST [30]) for novelty detection. The CIFAR-100 dataset is just like the CIFAR-10 except it has 100 classes containing 600 images each, while the MNIST dataset of handwritten digits has 70,000 examples. All examples in the two datasets are used for novelty detection except some images whose classes are included in the CIFAR-10 dataset.

Precise classification. In this experiment, the convolutional stages of three probabilistic CNNs were combined with the DS and expected utility layers, as shown in Table 4. The three probabilistic CNNs have the same number of output feature maps but different convolutional and pooling layers. As shown in Table 5, the proposed classifiers slightly outperform the probabilistic ones in precise classification, except with ViT-L/16 feature extraction. McNemar's test results indicate a small but statistically significant effect of the proposed combination on the image classification task with *p*-values below 5%. These results suggest that the utility of an evidential classifier is larger than that of a probabilistic CNN classifier with the same stage as the evidential one. They also demonstrate that the use of the convolutional and pooling layers in Step 1 of Section 3.1 allows for good precise-classification performance of the evidential deep-learning classifier.

Transfer learning. The feasibility of transfer learning on the proposed classifier was also verified in this study. The three evidential deep-learning classifiers trained on the CIFAR-10 classification task, as well as the three probabilistic CNNs, were fine-tuned using the training set of the CIFAR-100 dataset as a new task. Table 6 shows the testing utilities of fine-tuned classifiers on the CIFAR-100 dataset. The evidential and probabilistic classifiers achieve close results for precise classification after fine-tuning. Besides, the average utilities of the evidential deep-learning classifiers are close to those already reported in [34,43,14]. This demonstrates the feasibility of transfer learning with the proposed classifiers.

Set-valued classification. Before evaluating the performance of the proposed classifiers in set-valued assignments, we need to determine the optimal pessimism index v in Eq. (13a) once given
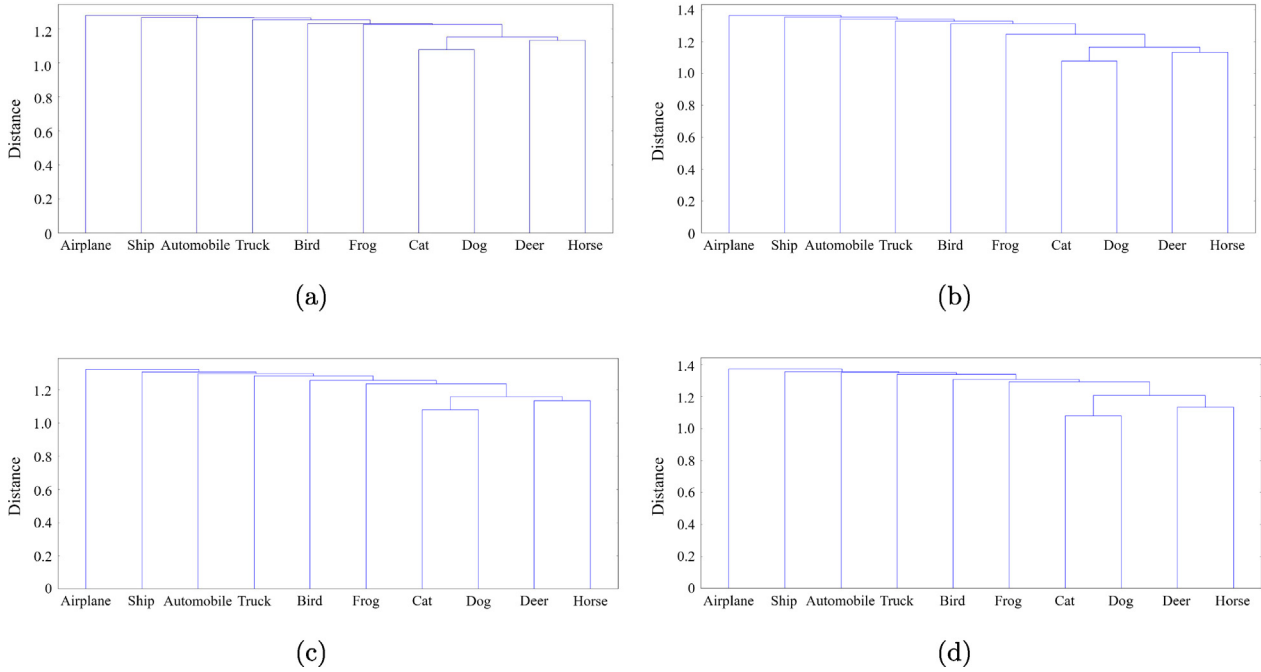
Fig. 6. Dendrograms for the CIFAR-10 dataset: single linkage (a), complete linkage (b), average linkage (c), and Ward linkage (d).

a value of imprecision tolerance degree γ. Based on the v-utility curves on the training set (Fig. 4), we can determine the optimal v for any given γ. As we consider all of the $2^{|\Omega|}$ acts, the three proposed classifiers always achieve average utilities of 1 when γ equals 1. The value of v has an apparent effect on the average utilities when γ is higher than 0.7. These curves show that parameter v should be carefully tuned to ensure optimal performance of the proposed classifier in set-valued assignments.

Fig. 5 shows the test average utilities and cardinalities of the evidential deep-learning classifiers as functions of γ with the optimal v. When the imprecision tolerance degree increases, the average cardinalities increase. This indicates that the proposed classifiers tend to perform set-valued assignments when given a large tolerance degree of imprecision. The test average utilities decrease slightly and then increase when γ increases. To explain this behavior, Table 7 provides four examples with their assignments and corresponding utilities. For the first example, the utility increases from 0 to 1 as γ becomes larger. However, for examples correctly classified when $\gamma = 0.5$ (#2 and #3), their utilities first decrease and then increase back to 1. The majority of examples in the CIFAR-10 testing set fall in the latter category. Therefore, the test average utilities decrease slightly and then increase when γ increases from 0.5 to 1.

The use of the DS and expected utility layers has an effect when there is a lack of evidence in the feature-extraction part. In Fig. 5, when γ is increased from 0.5 to 0.9, the largest gains in average utility are obtained by the evidential classifier with the NIN stages [34], whose feature extraction was found to be the worst among the three proposed classifiers since it achieved the minimum utility in the precise assignments (Table 5). Thus, the classifier with the NIN stages is more affected by the DS and expected utility layers than the other two classifiers. Therefore, we can conclude that the effects of DS and expected utility layers are more significant if there is a lack of evidence in the feature extraction part.

As shown in Fig. 5, the proposed model with a DS layer and an expected layer outperforms probabilistic CNN classifiers for set-valued classification. The average utilities of the proposed classifiers increase significantly when γ increases from 0.5 to 0.9. In con-

trast, the average utilities of the probabilistic CNN classifiers only increase sharply when γ increases from 0.9 to 1.0. This is evidence that the proposed classifiers make well-distributed set-valued classification based on the user's tolerance degree of imprecision, while the probabilistic CNN classifiers only assign samples to the multi-class sets when the tolerance is large. This phenomenon is caused by the use of DS and expected utility layers in the proposed classifiers. The DS layer tends to generate uniformly distributed masses when the features are not informative. As a result, the expected utility of a set-valued classification is the maximum among all acts, rather than the utility of a precise classification. This effect explains the superiority of the proposed approach for set-valued classification. However, the average utilities of the evidential classifiers are less than those of the probabilistic CNN classifiers for $\gamma = 0.7$. The reason is that the probabilistic CNN classifiers make few set-valued assignments for $\gamma = 0.7$, and the evidential classifiers are so cautious that they perform set-valued assignments for some instances that are correctly classified when γ is less than 0.7, such as #2 and #3 in Table 7.

In [62], we found that some ambiguous patterns always led the incorrect classification. Thus, we do not need to consider all of the 2^Ω acts, as mentioned in Section 3.4. In this experiment, the performances of the classifiers with partial acts are compared to those

Table 8
Set-valued assignment rates using the selected and 2^Ω acts (unit:%).

γ		0.5	0.6	0.7	0.8	0.9	1
CIFAR-10	Selected acts	0	0.52	1.74	13.24	19.62	52.04
	2^Ω acts	0	0.52	1.76	14.21	22.67	100
UrbanSound 8K	Selected acts	0	2.47	9.10	23.96	49.91	64.43
	2^Ω acts	0	2.47	9.71	28.74	55.62	100
SemEval-2010 Task 8	Selected acts	0	1.69	8.11	17.62	43.11	66.62
	2^Ω acts	0	1.69	8.57	27.71	52.77	100

Table 9
Proportions of samples correctly assigned to acts in 2^Ω and incorrectly assigned to selected acts, for different values of γ.

γ	0.5	0.6	0.7	0.8	0.9	1
CIFAR-10	0	0	0	0.18	0.47	2.87
UrbanSound 8K	0	0	0	0.42	0.95	6.62
SemEval-2010 Task 8	0	0	0.11	0.48	0.74	4.43

with all 2^Ω acts. Taking the evidential classifier with a network as in [14] as an example, we used the strategy introduced in Section 3.4 to generate the dendrograms, as shown in Fig. 6. When using Ward linkage [66], we get an inflection point to cut the dendrogram, with the CHI equal to 1.286 and the corresponding distance equal to 1.238. The selected multi-class sets consist of $\{cat, dog\}, \{deer, horse\}, \{cat, dog, deer, horse\}$, and $\{cat, dog, deer, horse, frog\}$ in the comparison study. Table 8 reports the testing rates of set-valued classification using the selected and 2^Ω acts. The rates of the classifiers with the selected and 2^Ω acts are close when γ is less than 0.9. Besides, the rates of the samples assigned correctly using 2^Ω acts but incorrectly using the selected

acts are small when γ is less than 0.9, as shown in Table 9. A set-valued assignment is regarded as correct if the multi-class set contains the true label. Thus, the proposed strategy is useful once an evidential classifier has a value of γ in the range of 0.5–0.9.

Novelty detection Fig. 7 displays the results of novelty detection using evidential deep-learning and probabilistic classifiers. The evidential deep-learning classifiers can assign outliers and a few of the known-class examples to set Ω when values of γ are between 0.7 and 0.9, while the probabilistic CNN classifiers cannot, which demonstrates that the proposed models outperform the probabilistic CNN classifiers for rejecting outliers together with ambiguous samples. This is due to the fact that, when the feature vector fed into the DS layer is far from all prototypes, the activations of the RBF units in the DS layer become close to zero, as shown by Eq. (5). As a consequence, all the mass functions m_i computed by Eq. (6) assign a large mass to set Ω, and so does their orthogonal sum m. The output of the DS layer thus reflects ignorance about the class of the input sample (whereas the probabilistic output of the softmax layer does not), leading to the assignment of the sample to set Ω.

We also applied McNemar's test with the CIFAR-100 and MNIST datasets, where outliers assigned to Ω are regarded as positive samples, and the others are negative ones. The results indicate
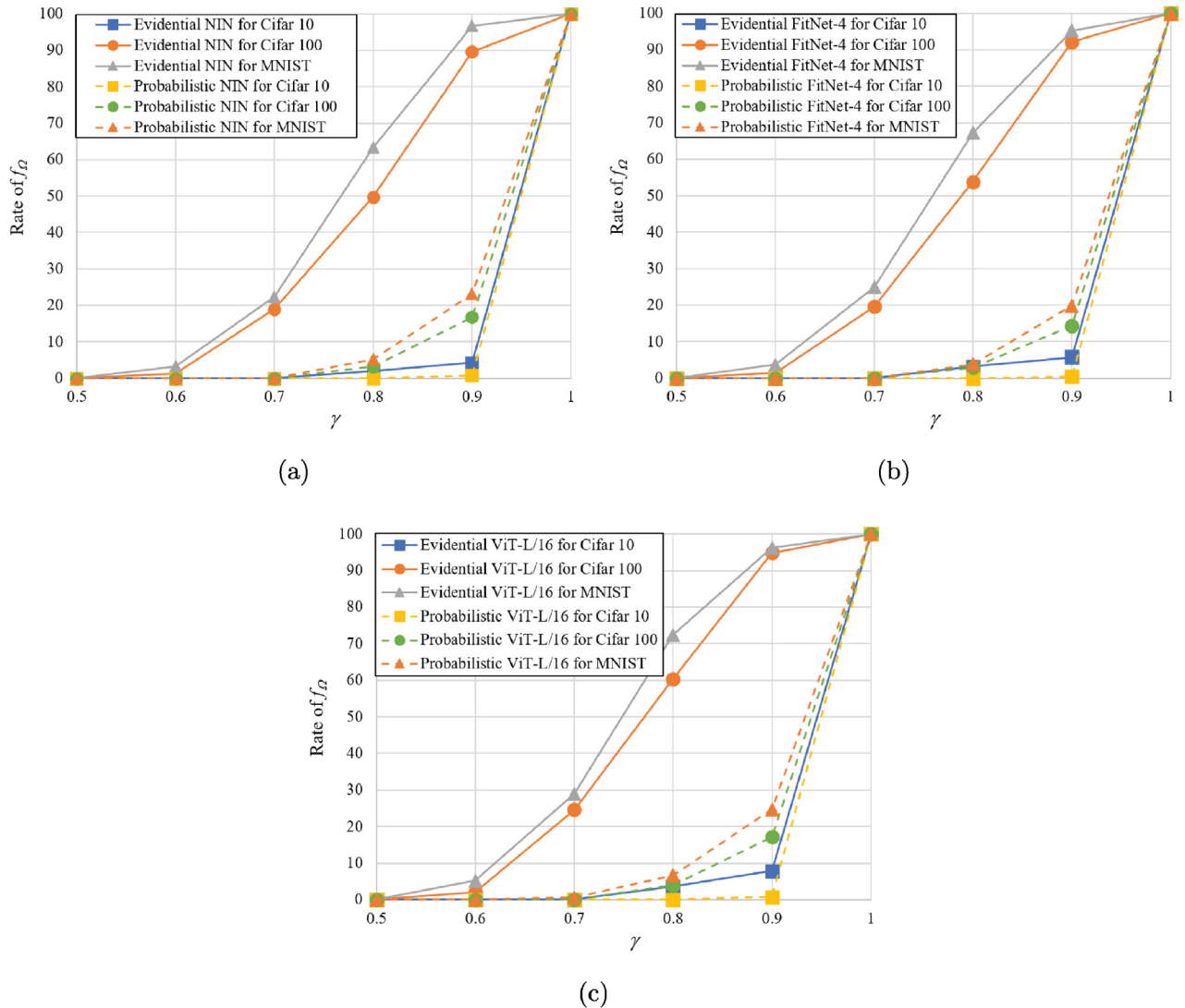

(a)

(b)

(c)

Fig. 7. Rate of f_Ω vs. γ for novelty detection in the image-classification experiment: NIN (a), FitNet-4 (b), and ViT-L/16 (c).

Table 10
The three baseline stages used for UrbanSound 8K.

Stage 1 [47]	Stage 2	Stage 3
	Pre-processing: clip, data augmentation, and segmentation	
	Input: 60 × 41 × 2	
57 × 6 Conv. 80 *ReLU*	57 × 6 Conv. 80 *ReLU* 1 × 1 Conv. 80 *ReLU*	29 × 3 Conv. 80 *ReLU* 29 × 3 Conv. 80 *ReLU*
	4 × 3 max-pooling stride 1 × 3 with 50% dropout	
1 × 3 Conv. 80 *ReLU*	1 × 3 Conv. 80 *ReLU* 1 × 1 Conv. 80 *ReLU*	1 × 2 Conv. 80 *ReLU* 1 × 2 Conv. 80 *ReLU*
	1 × 3 max-pooling stride 1× 3 without dropout	

Table 11
Test average utilities in precise classification on UrbanSound 8K.

Models	Stage 1 [47]		Stage 2		Stage 3	
	Probabilistic classifier	Evidential classifier	Probabilistic classifier	Evidential classifier	Probabilistic classifier	Evidential classifier
Utility	0.7132	0.7261	0.7164	0.7284	0.7210	0.7302
p-value (McNemar's test)	0.0234		0.0319		0.0365	
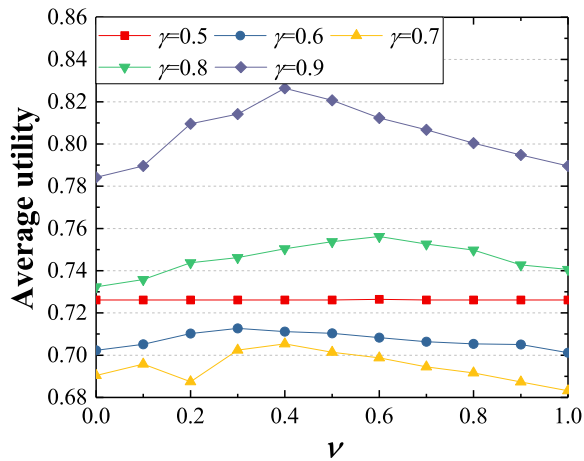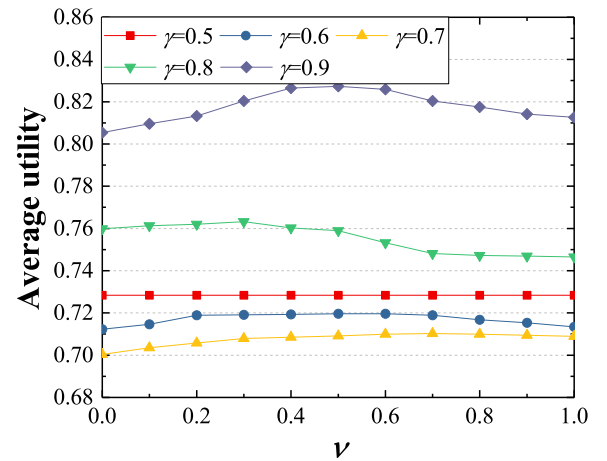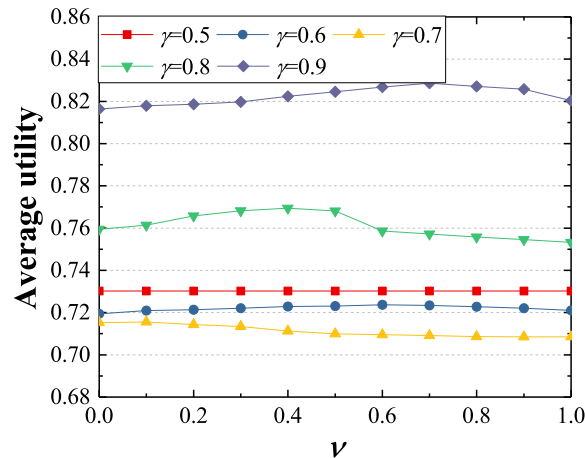
Fig. 8. Average utility vs. in *ν* for the proposed classifiers on the UrbanSound 8K dataset: Stage 1 (a), Stage 2 (b), and Stage 3 (c).
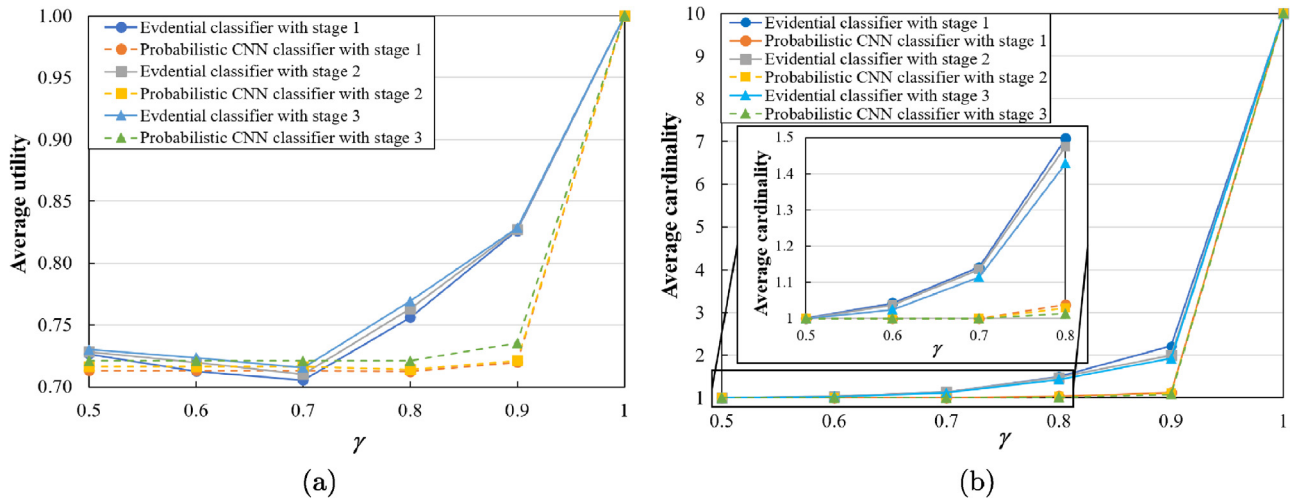
Fig. 9. Average utility (a) and average cardinality (b) vs. γ for the proposed classifiers and the probabilistic CNN classifiers on the UrbanSound 8K dataset.
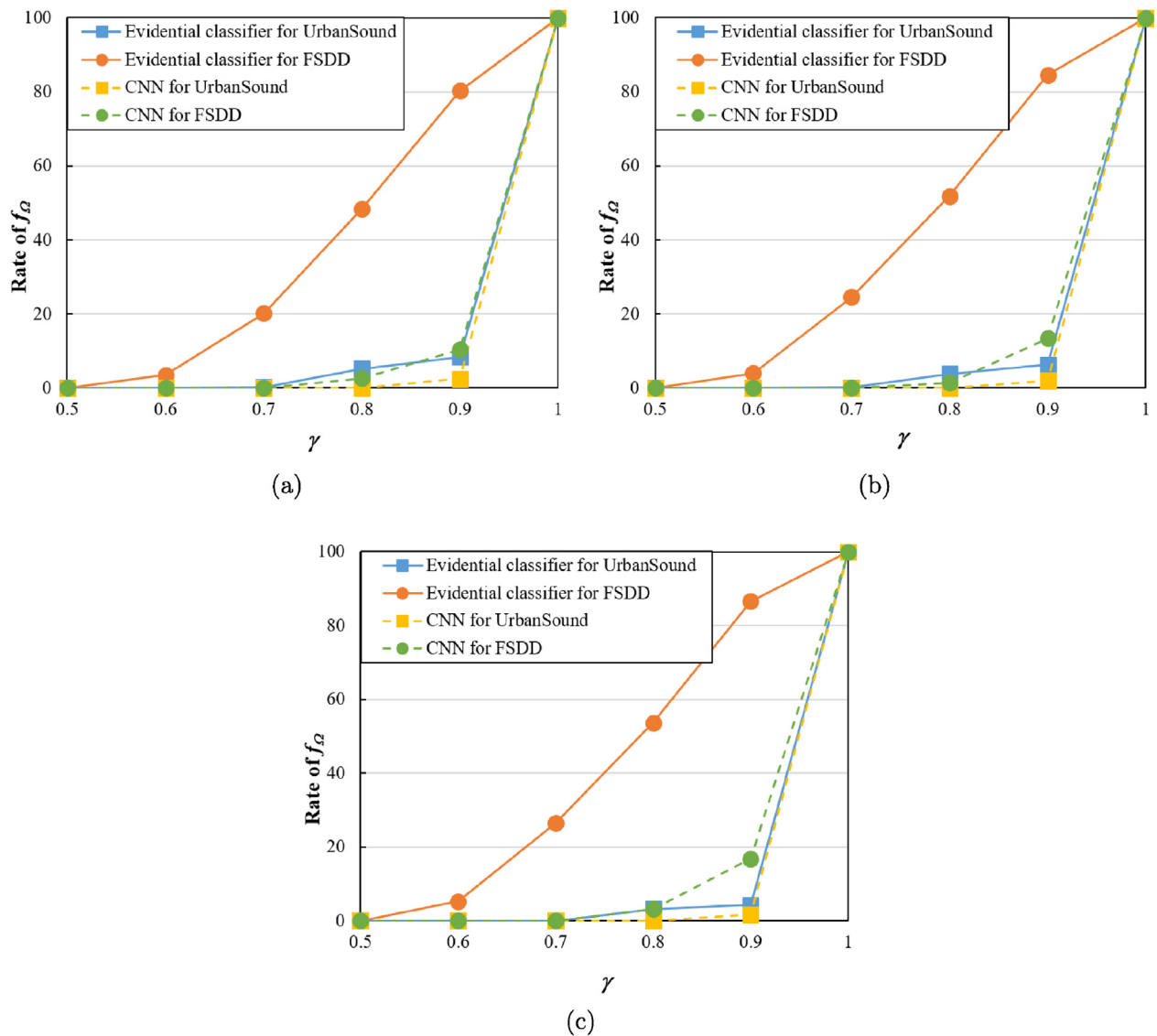

Fig. 10. Rate of f_Ω vs. γ for novelty detection in the signal-classification experiment: Stage 1 (a), Stage 2 (b), and Stage 3 (c).
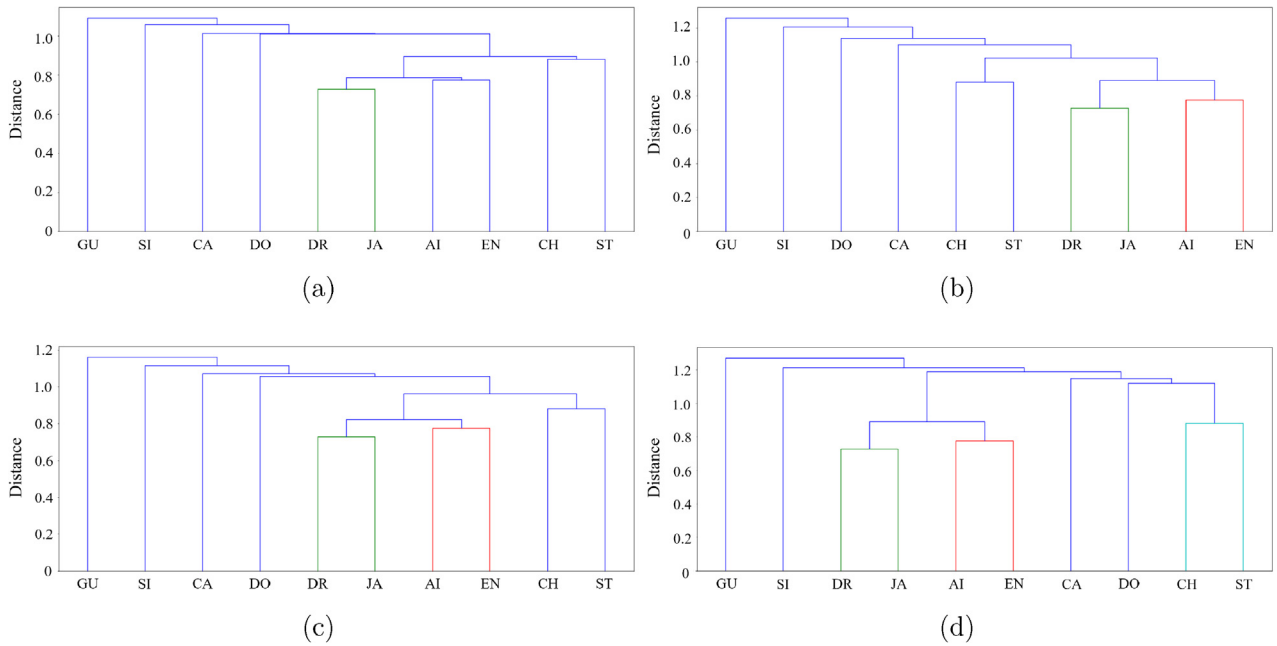
Fig. 11. Dendrograms for the UrbanSound 8K dataset: single linkage (a), complete linkage (b), average linkage (c), and Ward linkage (d).

Table 12
The three baseline stages used on SemEval-2010 Task 8.

Stage 1 [73]	Stage 2	Stage 3
Pre-processing: word representation		
Input: $50 \times 1 \times t$, in which t is the number of input sentences		
3×1 Conv. 200 *ReLU*	3×1 Conv. 200 *ReLU* 1×1 Conv. 200 *ReLU*	2×1 Conv. 200 *ReLU* 2×1 Conv. 200 *ReLU*
1×1 Conv. 100 *tanh*	1×1 Conv. 200 *tanh* 1×1 Conv. 100 *tanh*	1×1 Conv. 200 *tanh* 1×1 Conv. 100 *tanh*
1×1 mean-pooling stride 1×1		

Table 13
Test average utilities in precise classification on SemEval-2010 Task 8.

Models	Stage 1 [73]		Stage 2		Stage 3	
	Probabilistic classifier	Evidential classifier	Probabilistic classifier	Evidential classifier	Probabilistic classifier	Evidential classifier
Utility	0.8255	0.8347	0.8351	0.8425	0.837	0.8436
p-value (McNemar's test)	0.0301		0.0415		0.0430	

the use of the DS and expected utility layers has a distinct effect on novelty detection since all *p*-values are smaller than 0.001. However, none of classifiers performs well when γ is less than 0.7 since these classifiers favor precise decisions. The classifiers tend to reject outliers whose features are different from the known classes. For example, the proposed classifiers reject more samples in the MNIST dataset than in the CIFAR-100 dataset since the handwritten digits are very different from the patterns in the CIFAR-10 dataset.

4.3. Signal classification experiment

In the application of the proposed classifier on signal processing, we used the UrbanSound 8K dataset [51] composed of 8732 short (less than 4 s) excerpts of various urban sound sources (air conditioner (*AI*), car horn (*CA*), playing children (*CH*), dog bark (*DO*), drilling (*DR*), engine idling (*EN*), gun shot (*GU*), jackhammer (*JA*), siren (*SI*), street music (*ST*)) prearranged into 10 classes. The ratio between the training and testing set is about 3:1. We randomly selected 25% of the training samples as validation data. Free Spoken Digit Dataset (FSDD) [52], was used to evaluate the capacity of novelty detection in the signal classification experiment. FSDD is an audio/speech dataset with 2,000 recordings (50 of each digit per speaker) in English pronunciations.

The baseline stages in this experiment are shown in Table 10. The DS and expected utility layers show a significant difference in the precise classification as $0.01 < p < 0.05$ according to McNemar's test (Table 11). Similarly to CIFAR-10, this demonstrates that the performance of the proposed classifiers is better than those of probabilistic CNN classifiers for precise classification.

After determining the optimal v for each value of γ based on the v-utility curves (Fig. 8), we can compute the test average utilities

and cardinalities of the evidential deep-learning and CNN classifiers, as shown in Fig. 9. The proposed classifiers outperform the CNN models for the set-valued classification in the signal processing task. The proposed classifiers make more cautious decisions than do the probabilistic CNNs since it assigns ambiguous samples to multi-class sets. Additionally, the performance of the proposed classifiers exceeds those of the CNN classifiers in novelty detection (Fig. 10). The use of the DS and expected utility layers has significant effects on novelty detection as the results of *p*-value are close 0 according to McNemar's test.

For the testing of act-selection strategy, an inflection point was used to cut off the complete-linkage dendrogram [6] in Fig. 11, in which CHI is 2.198 and corresponding distance is 1.036. Thus, we selected partial multi-class sets including $\{DR, JA\}, \{AI, EN\}, \{CH, ST\}, \{DR, JA, AI, EN\}$, and $\{DR, JA, AI, EN, CH, ST\}$. From Tables 8 and 9, we can see that the strategy works well if γ is less than 0.9. This demonstrates that the proposed strategy is acceptable when the classifier has a reasonable γ.

4.4. Semantic-relationship classification experiment

For the semantic-relationship classification task, we used the SemEval-2010 Task 8 dataset [22]. It contains 10,717 annotated examples, including 8,000 training instances and 2,717 test instances. There are 10 semantic relationships in the dataset as cause-effect (*CE*), instrument-agency (*IA*), product-producer (*PP*), content-container (*CC*), entity-origin (*EO*), entity-destination (*ED*), component-whole (*CW*), member-collection (*MC*), message-topic (*MT*), and other (*O*). The approach to generate the validation set in this experiment is the same as those used in the experiments on the CIFAR-10 and UrbanSound 8K datasets. The FewRel dataset [21] with 100 semantic-relationship classes and 70,000 examples was used in novelty detection, in which the known-class examples were excluded in the experiment.

We referred to the stages shown in Table 12 to design the evidential deep-learning classifiers. In the precise classification, the use of DS and expected utility layers improves the test average utilities of the deep-learning models, as shown in Table 13. Thus, a DS layer and an expected utility layer instead of a softmax layer intro-
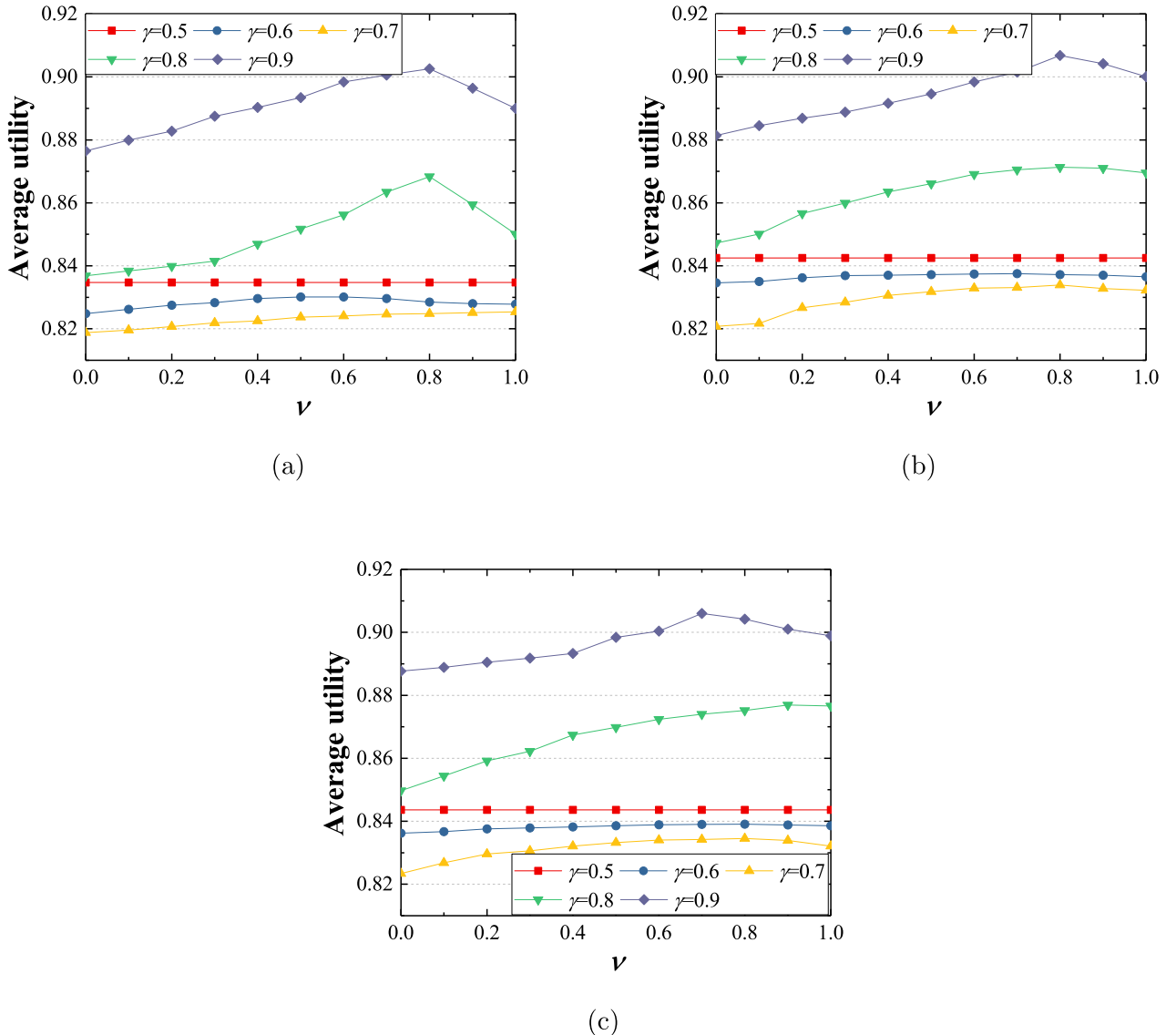

Fig. 12. Curves in *ν*-utility for the proposed classifiers on the SemEval-2010 Task 8 dataset: Stage 1 (a), Stage 2 (b), and Stage 3 (c).
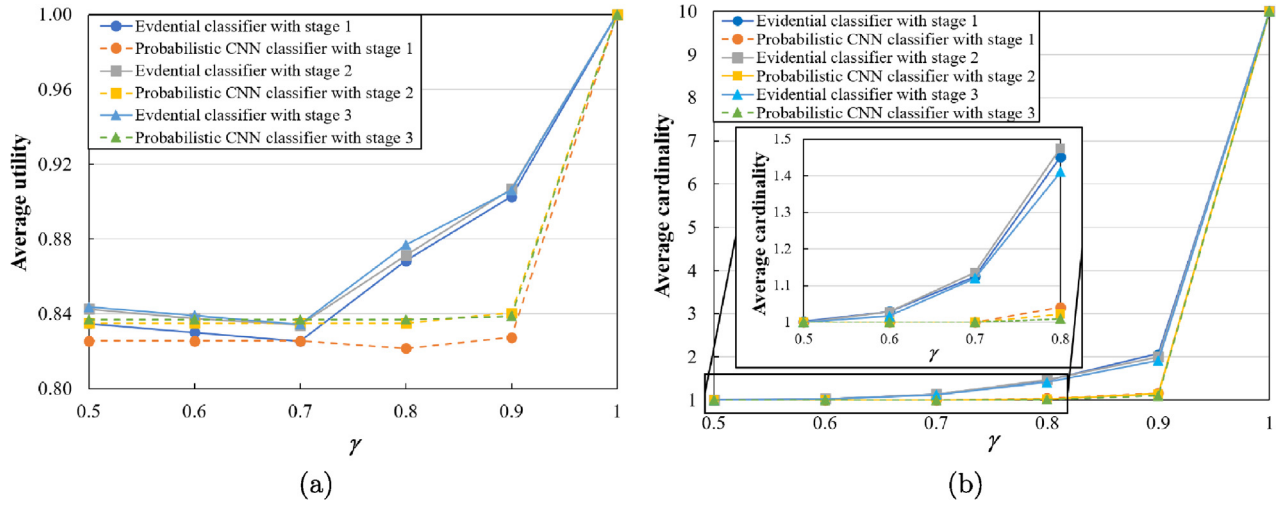
Fig. 13. Average utility (a) and average cardinality (b) vs. γ for the proposed classifiers and the probabilistic CNN classifiers on the SemEval-2010 Task 8 dataset.
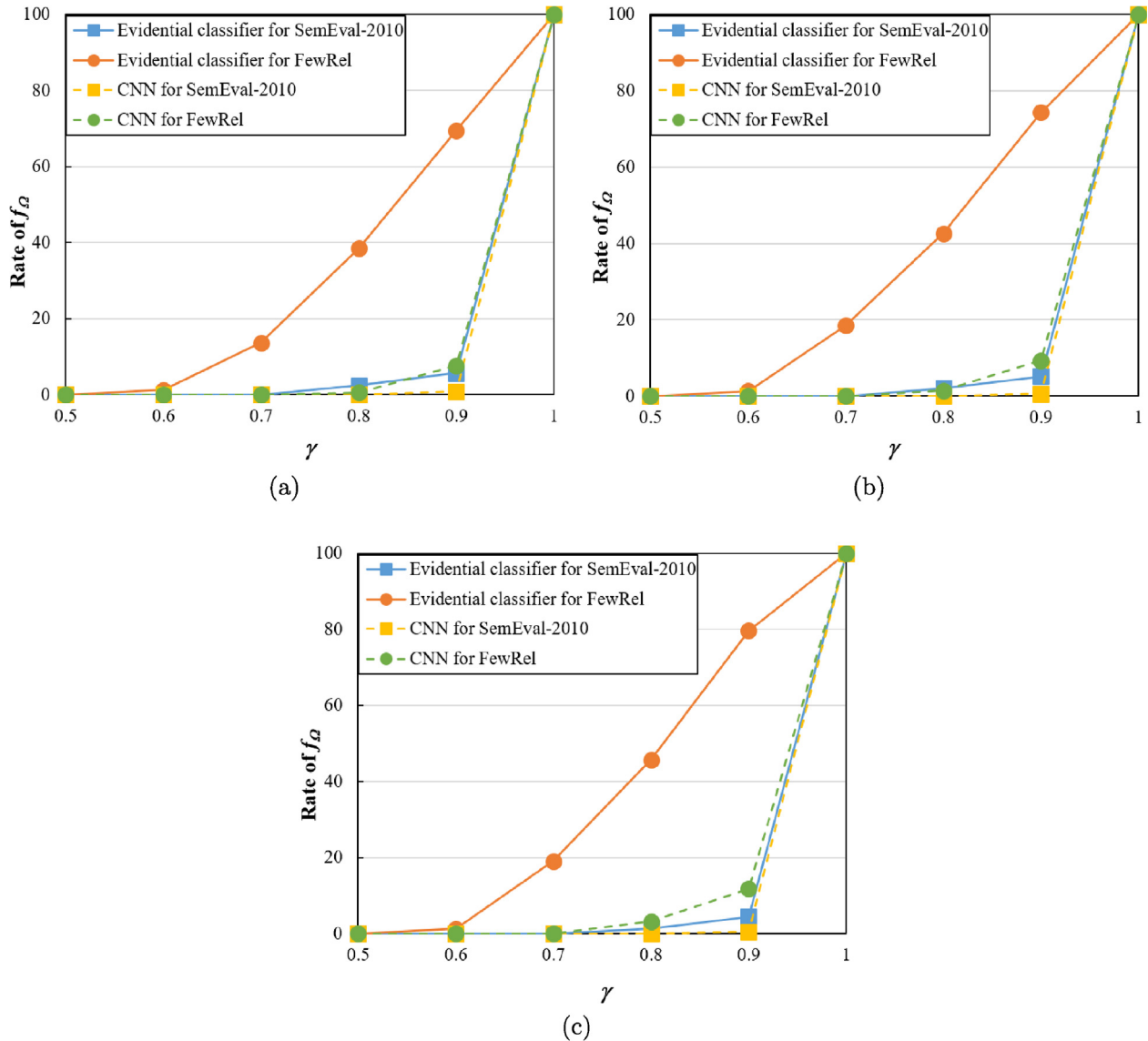

Fig. 14. Rate of f_Ω vs. γ for novelty detection in the semantic-relationship-classification experiment: Stage 1 (a), Stage 2 (b), and Stage 3 (c).
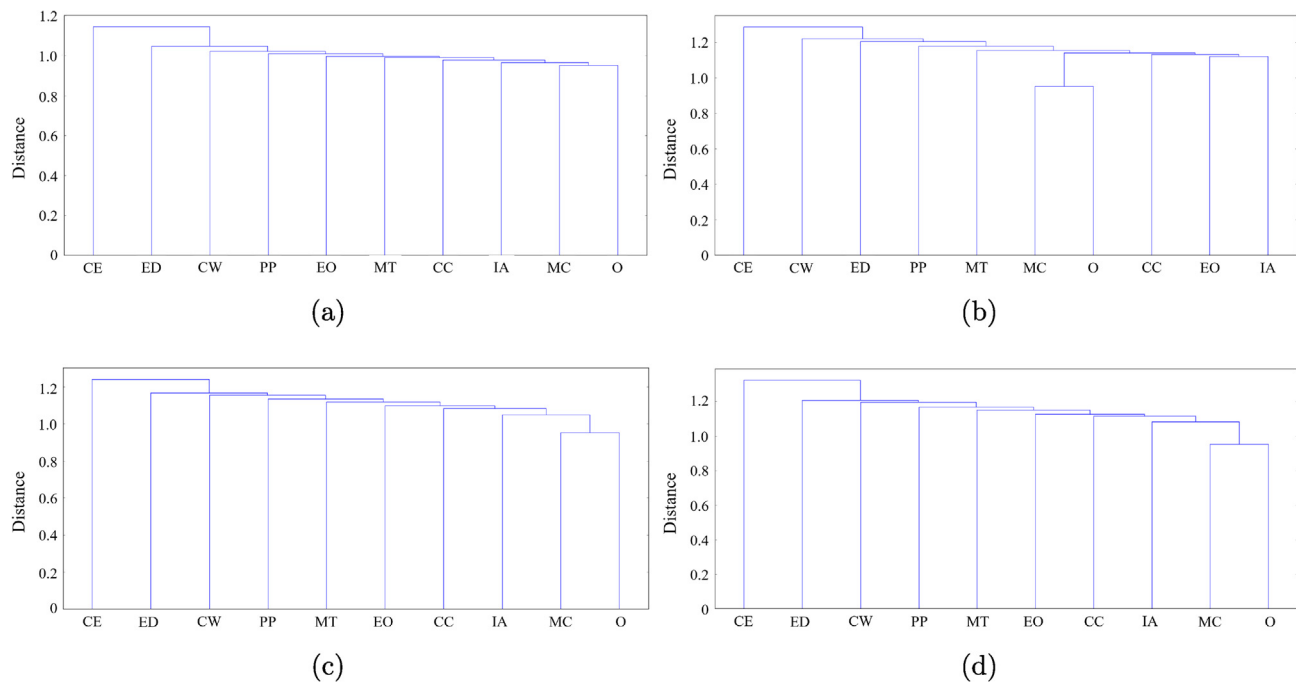
Fig. 15. Dendrograms for the SemEval-2010 Task 8 dataset: single linkage (a), complete linkage (b), average linkage (c), and Ward linkage (d).

duce a positive effect on the networks in the semantic-relationship classification.

The strategy for determining the optimal values of v in this experiment was the same as those in the CIFAR-10 and Urban-Sound 8K experiments (See Fig. 12). The test average utilities in set-valued classification of the two types of models are shown in Fig. 13, which demonstrate the superiority of the evidential deep-learning classifiers. Fig. 14 indicates the acceptable capacity of novelty detection in the evidential deep-learning classifiers. Similar as the CIFAR-10 and UrbanSound 8K dataset, the acts generated from the complete-linkage dendrogram (Fig. 15 and an inflection point whose CHI is 2.627 and a distance equals 1.107) works as well as the 2^Ω acts if the classifier has a suitable γ.

5. Conclusions

In this paper, we have presented a new neural network classifier based on deep CNN and DS theory for set-valued classification, called the evidential deep-learning classifier. This new classifier consists of several stages for feature representation, a DS layer to construct mass functions, and an expected utility layer to make set-valued assignments based on the mass functions. The classifier can be trained in an end-to-end way. Besides, we have proposed a strategy to select partial acts instead of considering all of them.

A major finding of this study is that the hybridization of deep CNNs and evidential neural networks by plugging DS and expected utility layers at the output of a CNN makes it possible to improve the performance of deep CNN models by assigning ambiguous patterns to multi-class sets. The proposed classifier is able to select a set of classes when the object representation does not allow us to select a single class unambiguously, which easily leads to incorrect classification in probabilistic classifiers. This result provides a novel direction to improve the cautiousness of deep CNNs for object recognition. The use of DS and expected utility layers also improves precise classification performance. The hybridization also makes it possible to reject outliers together with ambiguous patterns when the tolerance degree of imprecise is between 0.7

and 0.9. Additionally, the strategy of selecting partial multi-class acts works as well as that of considering all $2^{|\Omega|}$ acts.

Future work will focus on three main aspects. First, we will extend the proposed classifier to pixel-wise segmentation, where each pixel in an image must be assigned to one of the subsets of Ω. Secondly, other advanced evidential combination rules, such as contextual-discounting evidential K-nearest neighbor [13] will be studied to improve the performance of the proposed classifier. Finally, we will consider modifications of the model introduced in this paper to make it applicable to regression problems.

CRediT authorship contribution statement

Zheng Tong: Formal analysis, Methodology, Software, Writing - original draft. **Philippe Xu:** Methodology, Writing - review & editing. **Thierry Denœux:** Conceptualization, Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
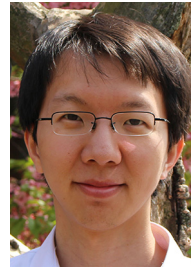
Acknowledgement

References

[1] Y. Bi, The impact of diversity on the accuracy of evidential classifier ensembles, International Journal of Approximate Reasoning 53 (4) (2012) 584–607.
[2] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics-theory and Methods 3 (1) (1974) 1–27.
[3] X.-L. Chen, P.-H. Wang, Y.-S. Hao, M. Zhao, Evidential KNN-based condition monitoring and early warning method with applications in power plant, Neurocomputing 315 (2018) 18–32.

[4] C.K. Chow, On optimum recognition error and reject tradeoff, IEEE Transactions on Information Theory IT-16 (1970) 41–46.

[5] A. Das, S. Ghosh, R. Sarkhel, S. Choudhuri, N. Das, M. Nasipuri, Combining multilevel contexts of superpixel using convolutional neural networks to perform natural scene labeling, in: Recent Developments in Machine Learning and Data Analytics, Springer, 2019, pp. 297–306.

[6] D. Defays, 01,An efficient algorithm for a complete link method, The Computer Journal 20 (4) (1977) 364–366.

[7] A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping, Annals of Mathematical Statistics 38 (1967) 325–339.

[8] T. Denœux, Analysis of evidence-theoretic decision rules for pattern classification, Pattern Recognition 30 (7) (1997) 1095–1107.

[9] T. Denœux, A neural network classifier based on Dempster-Shafer theory, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 30 (2) (2000) 131–150.

[10] T. Denœux, Decision-making with belief functions: a review, International Journal of Approximate Reasoning 109 (2019) 87–110.

[11] T. Denœux, Logistic regression, neural networks and Dempster-Shafer theory: A new perspective, Knowledge-Based Systems 176 (2019) 54–67.

[12] Denœux T., Dubois D., Prade H., Representations of uncertainty in artificial intelligence: Beyond probability and possibility, in: Marquis, P., Papini, O., Prade, H. (Eds.), A Guided Tour of Artificial Intelligence Research, vol. 1. Springer Verlag, Ch. 4, 2020, pp. 119–150.

[13] T. Denœux, O. Kanjanatarakul, S. Sriboonchitta, A new evidential k-nearest neighbor rule based on contextual discounting with partially supervised learning, International Journal of Approximate Reasoning 113 (2019) 287–302.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020. arXiv preprint arXiv:2010.11929..

[15] S. Du, S. Du, B. Liu, X. Zhang, Incorporating DeepLabv3+ and object-based image analysis for semantic segmentation of very high resolution remote sensing images, International Journal of Digital Earth (2020) 1–22.

[16] B. Dubuisson, M. Masson, A statistical decision rule with incomplete knowledge about classes, Pattern Recognition 26 (1) (1993) 155–165.

[17] A. Eitel, J.T. Springenberg, L. Spinello, M. Riedmiller, W. Burgard, Multimodal deep learning for robust RGB-D object recognition, in: Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2015, pp. 681–687.

[18] N. Guettari, A.S. Capelle-Laizé, P. Carré, Blind image steganalysis based on evidential K-Nearest Neighbors, in: Proceedings of the 2016 IEEE International Conference on Image Processing. Phoenix, USA, 2016, pp. 2742–2746.

[19] K. Guo, T. Xu, X. Kui, R. Zhang, T. Chi, ifusion: Towards efficient intelligence fusion for deep learning from real-time and heterogeneous data, Information Fusion 51 (2019) 215–223.

[20] T.M. Ha, The optimum class-selective rejection rule, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (6) (1997) 608–615.

[21] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, M. Sun, Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4803–4809.

[22] I. Hendrickx, S.N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz, SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals, In: Proceedings of the 5th International Workshop on Semantic Evaluation. Uppsala, Sweden, 2010, pp. 33–38..

[23] L. Hurwicz, The generalized Bayes minimax principle: a criterion for decision making under uncertainty, cowles Commission Discussion Paper 355, 1951..

[24] J.-Y. Jaffray, Linear utility theory for belief functions, Operations Research Letters 8 (2) (1989) 107–112.

[25] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, 2014, pp. 1746–1751..

[26] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Tech. rep., University of Toronto, 2009.

[27] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90.

[28] N. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and simile classifiers for face verification, in: Proceedings of the 12th International Conference on Computer Vision, IEEE Kyoto, Japan, 2009, pp. 365–372.

[29] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[30] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[31] B. Leng, Y. Liu, K. Yu, X. Zhang, Z. Xiong, 3D object understanding with 3D convolutional neural networks, Information Sciences 366 (2016) 188–201.

[32] G. Li, Z. Liu, L. Cai, J. Yan, Standing-posture recognition in human–robot collaboration based on deep learning and the dempster–shafer evidence theory, Sensors 20 (4) (2020) 1158.

[33] D. Liciotti, M. Bernardini, L. Romeo, E. Frontoni, A sequential deep learning application for recognising human activities in smart homes, Neurocomputing 396 (2020) 501–513.

[34] M. Lin, Q. Chen, S. Yan, Network in network, in: Proceedings of the 2014 International Conference on Learning Representations. Banff, Canada, 2014, pp. 1–10..

[35] Z. Liu, Q. Pan, J. Dezert, J.-W. Han, Y. He, Classifier fusion with contextual reliability evaluation, IEEE Transactions on Cybernetics 48 (5) (2018) 1605–1618.

[36] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[37] H. Ma, R. Xiong, Y. Wang, S. Kodagoda, L. Shi, Towards open-set semantic labeling in 3D point clouds: Analysis on the unknown class, Neurocomputing 275 (2018) 1282–1294.

[38] L. Ma, T. Denœux, Partial classification in the belief function framework, Knowledge-Based Systems 214 (2021) 106742.

[39] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, Recurrent neural network based language model, in: Proceedings of the 11th Annual Conference of the International Speech Communication Association. Chiba, Japan, 2010, pp. 1045–1048..

[40] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, S. Khudanpur, Extensions of recurrent neural network language model, in: Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Prague, Czech Republic, 2011, pp. 5528–5531..

[41] P. Minary, F. Pichon, D. Mercier, E. Lefevre, B. Droit, Face pixel detection using evidential calibration and fusion, International Journal of Approximate Reasoning 91 (2017) 202–215.

[42] P. Minary, F. Pichon, D. Mercier, E. Lefevre, B. Droit, Evidential joint calibration of binary SVM classifiers, Soft Computing 23 (2019) 4655–4671.

[43] D. Mishkin, J. Matas, All you need is a good init, 2015. arXiv preprint arXiv:1511.06422..

[44] S.M. Moosavi, A. Chidambaram, L. Talirz, M. Haranczyk, K.C. Stylianou, B. Smit, Capturing chemical intuition in synthesis of metal-organic frameworks, Nature Communications 10 (1) (2019) 1–7.

[45] T. Mortier, M. Wydmuch, K. Dembczyński, E. Hüllermeier, W. Waegeman, Efficient set-valued prediction in multi-class classification, 2019. arXiv preprint arXiv:1906.08129..

[46] M. O'Hagan, Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic, Twenty-Second Asilomar Conference on Signals, Systems and Computers. 2 (1988) 681–689.

[47] K.J. Piczak, Environmental sound classification with convolutional neural networks, in: Proceedings of the 25th International Workshop on Machine Learning for Signal Processing. IEEE Boston, USA, 2015, pp. 1–6.

[48] B. Quost, M.-H. Masson, T. Denœux, Classifier fusion in the Dempster-Shafer framework using optimized t-norm based combination rules, International Journal of Approximate Reasoning 52 (3) (2011) 353–374.

[49] K. Sakaguchi, M. Post, B. Van Durme, Efficient elicitation of annotations for human evaluation of machine translation, in: Proceedings of the Ninth Workshop on Statistical Machine Translation Baltimore, USA, 2014, pp. 1–11.

[50] J. Salamon, J.P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification, IEEE Signal Processing Letters 24 (3) (March 2017) 279–283.

[51] J. Salamon, C. Jacoby, J.P. Bello, A dataset and taxonomy for urban sound research, in: Proceedings of the 22nd ACM International Conference on Multimedia, Association for Computing Machinery, New York, USA, 2014, pp. 1041–1044.

[52] H. Salehinejad, Z. Wang, S. Valaee, Ising dropout with node grouping for training and compression of deep neural networks, in: 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2019, pp. 1–5.

[53] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (1) (2008) 61–80.

[54] F. Scarselli, Sweah Liang Yong, M. Gori, M. Hagenbuchner, Ah Chung Tsoi, M. Maggini, Graph neural networks for ranking web pages, in: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence Compiègne, France, 2005, pp. 666–672.

[55] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.

[56] R. Sibson, SLINK:an optimally efficient algorithm for the single-link cluster method, The Computer Journal 16 (1) (1973) 30–34.

[57] P. Smets, Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem, International Journal of Approximate Reasoning 9 (1) (1993) 1–35.

[58] R. Soua, A. Koesdwiady, F. Karray, Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 3195–3202.

[59] J.M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N.M. Donghia, C.R. MacNair, S. French, L.A. Carfrae, Z. Bloom-Ackerman, et al., A deep learning approach to antibiotic discovery, Cell 180 (4) (2020) 688–702.

[60] T.M. Strat, Decision analysis using belief functions, International Journal of Approximate Reasoning 4 (5–6) (1990) 391–417.

[61] Z. Tian, W. Shi, Z. Tan, J. Qiu, Y. Sun, F. Jiang, Y. Liu, Deep learning and dempster-shafer theory based insider threat detection, Mobile Networks and Applications (2020) 1–10.

[62] Z. Tong, P. Xu, T. Denœux, ConvNet and Dempster-Shafer theory for object recognition, in: Processing of the 13th international conference on Scalable Uncertainty Management, Springer International Publishing, Cham, France, 2019, pp. 368–381.

[63] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine learning. New York, USA, 2008, pp. 1096–1103.

[64] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of Machine Learning Research 11 (2010) 3371–3408.

[65] J. Wang, R. Ju, Y. Chen, G. Liu, Z. Yi, Automated diagnosis of neonatal encephalopathy on aEEG using deep neural networks, Neurocomputing 398 (2020) 95–107.

[66] J.H. Ward Jr, Hierarchical grouping to optimize an objective function, Journal of the American Statistical Association 58 (301) (1963) 236–244.

[67] P. Xu, F. Davoine, H. Zha, T. Denœux, Evidential calibration of binary SVM classifiers, International Journal of Approximate Reasoning 72 (2016) 55–70.

[68] Q. Xu, C. Zhang, B. Sun, Emotion recognition model based on the dempster–shafer evidence theory, Journal of Electronic Imaging 29 (2) (2020) 023018.

[69] R.R. Yager, On ordered weighted averaging aggregation operators in multicriteria decision-making, IEEE Transactions on Systems, Man, and Cybernetics 18 (1) (1988) 183–190.

[70] R.R. Yager, L. Liu, Classic Works of the Dempster-Shafer Theory of Belief Functions, vol. 219, Springer, Berlin, Heidelberg, 2008.

[71] B. Yuan, X. Yue, Y. Lv, T. Denoeux, Evidential deep neural networks for uncertain data classification, in: International Conference on Knowledge Science Engineering and Management, Springer, 2020, pp. 427–437.

[72] F. Yue, G. Zhang, Z. Su, Y. Lu, T. Zhang, Multi-software reliability allocation in multimedia systems with budget constraints using Dempster-Shafer theory and improved differential evolution, Neurocomputing 169 (2015) 13–22.

[73] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: Proceedings of the 25th International Conference on Computational Linguistics. Dublin, Ireland, 2014, pp. 2335–2344..

[74] C. Zhou, X. Lu, M. Huang, Dempster-Shafer theory-based robust least squares support vector machine for stochastic modelling, Neurocomputing 182 (2016) 145–153.

Philippe Xu received the B.S. and M.S. degrees in computer science from theÉcole Normale Supérieure de Cachan, France, in 2009 and 2011, respectively, and the Ph.D. degree from the Université de Technologie de Compiègne (UTC), France, in 2014. Since 2015, he has been an Associate Professor with the Computer Science Department, UTC, where he is carrying out his research at Heudiasyc UMR 7253 (UTC/CNRS). Since 2017, he has also been a member of the SIVALab, a joint laboratory between Renault, UTC, and CNRS. His research interests include information fusion, computer vision, and machine learning applied to localization and perception for autonomous vehicles.

Thierry Denœux is a Full Professor (Exceptional Class) with the Department of Information Processing Engineering at the University of technology of Compi?gne, France, and a senior member of the French Academic Institute (Institut Universitaire de France). His research interests concern reasoning and decision-making under uncertainty and, more generally, the management of uncertainty in intelligent systems. His main contributions are in the theory of belief functions with applications to statistical inference, pattern recognition, machine learning and information fusion. He has published more than 300 papers in this area. He is the Editor-in-Chief of the International Journal of Approximate Reasoning, and an Associate Editor of several journals including Fuzzy Sets and Systems and International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems.

Zheng Tong received his B.S. degree in civil engineering from Northeast Forestry University, China in 2015 and his M.S. degree in transportation engineering from Chang'an University, China in 2018. He is currently a Ph.D. student at the University of Technology of Compi?gne, France, under the supervision of Thierry Den?ux. His research interests concern pattern recognition, deep learning, and the application of belief function.