Modulation Recognition for Industrial Internet of Things Communication Signals Under Few-Shot Conditions Based on Attention Mechanism and Relation Network

Hualin Mu[1], Jie Zhang[2], Jerome Yen[3], Neal N. Xiong[4], Sergey M. Avdoshin[5]*

[1] College of Electronic Information and Communication, Huazhong University of Science & Technology, China
[2] College of Automation, Nanjing University of Science & Technology, China
[3] Faculty of Science and Technology, University of Macau, China
[4] Dept. of Computer Science, Southern New Hampshire University, USA
[5] School of Computer Engineering, Higher School of Economics University, Russia
mhl7588@163.com, zhangjie_njust@163.com, jeromeyen@um.edu.mo, n.xiong@snhu.edu, savdoshin@hse.ru

Abstract

In open, interference-prone scenarios, the scarcity of precisely annotated signal samples limits the application of deep learning–based modulation identification, which generally relies on extensive labeled data for stability. Relation Networks, as an emerging class of deep learning models, exhibit rapid convergence in few-shot learning tasks. Motivated by the fast convergence property of relation-based learning and practical deployment requirements, this study develops a deep learning framework for modulation recognition under few-shot conditions. Specifically, a lightweight attention mechanism is integrated into IQCNet to boost channel and temporal feature extraction. The resulting module acts as the embedding function in a Relation Network for few-shot modulation recognition. The proposed approach offers a reliable solution for modulation recognition of communication signals operating in complex and challenging electromagnetic scenarios.

Keywords: Deep learning, Few-shot learning, Modulation recognition, Communication signal, IIoT

1 Introduction

In Industrial Internet of Things (IIoT) and modern communication systems, the transmitter converts the original information into a signal suitable for the transmission medium through modulation. The receiver then needs to demodulate the signal to recover the information. This method is considered today as an integral part of the study of cognitive radio [1]. Communication systems have been developing, work on how to figure out the kind of signal (modulation identification) has changed a bit, it hasn't stuck strictly with figuring it all out by making comparisons for every possibility (maximum likelihood), it isn't focusing too much on old-fashioned ways to spot different things by hand (manually created feature separation). Now more interest is in using really smart computer programs called neural networks that seem to do pretty good stuff when they're taught something for a little while [2-3]. But the effectivity of these deep learning models usually comes at a huge price of labeled dataset and repeated training iterations. Labeled training datasets being low, such models end up overfitting towards the train samples which ultimately results in a drop in accuracy when classifying modulation. In practice, non-cooperative communication setting will always face great difficulty if trying to get enough labelled signal in some situations where environment, time variation of signals, or not enough previous knowledge exist. And these problems would even be worse if we were dealing with Industrial internet of thing (IIoT), since different types of machines will use the same frequencies, and so it makes getting information really tricky because lots of stuff changes quickly, and it's all spread out over lots of places.

Actually speaking we could see practical no co operative communication that would be the problem when you try to get your sample to be labelled it is restricted by the environment and restrictions of being secret for military use as well as time constraints when classifying during time of war and also a civil time which is where we have so much other information out there but still the same cost for labor and time for labelling those things. The study on few-shot modulation recognition has become important in order to deal with this situation [4-5].

In order to solve these kind of problems, we know from existing works, we can do this kind of problem solving by using the first way which is creating more train data via some transformation on our samples or use some meta learning framework. The former seeks to reduce over-fitting by increasing the size of the training set by making many kinds of transforms at the sample level or the feature level. Li et al. introduced an unsupervised meta-training stage to optimize data quality [6]. Patel et al. [7] and Gu et al. [8] both synthesized samples based on Generative Adversarial Networks (GANs). Wang et al. proposed a deep learning framework based on a Deep Residual Shrinkage Network and Generative Adversarial

*Corresponding Author: Jie Zhang; Email: zhangjie_njust@163.com
DOI: https://doi.org/10.70003/160792642026052703007

Network (DRSN-GAN), achieving a recognition accuracy of 92 percent on the RML2016.10a dataset (SNR 0dB) [9]. Perenda et al. leveraged channel and hardware impairment models for augmentation, generating diverse samples across varying noise conditions and enhancing modulation-classification robustness [10]. However, the pseudo-data generated by data augmentation has high requirements for the original dataset, limiting its applicability.

Relation Networks adopt the meta-learning approach, learning key information and learning abilities from a few-shot tasks. Within this framework, the decision process is carried out by evaluating the degree of similarity between query instances and the associated support samples defined in each newly formed task. In response to this, some scholars have adopted the idea of relation networks, using deep learning structures to build distance metric modules, so that the parameters in the metric module can be learned.

However, deploying modulation recognition in Industrial Internet of Things (IIoT) environments faces distinct challenges. Harsh industrial settings introduce severe electromagnetic noise and complex multipath effects, which obscure critical signal features. Furthermore, the diversity of proprietary protocols and high labeling costs result in extreme data scarcity, rendering traditional data-hungry deep learning models ineffective. These constraints necessitate a specialized approach capable of learning from limited samples while robustly filtering interference. Consequently, this study integrates an attention mechanism to isolate discriminative features amidst noise and adopts a few-shot learning framework to achieve high accuracy with minimal labeled data, directly addressing the critical bottlenecks of IIoT communications.

This enables the network to autonomously search for the optimal solution of distance metric parameters based on the classification results during training [11]. Li et al. proposed the AMR-CapsNet network, which achieved a classification accuracy of over 80 percent when trained on 3 percent of the dataset and SNR $\geq$ 2dB, a 20 percent improvement compared to convolutional neural networks [12]. Zhou et al. proposed an Automatic Modulation Classification Relation Network (AMCRN) with a recognition accuracy of over 90 percent, which is 10 percent-50 percent higher than classical methods when the SNR is > -2dB [13]. However, the above networks do not sufficiently exploit the intrinsic IQ-related characteristics and temporal structures of modulation signals. Moreover, under few-shot learning conditions, the representativeness of support samples may vary significantly across tasks, and treating all class prototypes equally may limit the discriminative capability of relation-based metric learning, especially under low signal-to-noise ratio and high inter-class similarity scenarios [14].

(1) To enhance the effectiveness of feature usage without increasing the computational burden of the original IQCNet architecture, an attention module based on the squeeze-and-excitation principle is incorporated into the network structure. This configuration simultaneously strengthens the distinction of features across channels and the responsiveness to temporal variations, allowing the network to more effectively identify the relative significance of each feature channel and to achieve higher recognition accuracy in environments with low signal-to-noise ratios [15].

(2) Incorporate a Relation network based metric learning framework for less labeled data available. The task-driven similarity learning done by it can also help decrease overfitting as well as increase the accuracy of our recognition when modulations happen with very few examples (few-shot situation).

(3) The comprehensive evaluation carried out with respect to the RML2016. 10a data set validates what we did was useful. From the results it is clear that when only 5%of the total labelled data are used, then SEIQCRnet model is capable of getting up to 91. 94 % avg class accuracy on SNR >= 0dB.

(4) A task-specific prototype weighting is employed so that we can directly reflect the changing weightings for different class prototype sets from one given task to another. In this approach, the stability of relation-based similarity learning is enhanced on few-shot scenario as well as when noise level is high by giving higher weightage to more informative class prototype.

The structure of this paper is as follows. The section 2 covers theory, like relations nets, attentions forms our model. And in the section three we will have the overall description on the SEIQCRnet architecture which consists of all details related to the features extraction part as well as the description on the whole distance metric module followed by task-based prototypes weightings schemes. The fourth one talks about experiments design and performance testing using RML2016.10a dataset and the results with different SNR condition and few-shot settings too.

2 Related Theories

Since the introduction of deep learning methods [16], these techniques have been widely adopted for communication signal modulation recognition and have maintained strong research attention. Deep learning architecture: Among various types of architecture we will see most is Convolution Neural network (CNN). The conceptual basis of CNNs originates from the backpropagation algorithm developed in 1986 [17]. Based on this, modern cnn has become a hierarchy network that contains convolution layers, pooling operations and fully connected layer which is the base for most current deep learning [18].

2.1 Relation Networks

Relation Networks are a novel type of convolutional neural network that adopts the meta-learning approach [19]. The core implementation of meta-learning is to construct different few-shot tasks, so that the meta-learner can learn to learn between different tasks, have the ability to distinguish between the similarities and differences between things, and generalize this ability to the target problem [20].

Within the meta-learning paradigm depicted in Figure

1, the training procedure is structured around task-oriented episodes. In each episode, an n-way k-shot task is constructed by randomly selecting n modulation categories from the dataset. For each category we choose, k labelled instances are chosen for constructing support sets and others will be assigned to queries.

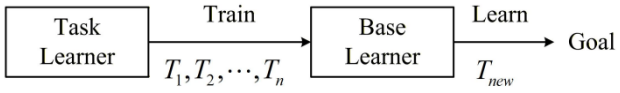


Figure 1. Meta-learning idea

The support set offers reference data for learning representations tailored to the specific task, while the query set is used to evaluate the model's ability to generalize within that task. The base learner is now able to more swiftly respond to novel modulation recognition tasks as a result of being trained over and over again on a myriad of episode that each had its own unique combination of classes; it has started developing the skill required for this. This task-level training strategy enables effective knowledge transfer across tasks and forms the foundation of the relation-based few-shot learning framework adopted in this work.

First, a task learner is established to train a large number of different tasks T_1, T_2, ..., T_n. The trained learner is the base learner [21]. When facing a new task with a small number of samples, the base learner can quickly search for an optimal parameter, so as to minimize the expected loss in each task:

$$\min_{\theta} E(L(Dt|\theta)), \tag{1}$$

Within meta learning framework, the tasks are usually organized as n way k shot, where n is the number of target class, k is the number of labeled sample given to each of them which is usually set to be 1,5 [22]. In this format every task is made up of a support set and a query set. Task formation: K samples were randomly drawn out of each of N classes to create a support set; Q additional samples per class were then chosen as queries [23].

During training, a sequence of such n-way k-shot tasks, referred to as episodes, is repeatedly fed into the network. This episodic learning strategy effectively reduces overfitting and enables the model to acquire transferable knowledge across different tasks [24].

Relation Networks adopt the meta-learning approach to perform learner learning and training under few-shot conditions. At the same time, through the self-learning module, it can accurately calculate the differences between different categories, so as to achiev effective classification of the target.

Relation network has three main parts: feature extracting part, concatenate and calculating distance part. The feature extraction module transforms raw input data into a latent representation space, and the distance metric module then measures the similarity between samples or tasks using these extracted feature embeddings.

$$r_{ij} = g_{\varphi}(Z(f_{\varphi}(x_i), f_{\varphi}(x_j)), \tag{2}$$

where x_i is the support set data, and x_j is the query set data. In the training process, the objective function is minimized to seek the optimal and φ, so that the network models the task optimally, and the network can quickly converge when accepting new small sample tasks. The objective function is as in Equation (3), where is the label of and is the label of x_j.

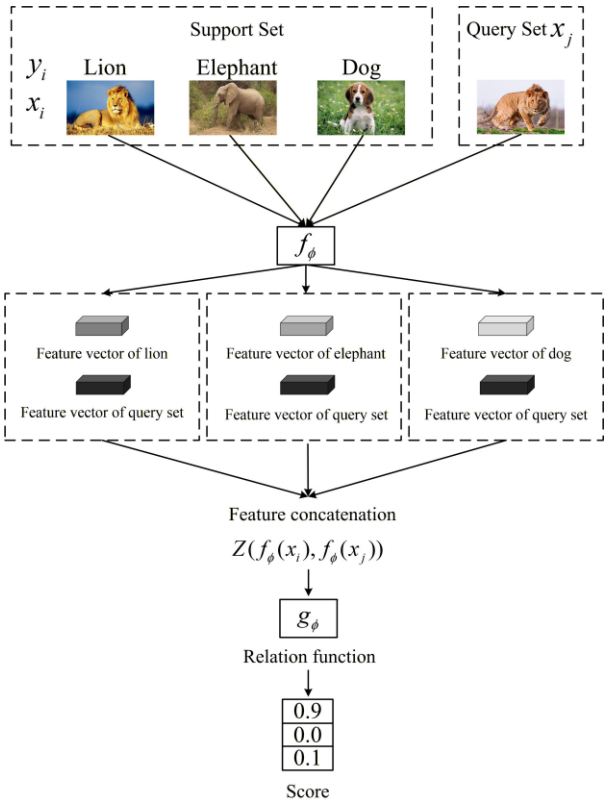


Figure 2. Basic process of relationship network

As depicted in Figure 2, the relation network processes data on a task-by-task basis under the few-shot learning paradigm. For each n-way k-shot task, both the support samples and query samples are first passed through a shared feature extraction network to obtain their corresponding embedding representations.

Each query sample's embedding is combined with the prototype embedding of every class derived from the support set. These combined feature pairs are input into the distance metric module, which models a nonlinear similarity function between the support and query pairs. The module produces a relation score normalized between 0 and 1, representing the similarity of the query sample to each modulation class. The predicted modulation category is assigned based on the class with the highest relation score. By training over multiple episodes, the relation network gradually acquires a task-adaptive similarity measure that can generalize to previously unseen few-shot modulation recognition tasks.

$$\arg\min_{\varphi,\varphi} \sum_{i=1}^{m}\sum_{j=1}^{n}[r_{i,j} - h(y_i, y_j)]^2, \tag{3}$$

$$h(y_i, y_j) = \begin{cases} 1, & y_i = y_j; \\ 0, & y_i \neq y_j. \end{cases} \tag{4}$$

2.2 Attention Mechanism

Attention Mechanism comes from how we understand people's eyes, and it has been adopted into RNN architecture for image classification so that it pays more attention to parts which have more recognizable information. In deep learning, adopting a similar attention mechanism can increase the algorithm's ability to focus on key features [25]. Attention mechanisms commonly adopted in deep neural networks can be categorized according to their operational scope into global attention, local attention, and internal attention mechanisms. Among these, global attention mechanisms are further differentiated based on the attention domain, including channel-wise attention, spatial attention, and hybrid attention models that jointly consider multiple feature dimensions [26].

3 Algorithm Framework

The overall architecture of the proposed SEIQCRNet, designed for IIoT-oriented few-shot modulation recognition within a relation network framework, is illustrated in Figure 3. Structurally, SEIQCRNet adheres to the standard relation network framework and is composed of three key components: a feature extraction module, a concatenation module, and a distance metric module.

SEIQCRNet conducts modulation recognition using a task-oriented few-shot learning approach. In each n-way k-shot task, a small set of labeled modulation signals is initially sampled to create the support set, while additional unlabeled signals are chosen to form the query set. Both the support and query samples pass through a common feature extraction module built on the SEIQCNet backbone to generate embeddings specific to the current task.

Class prototypes are derived from the embeddings of the support samples. To improve stability under limited data and varying signal-to-noise ratio (SNR) conditions, a task-specific prototype weighting mechanism is applied, which adaptively modulates the influence of each modulation class within the current task. For every query sample, its embedding is combined with the weighted class prototypes and fed into the distance metric module, producing relation scores that indicate similarity to each modulation class. The predicted modulation category is determined by the class with the highest relation score. Through repeated episodic training across many few-shot tasks, SEIQCRNet acquires a task-level similarity measure that generalizes effectively to IIoT communication settings with sparse labeled data.

To explicitly clarify the algorithmic workflow derived from Figure 2-Figure 5, the proposed SEIQCRNet operates through a structured, episodic procedure tailored for few-shot tasks. As illustrated in the overall framework (Figure 3), the algorithm initiates by constructing an N -way K -shot task, randomly sampling support and query sets from the IIoT signal pool. Both sets are then processed in parallel by the Feature Extraction Module (detailed in Figure 4 and Figure 5). In this stage, raw IQ signals first undergo 2×1 convolution to model I/Q correlations, followed by temporal convolutions and pooling to capture time-domain patterns. Crucially, the extracted features are refined by a dual-attention mechanism: a Channel Attention block (SE) recalibrates feature importance across channels, while a Temporal Attention block (T-SE) highlights discriminative time steps, ensuring robust embedding generation even under low SNR conditions.

Subsequently, the algorithm proceeds to the Metric Learning and Decision Phase. Embeddings from the support set are aggregated to form class prototypes, which are further optimized by a Task-Adaptive Prototype Reweighting mechanism to suppress noisy or ambiguous classes. Each query embedding is then concatenated with these weighted prototypes and fed into the Distance Metric Module (Figure 6). This module employs a series of convolutional layers to learn a nonlinear similarity function, outputting a relation score between 0 and 1 for each class. The final decision logic assigns the query sample to the modulation class yielding the highest relation score. This end-to-end process, iteratively optimized over numerous episodes, enables the network to learn a generalized metric space that effectively distinguishes modulation types with minimal labeled data.

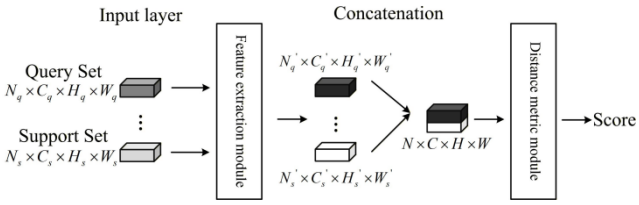


Figure 3. SEIQCRNet network structure

3.1 Feature Extraction Module

As illustrated in Figure 4, the feature extraction module operates as a task-level embedding algorithm shared by both the support set and the query set in each few-shot learning task. Given a batch of raw complex-valued IQ signal samples, the algorithm sequentially performs correlation modeling, temporal feature extraction, and feature refinement to generate discriminative representations.

Specifically, the algorithm does a convolution with a 2×1 filter at first to deal with both the In phase(I) part and Quadrature(Q) part of the signal. And that shows the existing link between the I channel and the Q channels explicitly as well as cuts down on the dimensions of the results so it can be processed further.

Next is some one dimensional convolution with kernel 1*3, to get high level time features in the signal sequence. So it allows us to detect these modulation

specific timeseries pattern in our i and q's. Then, we do a max pooling step to reduce the size of our feature map and remove any duplicate info.

Extracted features map go through Channel attetion so the result get some attention for every feature channels. The calibrated features then will be used as task specific embeddings and will be input to next relation-based similarity module.

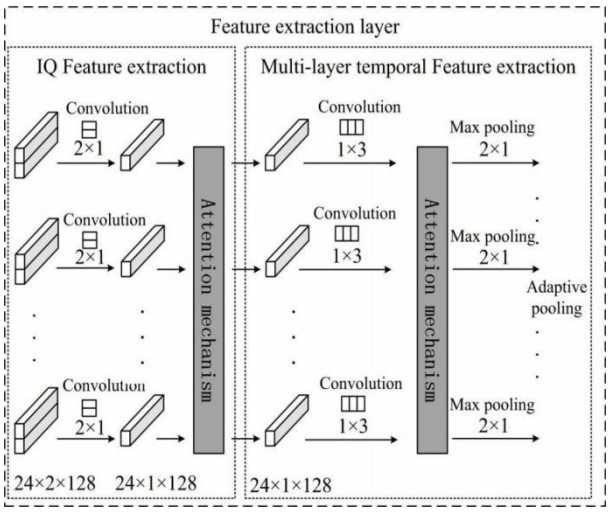


Figure 4. SEIQCRNet network feature extraction module

The Feature Extraction Model gets the training sets presented by N way, K shot with the Support Set and Query Set. Query set: $N_q \times C_q \times H_q \times W_q$, while the support set has $N_s \times C_s \times H_s \times W_s$ and:

$$\begin{cases} N_q = n \cdot k_q, N_s = n \cdot k_s; \\ C_q = C_s; \\ H_q = H_s; \\ W_q = W_s. \end{cases} \tag{5}$$

Under these terms, n means the different levels of modulating, k_q is the amount of times something has been sampled in a query set and k_s means how many times it has been sampled under a support set. The feature extraction part uses light-weighted, computationally cheaper IQCNet architecture [27]. The 2*1 convolutions is applied for raw I&Q signals with respect to I&Q correlation features so that both the I&Q dependencies can be accounted and reduced computations as well. And then we use several 1x3 temporal convolution kernel to get more deeper time-domain feature, after that do the max pooling operation to decrease the shape of our feature map.

In original IQCNet we have every single one of those things being extracted at the very same level of extraction without any thought paid towards whether or not different channels would depend on each other in terms of producing useful features, thus resulting possibly with less optimal features and a less recognizable result at difficult levels [28]. Overcoming this problem is done through using a channel attention which allows us to make the model aware about its own importance on the individual channels, thus improving the amount of useful information that can be taken from these channels.

To keep the lightness of the first network, we add the Squeeze-and-Excitation (SE) channel attention unit from [29], we get the SEIQCNet structure. As can be seen in Figure 5, SE contains three parts: squeeze, excitation and scale. The squeeze stage accumulates global context info over time. Excitation stage: inter-channel dependencies are modeled to generate dynamic channel weight. These weights are applied to the original feature maps in the scale stage, so that the informative channels are strengthened and the noise-sensitive channels are weakened.

More precisely, Adaptive Average Pooling reduces every feature channel to one single representative number to acquire the whole response of that channel [30]. These channel descriptors go through two successive learnable transforms within the excitation stage to model channel dependency and produce a set of importance weights. Normalize and scale them using the non-linearity. Then apply it on the original feature map with scale operation. It is ensured by means of this procedure that those channels carrying important information with respect to recognizing modulations get to be more highlighted; however less important ones and even noisy types get minimized.

With that channel attention included in the process of the feature extraction pipeline, it can be seen that we are able to modify our feature representations on an as need basis as to adapt itself toward a certain specific task and therefore improve robustness when faced with less shots and also fewer signals as compared to noise [31].

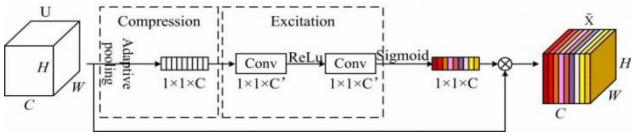


Figure 5. Implementation process of channel attention module

(1) Squeeze: Adaptive Pooling Function (Adaptive Average Pooling) is adopted so as to make the form of the output of this function be 1×1; thus, after going through the previous Convolution Layer, we have the feature-map dimension reduced from C×H×W into C×1×1 making sure that the receptive fields in all the channels are compressed up till the maximum extent possible across the whole region.

(2) Excitation: The two FC layer to be applied for change no.of channel out and do the excitation task. Compared to the regular version a 1×1 con is put in place of the FC layers so there aren't as many dimension changes to tensors when doing comp. As well, in order to improve representational capacities of feature weights we will use a channel expansion instead of a channel reduction from one layer to another.

(3) Scale: Excitation operation yields normalized weights and is carried out element-wise multiplication of it over original feature map thus obtaining scale feature map as result. In this map every channel gets its own weight and

value near 1 means very important for feature extraction but values closer to zero indicate not so important.

(4) To better understand time-changing features which play an essential part in modulation identification, a Temporal Squeeze-and-Excitation (T-SE) is brought up. In the modulation signal, temporal structure like symbols' changes and transient waveforms' change may also bring useful difference in time domain, but these are non-uniformly spreaded in time. The traditional temporal convolutional layers can grasp the local temporal characteristics, but it cannot make the significance judgement about every single moment on purpose. Different from the channel-focused SE block, we propose Temporal Squeeze-Excitation (T-SE) block which operates over the temporal axis to emphasize the relevant part of the sequential signal.

After squeeze – excitation – scale design was proposed and applied; the T - SE module does the same process for one time-step by firstly conducting a temporal squeeze which applies global average pooling on every single time step along its entire channel axis of a given tensor to make it collapse into a compressed version of shape [C,T] -> [1,T]. This is how the encoding works so we're getting just this sort of like overall activation strength per time position. And then we do a kind of temporal excitation operation that just takes those really tiny little 1x1 convs and does some intermediate expansion stuff to quickly pick up on connections between different times, and the sigmoid activation normalizes those values into temporal attention weightings.

In scales, temporal attention weight gets spreaded over channels and then gets multiplied with the original feature map elements, thus bringing focus to the important temporary parts leaving the less relevant or noisy areas behind. T-SE module is right after the channel attention module, so it can have channel relevancy and tempora relevance modeled at the same time but at no increase in compu. cost.

The algo part, it's basically the featextraction that gives us this kind of shared space for both supports as well as queries inside SEIQCRNet: they end up embedded together all right. This way through using the combination of both IQ model correlations and temp convolutions along with seq atten reweights we will have these kinds of features that we want from those sorts of relations type ML tasks on few shot datasets.

3.2 Distance Metric Module

Support set and Query set go through feature extration then it is converted to feature tensor with shape (N×C×H×W):

$$\begin{cases} C_s = C_q'; \\ H_s = H_q'; \\ W_s = W_q'. \end{cases} \tag{6}$$

The concat operation merges the feature vectors of all support samples with that of the query sample to form one matrix of size N × C × H × W, with:

$$\begin{cases} C = C_s' + C_q' = 2C_s' = 2C_q' \\ H = H_s' = H_q'; \\ W = W_s' = W_q'. \end{cases} \tag{7}$$

Concatenate support – query feature pairs go into the distance metric module for estimating similarities. This module captures relations among supports and queries by passing concatenations of the two types of data through many convolution layers that then get transformed via nonlinearity and normalization operations. Sigmoid function is used on output layer in order to bring the relation scores between [0,1] [32], and then choosing class with the most score as prediction.

Specifically, the convolutional layer has 32 channels with a 4 × 1 kernel that captures important patterns in joints [33]. Batch normalisation is placed between the conv layer and activation layer, it improves speed of convergences and decreases chance of overfitting, use relu and sigmoid functions as the acti and cla func respectively [34]. We max pool for more salient relations and to downplay unimportant differences.

Referring to Figure 6 we have seen, that the module of the distance metric is responsible for acting like the learner to give similarity for every single few shot task. We use the embeddings from the feature extraction module to make support-query feature pairs with concatenation so that they can be used for joint learning.

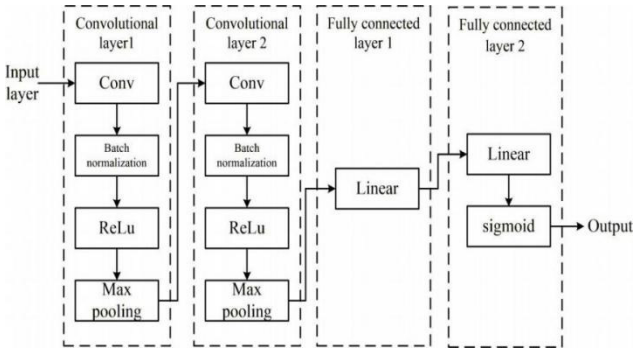


Figure 6. Structure of the distance measurement module

For each query sample, it's embedding gets combined with the embedding for every single class prototype gotten out of the support set. Then we pass these support-query pairs through several convolutional layers to get some joint representation, and in this process it will be extracted by the relation pattern among all kinds of support samples and query samples [35]. Batch normalization and nonlinear activation functions are used to stabilize training and enhance the expressiveness of the metric function. A subsequent max-pooling step highlights important relational features while suppressing redundant or irrelevant information.

Finally, a sigmoid activation normalizes the output of the distance metric module, producing relation scores in the range [0, 1]. These scores provide a quantitative

measure of the similarity between each query sample and the associated modulation classes.

3.3 Task-Adaptive Prototype Reweighting Module

In few-shot modulation recognition, especially within non-cooperative IIoT settings, only a very small number of labeled samples are available for each modulation class. Consequently, the representativeness of support samples can vary widely across different tasks and SNR conditions. Conventional relation-network approaches generally assume that all class prototypes contribute equally to similarity estimation, an assumption that often breaks down under low SNR or when inter-class similarities are high [36].

To overcome this issue, a task-adaptive prototype reweighting mechanism is integrated into the SEIQCRNet framework. Let $f(\cdot)$ represent the embedding function implemented by the SEIQCNet feature extractor. In an n-way k-shot task, the prototype for the (c)-th modulation class is calculated as the mean embedding of its support samples. So we do this reweighting, where you take the influence of every single one of your examples – or your prototypes – and then you adjust that for the situation that we're doing this job with and we get more like the answer you can use on really hard stuff.

Improves resistance towards changes within same classification, mixing with others and confusing by noise that often is found when actually sending data over IIoT and it doesn't add much trouble to the usual relation structure.

$$\mathrm{p}_c = \frac{1}{k}\sum_{i=1}^{k} f(\mathrm{x}_{c,i}), \tag{8}$$

Whereas $\mathbf{x}_{c,i}$ is the ith support sample from class c and $\mathbf{p}_c$ is the prototype of this class in embedding space [36]. And then we have some prototypes that can represent kind of like the main modulation that happens for just that current little chunk that you were doing right now for this few shot learning setup.

Rather than regarding all class prototypes to be equaly imporant, we introduce a lightweight but task aware weighting function that captures the relative weight of each prototype. More precisely, an importance score is assigned to every single prototype as follows:

$$s_c = g(p_c), \tag{9}$$

where $g(\cdot)$ represents a learnable mapping function using a small neural model like a MLP or a 1×1 conv [37]. Then each class prototype's task adaptive weight is got by a softmax normalization.:

$$w_c = \frac{\exp(s_c)}{\sum_{j=1}^{n}\exp(s_j)}, \tag{10}$$

which ensures that the weights across all modulation classes are non-negative and sum to one. Through this adaptive weighting strategy, the network can automatically emphasize more representative and reliable prototypes while suppressing those that are more susceptible to noise or ambiguity in the current task.

During relation-based similarity computation, the embedded feature of a query sample $\mathrm{x_q}$ is first obtained as $\mathrm{f(x_q)}$. The weighted prototype $\mathrm{w_c p_c}$ is then concatenated with the query embedding and fed into the distance metric module introduced in Section 3.2 to compute the relation score:

$$r_c = D([f(\mathrm{x}_q), w_c \mathrm{p}_c]), \tag{11}$$

Here, $D(\cdot)$ represents the relation network, and $[\cdot,\cdot]$ denotes the concatenation operation [38]. The resulting relation scores are normalized with a sigmoid function to produce similarity values between 0 and 1. The modulation label of a query sample is predicted by choosing the class with the highest relation score:

$$\hat{y} = \arg\max_c r_c. \tag{12}$$

Training wise we are optimising simultaneously over all of the embeddings, relations etc to try and minimise the difference that's there right now. That's this predicted values vs what was observed for every sample which is MSE. Task-adaptive prototype reweighting module increases the number of parameters only slightly but the training process remains as the same of SEIQCRNet framework.

Without changing the way we train our episodic stuff, or how big the network is, this module helps make our estimate of similarity better by changing how much the different things that represent classes should be considered. Dynamicly modify how to change each prototype with every tasks so that we can increase recogition under small number samples and noise so that all over SEIQCRNET system is increased in term sof performace.

4 Simulation Experiments and Performance Analysis

Authors need to interpret their own conclusions according to that before; authors' conclusion can be seen by the reader by seeing other author's result on what is being read. The expected as well as the unexpected should be considered, and what they add to our understanding of the field needs to be stressed as well. Also we have the conversation about the practical, theory meaning of this research and put it under the big science application scope. In addition, directions for future research might also be mentioned, where some unsolved problems and limits of the present work are identified and possible ways to extend or improve the method are suggested.

4.1 Dataset

The data set is prepared based on the RML2016. 10a data set, which consists of 11 modulation types: 8 types of digital modulations (BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, PAM4) and 3 types of analog modulations (WBFM, AM - SSB, AM - DSB). 1,000 two channel I/Q samples for each modulation type and the samples contain 128 points each. We run these simulations at 20 different SNRs - starting from -20 dB, incrementing by 2 dB up until we reach 18 dB - so there will be 220,000 in total. Summarized as Dataset partion is listed out in Table 1.

Table 1. Dataset partition details

Dataset name	Modulation category	Data volume
Training set	8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK	6 ×20 × 1000
Support set	PAM4, QAM16, QAM64, QPSK, WBFM	5 ×20 × 100
Test set	PAM4, QAM16, QAM64, QPSK, WBFM	5 ×20 ×900

Dataset splitting method is that it chooses samples randomly from all 6 modulations for training data; it would be like the source domain. The other 5 types of modulation, samples of which will be utilized as support set as well as test set. In this case, for every one of the snr's there would be about 10 percent of its entirety out of those five kinds of modulation taken at random for the support set representing a kind of few shot target domain type situation, leaving behind the other 90% in the form of the query set against which to measure how well the network performed modulation classification.

After dividing the three types of datasets, the n-way k-shot form is adopted for task extraction. The training set is constructed by directly sampling corresponding support and query sets to generate a sequence of training tasks, ranging from task 0 to task x, which together form the meta-training dataset. Within each task, the support set adopts an n-way k-s-shot sampling strategy to generate multiple support vectors, while the test set employs an n-way k-q-shot strategy to extract multiple query vectors. The support vectors and query vectors are combined to form multiple test tasks to form a meta-test set for testing the recognition accuracy of the network. The specific division and extraction method is shown in Figure 7.

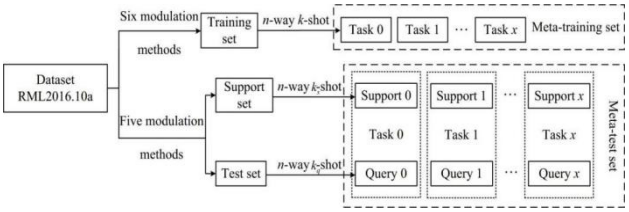


Figure 7. Data set partitioning extraction process

4.2 Experimental Conditions

The experiment consists of two processes: training and testing. The flowchart of the whole process is shown in Figure 8. First, the data of 6 modulation styles are used to train the model to obtain learning ability, and then the data of 5 modulation styles are tested to obtain the quantitative value of the network recognition accuracy.

During the experiments, both the feature extraction network and the relation network employ a stepwise learning rate schedule to improve training efficiency, with identical parameter settings: an initial learning rate of 0.001, adjustment every 10,000 episodes, and a reduction factor of 0.5. For network evaluation, a single task input cannot fully reflect recognition performance, so testing is repeated over 100 episodes, and the average accuracy is reported as the final result. An early stopping strategy is applied during training, terminating optimization if the loss does not decrease over 100 consecutive episodes.

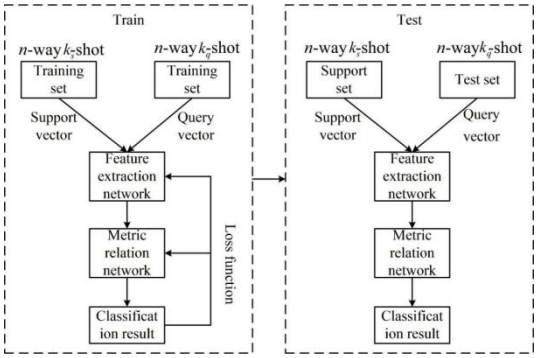


Figure 8. The entire experimental process

Specifically, the meta-training phase runs for 80,000 episodes using the Mean Squared Error (MSE) loss function and the Adam optimizer. To verify the effectiveness of the proposed SE and T-SE modules independently, we also conducted regular supervised training experiments. In this setting, the network was trained for 30 epochs with a batch size of 400, utilizing the Cross-Entropy loss and the Adam optimizer (learning rate = 0.001). All experiments were implemented in PyTorch 1.12.1 with Python 3.9 and executed on an NVIDIA GeForce GTX 1060 Laptop GPU.

Besides episodes and few-shots, we have some regular supervised training as well so we can get an idea if any attention works in isolation. With respect to the configuration, the network is being trained with the use of mini - batches for the total of 30 times under normal mini - batch improvement using the quantity which forms a single batch to be 400. We will use cross entropy loss with adam at lr 0.001. It was only aimed to test contributions by SE and T-SE module but doesn't interfere with episodic few-shot learning setup.

4.3 Effectiveness of Feature Recognition Module Structure

Figure 9 Classification accuracy vs. SNR for IQCNet (blue) and the proposed SEIQCRNet (red) under 5-way 1-shot setting. The main plot shows performance across SNRs from -20 dB to 18 dB; the inset zooms in on the high-SNR region (0–20 dB), highlighting SEIQCRNet's

consistent advantage, especially above 0 dB. Both models improve with increasing SNR, but SEIQCRNet achieves higher accuracy at all levels, demonstrating enhanced robustness in low- and medium-SNR regimes.

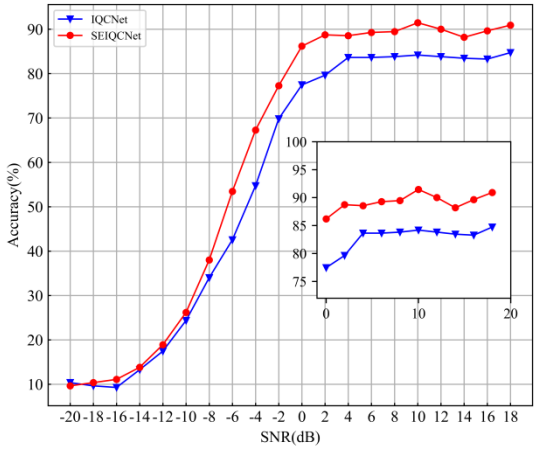


Figure 9. Comparison chart of network accuracy before and after improvement

For the purpose of confirming how much impact can be made with adding a lighter and simpler channel attention scheme on this networks recognition abilities a new adaptation pooling layer (named SEIQCNet), has been incorporated into our network framework right after our feature identification area so it would have individual conversations about recognizing the signals features. We select data randomly from the RML2016.10a dataset with a ratio of 8:1:1 as the train, val, and test set. Network recognition precision comparison shown in figure nine.

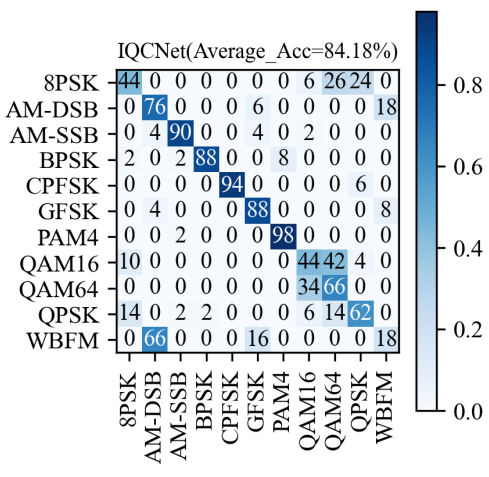
Overall, the SEIQCNet architecture incorporating attention mechanisms demonstrates superior recognition accuracy compared with the original IQCNet. Although SEIQCNet exhibits slightly lower performance at extremely low SNR levels below −16 dB, it shows a clear advantage across most other SNR conditions, particularly when the SNR exceeds 0 dB.

SEIQCNet's recognition accuracy at an SNR level at minus two decibels was near eighties, more than nineties if SNR went higher by six decibels.

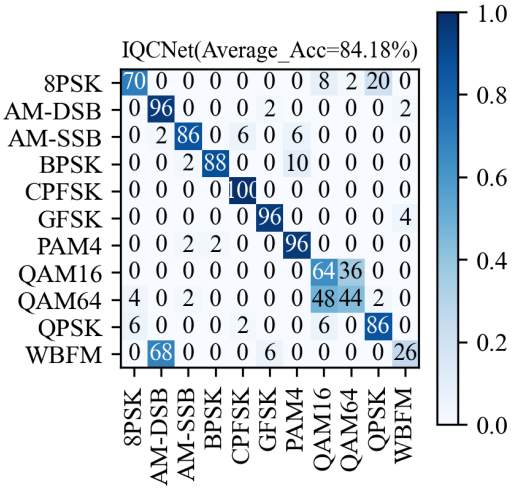
To emphasize enhancement between different kinds of modulations, we consider recognition results of all modulation types (SNR = −2 dB, 0dB, and 10 dB) as presented in Figure 10. According to this it can be said that the model is able to recognize AM-DSB and WBFM signals much better than it recognizes 16QAM signal. It has been observed that best performances can be seen from the −2 dB.

To confirm that our enhancement works, we compare SEIQCNet to several current top-of-the-line models: CLDNN, GRU2, IC-AMCNet, and MCNet. These comparison results are shown in Figure 11. Figure 11 Comparison of modulation recognition accuracy versus SNR among five deep learning models: CLDNN, GRU2, ICAMCNET, MCNet, and the proposed SEIQCRNet (red), under 5-way 1-shot few-shot setting. The main plot covers SNR from -20 dB to 18 dB; the inset zooms into the high-SNR region (0–20 dB) for finer comparison. SEIQCRNet consistently outperforms all baselines across the entire SNR range, particularly above 0 dB, demonstrating superior feature discrimination and robustness in low-data regimes.

In terms of the experiments, the results tell us that SEIQCNet does better than every other model in every case within the whole SNR range. In the whole section from -20db up to +18 db, the SEIQC-net has obtained an overall average correct recognition rate amount of 60. 92percent; this was more successful than compared against CLDNN, GRU2, IC-AMCnet, and MCNet with difference ratios at five point seven six percent, two point eight four percent, four point zero zero percent, and four point seventy-seven percent respectively. So it seems like what we can do to improve on our ability to distinguish between things in different situations is paying extra attention during the extraction process.



(a)



(b)

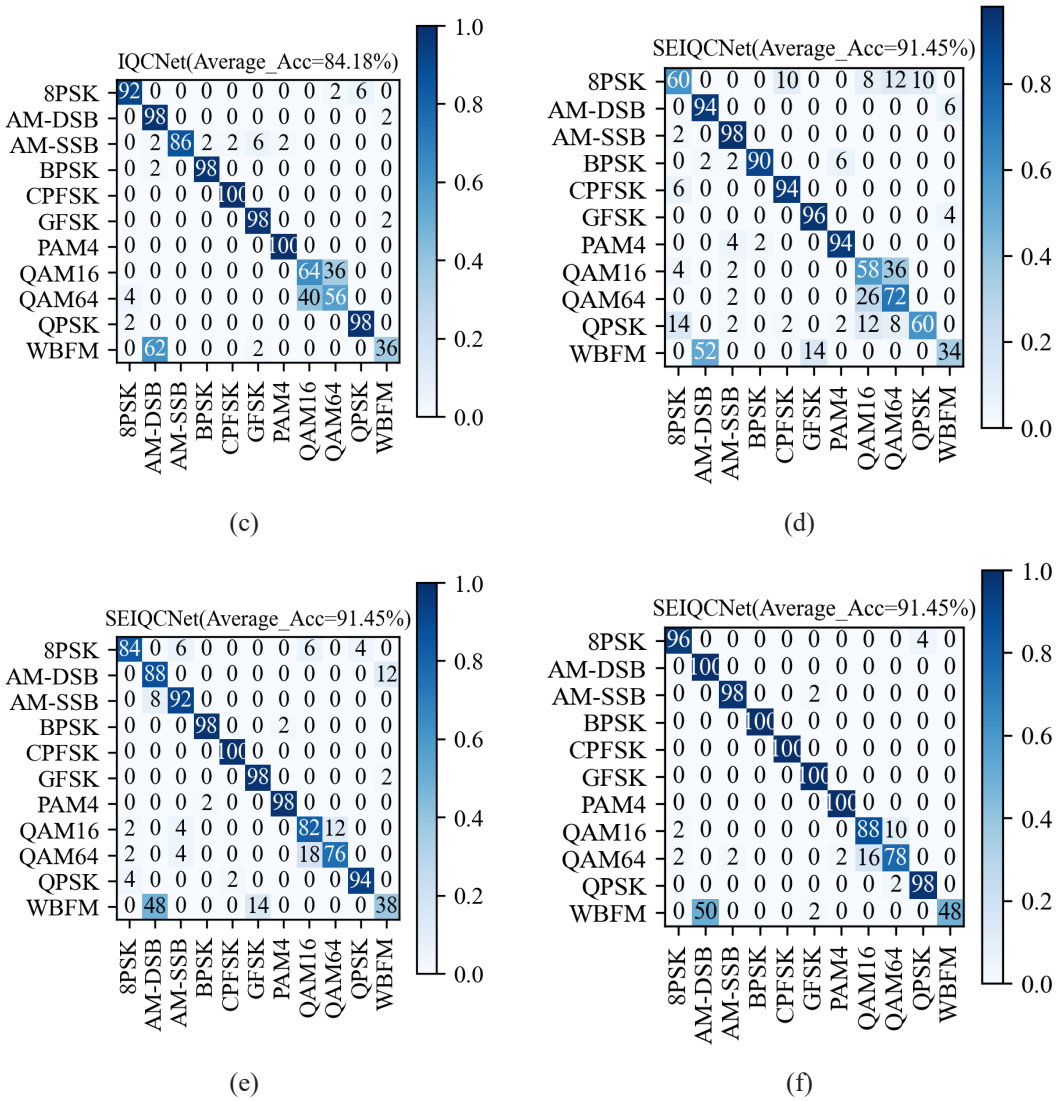


Figure 10. Confusion matrices of the network before and after improvement under three different signal-to-noise ratios: (a) IQCNet (SNR=-2dB); (b) IQCNet (SNR=0dB); (c) IQCNet (SNR=10dB); (d) SEIQCNet (SNR=-2dB); (e) SEIQCNet (SNR=0dB); (f) SEIQCNet (SNR=10dB)

In the case where there is no medium or low SNR from 0dB to 18dB, SEIQCNet has more benefits, it has been tested for the accuracy up to be as much as 89.24% and this was exceeded than the equivalent other models at 7.84%, 4.28%, 4.64% and 7.42%. Additionally, within the low-SNR range from −20 dB to −2 dB, SEIQCNet achieves the highest recognition accuracy among all compared models, highlighting its improved robustness to noise interference. The average recognition accuracy of different networks for various SNR ranges is presented in Table 2.

Table 2. Average recognition accuracy of different network for various SNR ranges

Model	-20~18dB	-20~-2dB	0~18dB
CLDNN	55.16%	28.93%	81.40%
GRU2	58.08%	31.20%	84.96%
IC-AMCNet	56.92%	29.24%	84.60%
MCNet	56.15%	30.47%	81.82%
SEIQCNet	60.92%	32.60%	89.24%

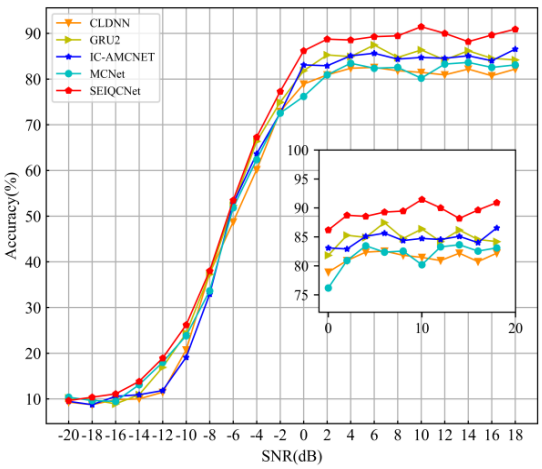


Figure 11. Comparison of different network recognition accuracy results

4.4 Effectiveness of the Overall Network SEIQCNet Structure

To assess the effectiveness of the SEIQCRNet architecture, its recognition accuracy is compared directly with that of the SEIQCNet network. Among them, the SEIQCNet network performs sample extraction during the experiment. In the process, two extraction methods are adopted: one is to extract 10 percent, 50 percent and 80 percent of the sample number under the target domain dataset category divided in Section 3.1, and the other is to extract the same amount of samples as the target domain 10 percent, 50 percent and 80 percent of the data volume under the 11 types of modulation recognition styles in Section 3.3. The datasets generated by the two extraction methods contain the same number of samples, but differ in the total number of modulation classes, with 5 classes in one case and 11 in the other. The comparison results are presented in Figure 12.

Figure 12 Impact of support set coverage on modulation recognition accuracy under varying SNR conditions. The red curve denotes the full-support baseline (100% classes with K-shot samples). Other curves represent scenarios where only X% of classes have support samples (either K-shots or 1-shot per class). Results show that SEIQCRNet maintains robust performance even when ≤50% of classes are supported, especially at medium-to-high SNR. The inset highlights stability differences in the 6–18 dB range, where reduced support leads to increased variance.

As illustrated in Figure 12, SEIQCRNet, SEIQCNet-10%-5classes, and SEIQCNet-10%-11classes are trained on identical data volumes—each utilizing 5% of the RML2016.10a dataset. The figure shows that recognition accuracy steadily improves as the number of training samples increases. Notably, under the same sample budget (i.e., 10% of the target-domain dataset), SEIQCNet—lacking the relation network module—exhibits lower classification performance compared to SEIQCRNet, which incorporates the relation-based framework.

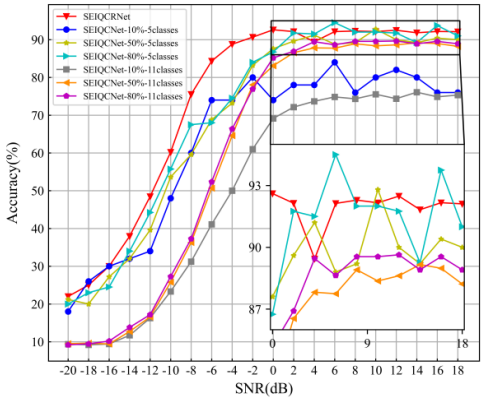


Figure 12. Comparison of network accuracy before and after introducing the relationship network framework

Under few-shot conditions (10% labeled samples per class) across a signal-to-noise ratio (SNR) range from −20 dB to 18 dB, SEIQCRNet achieves average accuracy gains of 11.11% (for 5-class classification) and 23.96% (for 11-class classification) over SEIQCNet. When SNR exceeds 0 dB, SEIQCRNet attains an average recognition accuracy of 91.94%, outperforming SEIQCNet by 13.54% (5-class) and 17.91% (11-class) in the same SNR regime. These results underscore that integrating a relation network significantly enhances modulation recognition capability in data-scarce scenarios.

This performance improvement is partially attributable to the proposed task-adaptive prototype reweighting strategy, which refines similarity estimation by emphasizing more discriminative and representative prototypes within each few-shot learning task. Furthermore, this module enhances model robustness by dynamically modulating the influence of individual class prototypes at the task level—a feature particularly advantageous under low-SNR conditions or when inter-class similarities are high.

To evaluate the trade-off between performance and model complexity, we compare the number of trainable parameters across several representative modulation recognition architectures. Although the incorporation of attention mechanisms and relation-based metric learning introduces moderate computational overhead, it remains critical to assess whether the proposed architecture retains a lightweight profile relative to existing approaches.

Table 3 summarizes the parameter counts of CLDNN, GRU2, IC-AMCNet, MCNet, and the proposed SEIQCNet. Models such as IC-AMCNet and GRU2 employ deep recurrent or multi-branch structures, resulting in substantially higher parameter counts. In contrast, SEIQCNet adopts a compact convolutional backbone augmented with channel-wise attention, effectively curbing model complexity.

As shown in Table 3, SEIQCNet exhibits markedly fewer parameters than all benchmark models. Specifically, its parameter count is reduced by over two orders of magnitude compared to IC-AMCNet, yet it achieves superior recognition accuracy—as evidenced in Section 4.3—demonstrating that the performance gains are not achieved through excessive model scaling or depth inflation.

While the integration of attention mechanisms incurs a marginal increase in computational cost relative to the baseline IQCNet, the overall architecture remains highly efficient. This favorable balance between accuracy and model compactness renders SEIQCNet particularly well-suited for resource-constrained Industrial Internet of Things (IIoT) communication environments, where low-latency inference and minimal deployment footprint are essential.

Table 3. Comparison of model parameter quantities

Model	CLDNN	GRU2	IC-AMCNet	MCNet	SEIQCNet
Parameters	76683	151179	1263627	120267	19403

4.5 Data Generalization Experiment

To overcome the limitations associated with single-dataset evaluation and to further assess robustness and

generalization capability, additional experiments are conducted on the RML2018.01a dataset. Compared with RML2016.10a, RML2018.01a is larger in scale, includes a greater variety of modulation schemes, and spans a broader SNR range, making it more suitable for evaluating generalization performance.

The RML2018.01a dataset contains 24 modulation types, encompassing both digital and analog schemes such as OOK, ASK, PSK, APSK, QAM, AM, FM, and GMSK. For each modulation type, 4096 two-channel IQ samples of length 1024 are generated across 26 SNR levels from −20 dB to 30 dB with a step size of 2 dB, resulting in a total of 2,555,904 samples. Detailed dataset information is provided in Table 4.

Table 4. RML2018.01a dataset information

Modulation category	Data volume	SNR range/ step size	Data format
OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK	2555904	-20dB~30dB/2dB	2×1024

To ensure consistency with the experimental protocol used in previous sections, the RML2018.01a dataset is partitioned following the same modulation-level division strategy. Specifically, 12 modulation types are randomly selected as source-domain classes, and all corresponding samples are used for training. The remaining 12 modulation types are treated as unseen target-domain classes and are used to construct the support and query sets.

For each target-domain modulation type and SNR level, 10% of the samples are randomly chosen to create the support set, and the remaining 90% are used as the query set to evaluate performance. Compared with conventional deep learning scenarios with abundant data, this setting reduces the target-domain dataset size to approximately 5% of the original data volume. The detailed partitioning scheme is summarized in Table 5.

Following dataset partitioning, an n-way k-shot task sampling strategy is applied to assess recognition performance in few-shot scenarios. The detailed experimental results under different settings are presented in the following subsections.

Among metric-based meta-learning methods, prototypical networks (PNs) and relation networks (RNs) are two widely adopted frameworks. PNs rely on fixed distance metrics in the embedding space, which may limit discriminative capability across heterogeneous tasks, whereas RNs learn task-adaptive similarity functions that provide greater flexibility in modeling inter-class relationships.

Table 5. Dataset partition of RML2018.01a

Dataset	Modulation category	Data volume
Training set	OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK	12×26×1000
Support set	16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK	12×26×100
Test set	16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK	12×26×900

Using the constructed few-shot tasks, comparative experiments are carried out on the RML2018.01a dataset to evaluate the performance of relation network–based (RN) and prototypical network–based (PN) methods. The overall recognition results under few-shot task sampling are illustrated in Figure 13.

As shown in Figure 13, both frameworks exhibit similar performance trends across the SNR range. However, the proposed method consistently achieves higher recognition accuracy, particularly in medium- and high-SNR regions, indicating that the learned feature representations and similarity metrics generalize well to unseen modulation categories and larger-scale datasets.

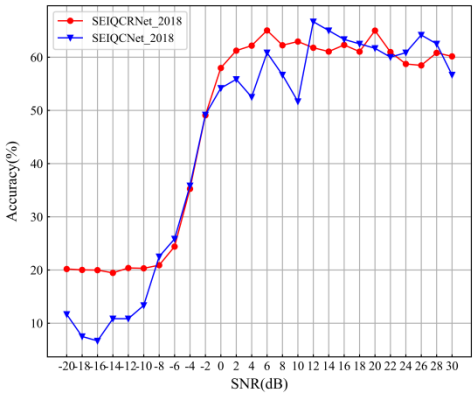


Figure 13. Recognition accuracy comparison between prototypical network and relation network on the RML2018.01a dataset under different few-shot settings

The task-adaptive prototype reweighting mechanism further enhances generalization performance by mitigating task-dependent prototype bias across different datasets.

To further investigate the effectiveness of different metric-learning strategies, a comparative analysis between the relation network and the prototypical network is

conducted under k = 1 and k = 5 few-shot settings. Recognition accuracy curves across different SNR levels are shown in Figure 14, and the corresponding average accuracies under various SNR intervals are summarized in Table 6.

It can be observed from Figure 14 and Table 6 that, under both k = 1 and k = 5 conditions, the networks based on the relation network framework consistently outperform those based on the prototypical network across the entire SNR range. In particular, under the 0–18 dB SNR interval, the relation-network-based model achieves average recognition accuracies of 79.36% and 91.94%, respectively, which are significantly higher than those obtained by the prototypical network under the same conditions. This demonstrates that the adaptive relation-based metric is more effective in capturing the similarity between modulation features in few-shot scenarios.

Figure 14 compares SEIQCRNet with Relation Network (RN) and Prototype Network (PN) under 5-way 1-shot and 5-shot settings. Our model consistently outperforms both baselines across all SNRs, especially in the high-SNR regime (inset), where it reaches ~90% accuracy in the 5-shot case — significantly higher than RN (~82%) and PN (~80%). Even in the 1-shot setting, SEIQCRNet shows stronger generalization, validating the contribution of its dual-attention and adaptive prototype modules.

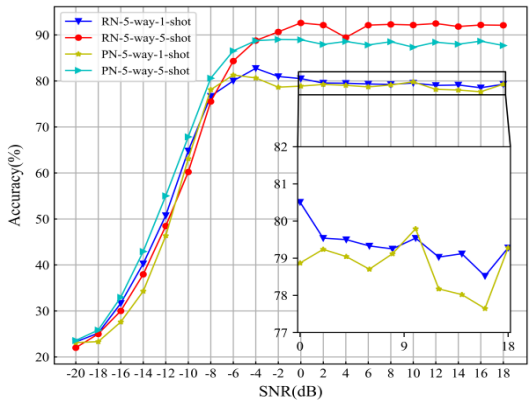


Figure 14. Comparison of recognition accuracy between prototype networks and relational networks

Table 6. Average accuracy in different signal-to-noise ratio ranges

SNR	-20~18dB	-20~-2dB	0~18dB
PN-5-way-1-shot	66.22%	53.64%	78.79%
PN-5-way-5-shot	73.75%	59.30%	88.19%
RN-5-way-1-shot	67.49%	55.63%	79.36%
RN-5-way-5-shot	74.11%	56.29%	91.94%

In addition, to further analyze the influence of task sampling on recognition performance, experiments are conducted under four different small-sample settings by varying the number of samples per class in the support set, namely k = 1, 5, 10, and 20.

The corresponding recognition accuracy curves are shown in Figure 15, and the average recognition accuracies under different SNR intervals are summarized in Table 7.

As illustrated in Figure 15, the recognition accuracy generally increases as the value of k increases across the entire SNR range. When k increases from 1 to 5, the improvement in recognition accuracy is relatively significant, indicating that additional support samples provide more discriminative reference information for the network. But when k >5, there will be no more improvement at that point because the performance gains get smaller with every additional increase as well as being much smaller as well with each addition from here onwards. This trend is also visible on the number in Table 7.

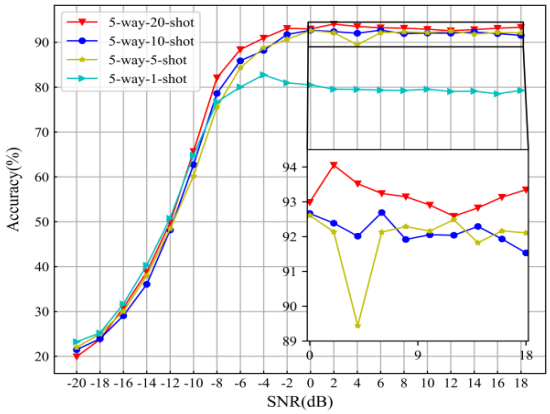


Figure 15. Compared the network recognition accuracy results after extracting tasks with various k value

Table 7. Average accuracy in different signal-to-noise ratio ranges

SNR	-20~18dB	-20~-2dB	0~18dB
5-way-1-shot	67.49%	55.63%	79.36%
5-way-5-shot	74.11%	56.29%	91.94%
5-way-10-shot	74.36%	56.57%	92.15%
5-way-20-shot	75.75%	58.32%	93.17%

Figure 15 Modulation recognition accuracy vs. SNR under 5-way classification with varying support set sizes (1-shot to 20-shot). All curves show improved performance with increasing SNR and larger support sets. The inset highlights the high-SNR region (0–18 dB), where 20-shot and 10-shot achieve near-optimal accuracy (~94%), while 1-shot lags significantly (~81%). Notably, 5-shot exhibits higher variance, suggesting sensitivity to support sample selection. Results confirm that beyond 10 shots, marginal gains diminish, supporting efficient deployment strategies in resource-constrained scenarios.

In overall we see from the experiments of RML2018.01a dataset that the suggested network has very good generalization all over many types of modulation, different levels of SNR, and all kinds of tasks that have been taken for evaluation. These answers speak to worries regarding generalizing from one set of evaluations, and

go on to reinforce the reliability and capability of the technique when dealing with lots of different few-shot modulation recognitions.

The proposed SEIQCRNet is designed with practical deployment in mind, maintaining a lightweight architecture comparable to standard baselines like Prototypical Networks. Although it incorporates dual-attention modules and a prototype reweighting mechanism, these components rely on efficient channel-wise operations and simple vector computations that introduce negligible overhead in terms of parameters and Floating Point Operations (FLOPs). Consequently, the model preserves low inference latency, making it suitable for real-time signal processing on resource-constrained devices without requiring heavy computational resources.

In summary, SEIQCRNet achieves an optimal trade-off between efficiency and performance. The marginal increase in computational cost is vastly outweighed by the significant improvements in recognition accuracy, particularly under challenging low-SNR and few-shot conditions. This favorable efficiency profile confirms that the method is not only theoretically robust but also practically viable for deployment in dynamic electromagnetic environments where both high precision and low latency are critical.

5 Conclusions and Future Work

This paper introduces a kind of deep learning method for recognising modulations which was developed for only few shot communications. SEIQCNet combines with the lightweight IQCNet backbone and channel wise attention in order that it could enhance the distinguishableness of extracted features but still keeps its simple calculations. To overcome problems created from lack of labelled information, add in relation network based metric learning to get SEIQCRNet. Instead of a fixed distance measure we learn task-adaptive similarity measure with SEIQCRNet that can do robust modulation recogniion even with limited amount of data and at different SNR level.

Comprehensive experiments on the RML2016.10a dataset prove that our method is better than other common model methods throughout most SNR (signal-noise ratio) conditions, more significant progress seen when SNRs are smaller or there are fewer training samples.

Moreover, the more difficult cross-dataset evaluation on the larger RML2018.01a dataset verifies the powerful generalization ability of the newly designed structure in regard to the untrained new kind of modulation classes and more data.

Future work would look into applying this to a more realistic communication environment with hardware and channel effects, as well as online/incremental learning. Also, using even better adaptive optimization and cross domain transfers methods are being investigated for making it a little bit more sturdy and also practical for usage within a live network as far as a real world wireless system goes.

References

[1] M. U. Muzaffar, R. Sharqi, A review of spectrum sensing in modern cognitive radio networks, *Telecommunication Systems*, Vol. 85, No. 2, pp. 347–363, February, 2024. https://doi.org/10.1007/s11235-023-01079-1

[2] T. Huynh-The, C. H. Hua, Q. V. Pham, D. S. Kim, MCNet: An efficient CNN architecture for robust automatic modulation classification, *IEEE Communications Letters*, Vol. 24, No. 4, pp. 811–815, April, 2020. https://doi.org/10.1109/LCOMM.2020.2968030

[3] A. P. Hermawan, R. R. Ginanjar, D. S. Kim, J. M. Lee, CNN-based automatic modulation classification for beyond 5G communications, *IEEE Communications Letters*, Vol. 24, No. 5, pp. 1038–1041, May, 2020. https://doi.org/10.1109/LCOMM.2020.2970922

[4] M. Zhang, J. Ma, Z. Zhang, F. Zhou, FedeAMR-CFF: A federated automatic modulation recognition method based on characteristic feature fine-tuning, *Sensors*, Vol. 25, No. 13, Article No. 4000, July, 2025. https://doi.org/10.3390/s25134000

[5] H. Yang, H. Xu, Y. Shi, Y. Zhang, S. Zhao, A few-shot automatic modulation classification method based on temporal singular spectrum graph and meta-learning, *Applied Sciences*, Vol. 13, No. 17, Article. No. 9858, September, 2023. https://doi.org/10.3390/app13179858

[6] J. Li, K. Shi, G. S. Xie, X. Liu, J. Zhang, T. Zhou, Label-efficient few-shot semantic segmentation with unsupervised meta-training, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2024, pp. 3109–3117. https://doi.org/10.1609/aaai.v38i4.28094

[7] M. Patel, X. Wang, S. Mao, Data augmentation with conditional GAN for automatic modulation classification, *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, Linz, Austria, 2020, pp. 31–36. https://doi.org/10.1145/3395352.3402622

[8] Q. Gu, Y. Chang, N. Xiong, L. Chen, Forecasting nickel futures price based on the empirical wavelet transform and gradient boosting decision trees, *Applied Soft Computing*, Vol. 109, Article No. 107472, September, 2021. https://doi.org/10.1016/j.asoc.2021.107472

[9] Y. Wang, M. Liu, J. Yang, G. Gui, Data-driven deep learning for automatic modulation recognition in cognitive radios, *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 4, pp. 4074–4077, April, 2019. https://doi.org/10.1109/TVT.2019.2900460

[10] E. Perenda, G. Bovet, M. Zheleva, S. Pollin, Channel and hardware impairment data augmentation for robust modulation classification, *IEEE Transactions on Cognitive Communications and Networking*, Vol. 10, No. 4, pp. 1263–1279, August, 2024. https://doi.org/10.1109/TCCN.2024.3379379

[11] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, T. M. Hospedales, Learning to compare: Relation network for few-shot learning, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 1199–1208. https://doi.org/10.1109/CVPR.2018.00131

[12] L. Li, J. Huang, Q. Cheng, H. Meng, Z. Han, Automatic modulation recognition: A few-shot learning method based on the capsule network, *IEEE Wireless Communications*

Letters, Vol. 10, No. 3, pp. 474–477, March, 2021. https://doi.org/10.1109/LWC.2020.3034913

[13] Q. Zhou, R. Zhang, J. Mu, H. Zhang, F. Zhang, X. Jing, AMCRN: Few-shot learning for automatic modulation classification, *IEEE Communications Letters*, Vol. 26, No. 3, pp. 542–546, March, 2022. https://doi.org/10.1109/LCOMM.2021.3135688

[14] S. Li, C. Wu, N. Xiong, Hybrid architecture based on CNN and transformer for strip steel surface defect classification, *Electronics*, Vol. 11, No. 8, Article No. 1200, April, 2022. https://doi.org/10.3390/electronics11081200

[15] X. Li, J. Wu, Z. Lin, H. Liu, H. Zha, Recurrent squeeze-and-excitation context aggregation net for single image deraining, *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 2018, pp. 262–277. https://doi.org/10.1007/978-3-030-01234-2_16

[16] Y. Feng, J. Chen, T. Zhang, S. He, E. Xu, Z. Zhou, Semi-supervised meta-learning networks with squeeze-and-excitation attention for few-shot fault diagnosis, *ISA Transactions*, Vol. 120, pp. 383–401, January, 2022. https://doi.org/10.1016/j.isatra.2021.03.013

[17] K. Jiang, J. Zhang, H. Wu, A. Wang, Y. Iwahori, A novel digital modulation recognition algorithm based on deep convolutional neural network, *Applied Sciences*, Vol. 10, No. 3, Article No. 1166, February, 2020. https://doi.org/10.3390/app10031166

[18] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

[19] X. Jin, Y. Xie, X. S. Wei, B. R. Zhao, Z. M. Chen, X. Tan, Delving deep into spatial pooling for squeeze-and-excitation networks, *Pattern Recognition*, Vol. 121, Article No. 108159, January, 2022. https://doi.org/10.1016/j.patcog.2021.108159

[20] N. Xiong, A. V. Vasilakos, L. T. Yang, C. X. Wang, R. Kannan, C. C. Chang, Y. Pan, A novel self-tuning feedback controller for active queue management supporting TCP flows, *Information Sciences*, Vol. 180, No. 11, pp. 2249–2263, June, 2010. https://doi.org/10.1016/j.ins.2009.12.001

[21] C. Lin, N. Xiong, J. H. Park, T. Kim, Dynamic power management in new architecture of wireless sensor networks, *International Journal of Communication Systems*, Vol. 22, No. 6, pp. 671–693, June, 2009. https://doi.org/10.1002/dac.989

[22] C. Xu, J. Shen, X. Du, A method of few-shot network intrusion detection based on meta-learning framework, *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 3540–3552, 2020. https://doi.org/10.1109/TIFS.2020.2991876

[23] K. Stuhlmuller, N. Farber, M. Link, B. Girod, Analysis of video transmission over lossy channels, *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 6, pp. 1012–1032, June, 2000. https://doi.org/10.1109/49.848253

[24] Y. Zeng, C. J. Sreenan, N. Xiong, L. T. Yang, J. H. Park, Connectivity and coverage maintenance in wireless sensor networks, *The Journal of Supercomputing*, Vol. 52, No. 1, pp. 23–46, April, 2010. https://doi.org/10.1007/s11227-009-0268-7

[25] S. Li, Y. Chen, L. Chen, J. Liao, C. Kuang, K. Li, W. Liang, N. N. Xiong, Post-quantum security: Opportunities and challenges, *Sensors*, Vol. 23, No. 21, Article. No. 8744, November, 2023. https://doi.org/10.3390/s23218744

[26] S. Ringel, R. Davidson, Proactive ephemerality: How journalists use automated and manual tweet deletion to minimize risk and its consequences for social media as a public archive, *New Media & Society*, Vol. 24, No. 5, pp. 1216–1233, May, 2022. https://doi.org/10.1177/1461444820972389

[27] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, J. Choo, StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8789–8797. https://doi.org/10.1109/CVPR.2018.00916

[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, *Proceedings of the Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 3146–3154.

[29] M. M. Rahman, S. Al Shakil, M. R. Mustakim, A survey on intrusion detection system in IoT networks, *Cyber Security and Applications*, Vol. 3, Article No. 100082, December, 2025. https://doi.org/10.1016/j.csa.2024.100082

[30] T. Li, W. Liu, Z. Zeng, N. N. Xiong, DRLR: A deep-reinforcement-learning-based recruitment scheme for massive data collections in 6G-based IoT networks, *IEEE Internet of Things Journal*, Vol. 9, No. 16, pp. 14595–14609, August, 2022. https://doi.org/10.1109/JIOT.2021.3067904

[31] L. Xu, Z. Song, D. Wang, J. Su, Z. Fang, C. Ding, W. Gan, Y. Yan, X. Jin, X. Yang, W. Zeng, W. Wu, Actformer: A GAN-based transformer towards general action-conditioned 3D human motion generation, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Paris, France, 2023, pp. 2228–2238. https://doi.org/10.1109/ICCV51070.2023.00212

[32] X. Chai, Y. Wang, X. Chen, Z. Gan, Y. Zhang, TPE-GAN: Thumbnail preserving encryption based on GAN with key, *IEEE Signal Processing Letters*, Vol. 29, pp. 972–976, March, 2022. https://doi.org/10.1109/LSP.2022.3163685

[33] D. M. Anisuzzaman, J. G. Malins, P. A. Friedman, Z. I. Attia, Fine-tuning large language models for specialized use cases, *Mayo Clinic Proceedings: Digital Health*, Vol. 3, No. 1, Article No. 100184, March, 2025. https://doi.org/10.1016/j.mcpdig.2024.11.005

[34] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C. Chan, W. Chen, J. Yi, W. Zhao, X. Wang, Z. Liu, H. Zheng, J. Chen, Y. Liu, J. Tang, J. Li, M. Sun, Parameter-efficient fine-tuning of large-scale pre-trained language models, *Nature Machine Intelligence*, Vol. 5, No. 3, pp. 220–235, March, 2023. https://doi.org/10.1038/s42256-023-00626-4

[35] Y. Song, T. Wang, P. Cai, S. K. Mondal, J. P. Sahoo, A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities, *ACM Computing Surveys*, Vol. 55, No. 13s, pp. 1–40, December, 2023. https://doi.org/10.1145/3582688

[36] H. Gharoun, F. Momenifar, F. Chen, A. H. Gandomi, Meta-learning approaches for few-shot learning: A survey of recent advances, *ACM Computing Surveys*, Vol. 56, No. 12, pp. 1–41, December, 2024.

https://doi.org/10.1145/3659943

[37] J. B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millicah, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, K. Simonyan, Flamingo: A visual language model for few-shot learning, *Proceedings of the Advances in Neural Information Processing Systems*, New Orleans, LA, USA, 2022, pp. 23716–23736.

[38] W. Li, Z. Wang, X. Yang, C. Dong, P. Tian, T. Qin, and J. Luo, Libfewshot: A comprehensive library for few-shot learning, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 45, No. 12, pp. 14938–14955, December, 2023.
https://doi.org/10.1109/TPAMI.2023.3312125

Biographies

Hualin Mu is currently pursuing his Bachelor's degree in Electromagnetic Field and Wireless Technology at Huazhong University of Science and Technology, Wuhan, China. His current research interests include deep learning, signal processing, and computer vision.

Jie Zhang received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in control theory and control engineering from Nanjing University of Science and Technology, Nanjing, China, in 2002, 2004, and 2011, respectively. From April 2013 to March 2014, he was an Academic Visitor with the Department of Information Systems and Computing, Brunel University, London, U.K. He is currently an Research Fellow with the School of Automation, Nanjing University of Science and Technology.

Jerome Yen (Member, IEEE) received the Ph.D. degree in systems engineering (management information systems) from the University of Arizona, in 1992. He joined as a Faculty Member with the University of Macau and a Distinguished Professor with the Department of Computer and Information Science. He also works extensively in financial technologies (FinTech), including blockchain and artificial intelligence, as well as investment and portfolio management involving equities, fixed income, and structured products.

Neal N. Xiong (S'05–M'08–SM'12) received his both PhD degrees in Wuhan University (2007, about Software system engineering), and Japan Advanced Institute of Science and Technology (2008, about dependable communication networks), respectively. Before he attended Southern New Hampshire University, he worked in Georgia State University, Northeastern State University, and Colorado Technical University (full professor about 5 years) about 17 years. His research interests include Cloud Computing, Security and Dependability Service, Parallel and Distributed Computing, Networks Service, and Optimization Theory. Dr. Xiong published over 300 IEEE journal papers and over 100 international conference papers. Some of his works were published in IEEE JSAC, IEEE or ACM transactions, ACM Sigcomm workshop, IEEE INFOCOM, ICDCS, and IPDPS. He has been a General Chair, Program Chair, Publicity Chair, Program Committee member and Organizing Committee member of over 100 international conferences, and as a reviewer of about 100 international journals, including IEEE JSAC, IEEE SMC (Park: A/B/C), IEEE Transactions on Communications, IEEE Transactions on Mobile Computing, IEEE Trans. on Parallel and Distributed Systems. He is serving as an Editor-in-Chief, Associate editor or Editor member for over 10 international journals (including Associate Editor for IEEE Tran. on Industrial Applications; IEEE Tran. on Systems, Man & Cybernetics: Systems; IEEE Tran. on Network Science and Engineering; and Information Science. Editor-in-Chief for Journal of Parallel & Cloud Computing (PCC)), and a guest editor for over 10 international journals, including Sensor Journal, WINET and MONET. He has received the Best Paper Award in the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08) and the Best student Paper Award in the 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009). Dr. Xiong is the Chair of "Trusted Cloud Computing" Task Force, IEEE Computational Intelligence Society (CIS), "http://www.cs.gsu.edu/~cscnxx/index-TF.html", and the Industry System Applications Technical Committee, "http://ieee-cis.org/technical/isatc/"; He is a Senior member of IEEE Computer Society from 2012, E-mail: xiongnaixue@gmail.com; dnxiong@ieee.org.

Sergey M. Avdoshin is a Distinguished Professor at HSE University and founder of Russia's first Software Engineering program. His research interests include software engineering, AI, cybersecurity, machine learning, blockchain, and system trustworthiness. He is an IEEE Senior Member and recipient of a Russian Government Commendation and HSE awards.