IoT

MDPI

Article

Enhancing Automatic Modulation Recognition for IoT Applications Using Transformers

Narges Rashvand [1,*], Kenneth Witham [2], Gabriel Maldonado [1], Vinit Katariya [1], Nishanth Marer Prabhu [3], Gunar Schirner [3] and Hamed Tabkhi [1]

[1] Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, Charlotte, NC 28223, USA; gmaldon2@uncc.edu (G.M.); vkatariy@uncc.edu (V.K.); htabkhiv@uncc.edu (H.T.)
[2] Kostas Research Institute, Northeastern University, Boston, MA 01803 , USA; k.witham@kri.neu.edu
[3] Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA; marerprabhu.n@northeastern.edu (N.M.P.); schirner@ece.neu.edu (G.S.)
* Correspondence: nrashvan@uncc.edu

Abstract: Automatic modulation recognition (AMR) is vital for accurately identifying modulation types within incoming signals, a critical task for optimizing operations within edge devices in IoT ecosystems. This paper presents an innovative approach that leverages Transformer networks, initially designed for natural language processing, to address the challenges of efficient AMR. Our Transformer network architecture is designed with the mindset of real-time edge computing on IoT devices. Four tokenization techniques are proposed and explored for creating proper embeddings of RF signals, specifically focusing on overcoming the limitations related to the model size often encountered in IoT scenarios. Extensive experiments reveal that our proposed method outperformed advanced deep learning techniques, achieving the highest recognition accuracy. Notably, our model achieved an accuracy of 65.75 on the RML2016 and 65.80 on the CSPB.ML.2018+ dataset.

Keywords: automatic modulation recognition; deep learning; attention mechanism; Transformer network; IoT

check for updates

Citation: Rashvand, N.; Witham, K.; Maldonado, G.; Katariya, V.; Marer Prabhum, N.; Schirner, G.; Tabkhi, H. Enhancing Automatic Modulation Recognition for IoT Applications Using Transformers. *IoT* **2024**, *5*, 212–226. https://doi.org/10.3390/iot5020011

Academic Editor: Amiya Nayak

Received: 29 February 2024
Revised: 3 April 2024
Accepted: 6 April 2024
Published: 9 April 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In wireless communication systems, Internet of Things (IoT) devices acting as transmitters utilize various modulation techniques to optimize data transmission rates and bandwidth utilization. While the transmitters determine the modulation type, IoT devices acting as receivers often lack this information. Automatic modulation recognition (AMR) plays a crucial role in non-cooperative communication systems and IoT applications, serving as a fundamental component for demodulating unknown signals. Its significance extends to various applications including spectrum sensing, signal surveillance, interference localization, and cognitive radio [1,2].

Approaches to AMR are broadly classified into two main categories: the likelihood-based (LB) approach and the feature-based (FB) methods. While the LB method is capable of achieving remarkable results, it requires prior knowledge of the probability density function (PDF), adding considerable complexity to the process. In contrast, FB approaches concentrate on directly extracting features from the signal, eliminating the necessity for additional channel or signal information and, thereby, reducing computational demands. These methods depend significantly on extracting and analyzing signal features [2].

Deep learning (DL) has recently made notable advancements, demonstrating remarkable potential across a range of fields, from smart cities [3,4] and computer vision [5,6] to signal processing [7,8]. Furthermore, its application in wireless communications, especially in DL-based AMR techniques, has been expanding. Recent advancements in neural network architectures for the AMR tasks have broadened from convolutional-based neural networks to include innovative designs like Residual Networks (ResNets) [7] and Densely

Connected Convolutional Networks (DenseNets) [8]. LSTM-based recurrent neural networks (RNNs) [9] and convolutional LSTM deep neural networks (CLDNNs) [10] have also been introduced in this domain. CLDNNs and ResNets are particularly effective at lower signal-to-noise ratios (SNRs), whereas LSTMs and ResNets are better at higher SNRs. However, none of these architectures demonstrate superior performance across all SNR values [11]. Moreover, due to their complex nature, these models may not be the most suitable choices for the constrained resources of IoT devices [2,12].

In recent years, the self-attention mechanism, introduced in [5], has emerged as a significantly improved method for modeling sequences. This innovation, introduced through the Transformer model, addresses the parallelization challenges inherent in recurrent neural networks (RNNs) and has achieved broad adoption across the field of computer vision and natural language processing (NLP). The self-attention mechanism is the core of the Transformer architecture, evaluating the correlations between pairs of tokens (symbols or units within sequence data) across the sequence. Transformers have shown promising results in various sequential-pattern-recognition tasks, and especially, the introduction of the vision Transformer (ViT) model [13] marked a novel application of the Transformer's encoder for image-classification tasks.

Despite the rapid advancement of Transformers in the computer vision field, their potential to recognize signal modulation patterns is still largely unexplored. Only a few studies in wireless communications and AMR have explored this area [11,14,15]. AMR involves processing sequential data, time-series signals, and Transformer networks have the high potential to handle time-series data via their self-attention mechanism. This mechanism enables them to capture dependencies across different parts of the input sequence, presenting a significant opportunity for exploration in the field of AMR. Additionally, Transformers can process input sequences in parallel, unlike RNNs, leading to faster training and inference, a crucial advantage for real-time applications in IoT environments. Furthermore, their scalability and ability to handle variable-length sequences make them well suited for processing signals of varying duration.

Inspired by vision Transformers, we introduce Transformer-based architectures for modulation-recognition tasks. The proposed Transformer-based AMR approaches enable rapid processing and minimal resource usage, a critical aspect of edge computing systems within IoT ecosystems. By integrating our proposed models into edge devices, our research not only promises improvements in edge computing, but also addresses key challenges such as bandwidth efficiency and energy consumption. This ensures that IoT devices can operate longer and more effectively within their constrained environments.

The rest of this paper is organized as follows. Section 2 addresses the two datasets, RadioML2016.10b and our modified CSPB.ML.2018 dataset with channels (CSPB.ML.2018+), and emphasizes their properties. Section 3 focuses on the Transformer-based architectures for the task of AMR, followed by an introduction to four distinct tokenization strategies. Experiments and discussions are detailed in Section 4, with future investigation in Section 5 and conclusions summarized in Section 6.

2. Datasets

- RadioML2016.10b [16]:
  This dataset is composed of ten modulations, including eight digital and two analog modulation types over SNR values ranging from -20 dB to +18 dB, in increments of 2 dB, i.e., $\{-20, -18, \ldots, +16, +18\}$. These samples are uniformly distributed across this SNR range. The dataset, which includes a total of 1.2 million samples with a frame size of 128 complex samples, is labeled with both SNR values and modulation types. The dataset is split equally among all considered modulation types. At each SNR value, the dataset contains 60,000 samples, divided equally among the ten modulation types, with 6000 samples for each type. For the channel model, simple multi-path fading with less than five paths was randomly simulated in this dataset. It also includes random channel effects and hardware-specific noises through a variety of models,

including sample rate offset, the noise model, center frequency offset, and the fading model. Thermal noise was used to set the desired SNR of each data frame.

- CSPB.ML.2018+ [17]:
  This dataset is derived from the CSPB.ML.2018 [17] dataset, which aims to solve the known problems and errata [18] with the RadioML2016.10b [16] dataset. CSPB.ML.2018 only provides basic thermal noise as the transmission channel effects. We extended CSPB.ML.2018 by introducing realistic terrain-derived channel effects based on the 3GPP 5G channel model [19]. CSPB.ML.2018+ contains eight different digital modulation modes, totaling 3,584,000 signal samples. Each modulation type has signals with a length of 1024 *IQ* samples. The channel effects applied include slow and fast multi-path fading, Doppler, and path loss. The transmitter and receiver placements for the 3GPP 5G channel model are randomly selected inside a 6×6 km square. The resulting dataset covers an SNR (E_s/N_0) range of -20 to 40 dB with the majority of SNRs distributed log-normally with $\mu_{SNR} \approx 0.71$ dB and $\sigma_{SNR} \approx 8.81$ dB using $SNR_{dB} = 10\log_{10}(SNR_{linear})$ as the log conversion method.

  The characteristics of these two datasets are detailed in Table 1.

Table 1. Dataset characteristics: RadioML2016.10b [16] vs. CSPB.ML.2018+

	RadioML2016.10b [16]	**CSPB.ML.2018+**
Number of Modulation Types	10 (8 digital and 2 analog modulations)	8 digital modulations
Modulation Pool	BPSK, QPSK, 8PSK, QAM16, QAM64, BFSK, CPFSK, PAM4, WBFM, AM-DSB	BPSK, QPSK, 8PSK, DQPSK, MSK, 16-QAM, 64-QAM, 256-QAM
Signal Length	128	1024
SNR Range	-20 dB to +18 dB	-19 dB to +40 dB
Number of Samples	1,200,000	3,584,000
Sample Distribution across SNR Range	log-uniform distribution	log-normal distribution
Channel Effects	• thermal noise • simple multi-path fading • center-frequency and sample rate offset	• path loss • 3GPP channel model with correlated slow and fast multi-path fading • center-frequency and sample rate offset

3. Transformer Architectures

The Transformer model, presented in [5], marked a significant advancement in the field of sequence-to-sequence processing. Comprising an encoder and a decoder, each with multiple layers, this model efficiently maps an input sequence of symbols (words) to a sequence of continuous representations. The decoder then transforms these representations to produce an output sequence. This architecture has demonstrated remarkable accuracy across a variety of sequence-to-sequence tasks, including machine translation and text summarization.

The Transformer architecture, initially designed for language tasks, has been effectively adapted for image processing in the ViT paper [13]. Unlike the original Transformer model, which includes both an encoder and decoder for sequence-to-sequence tasks, ViT adapts the Transformer's encoder to process images. ViT treats an image as a sequence of patches and applies the Transformer encoder to these patches to perform image classification and process images as sequences of flattened patches. The encoder consists of several identical layers, each comprising two sub-layers: a multi-head self-attention mechanism and a feed-forward neural network. Additionally, it has residual connections and layer normalization to mitigate the vanishing gradient issue that arises with deep models. The feed-forward neural network consists of a fully connected layer architecture. In the self-attention mechanism, each element of the input sequence interacts with all other elements to calculate the attention weights, highlighting the importance of the relationships among

different positions within the sequence. The multi-head attention mechanism splits the attention calculation across several heads, with each head independently performing the attention computation. This allows each head to concentrate on distinct features of the input sequence, providing a better understanding of the sequence. This shows how the Transformer can be used not just for text, but also for images, demonstrating its versatility.

In this section, we introduce different methods for recognizing signal modulation types using a Transformer-based architecture. Figure 1 illustrates the overall architecture of our Transformer-based approaches, comprising three key components: The tokenization module, the Transformer-encoder module, and the classifier module. The raw *IQ* data, composed of in-phase (I) and quadrature (Q) components, forms a two-channel input. However, Transformer networks require their inputs to be in the form of tokens, which is achieved through the tokenization module. Various strategies for tokenization are employed in each Transformer model discussed in subsequent sections. In all these architectures, the input undergoes tokenization to generate tokens before being fed into the Transformer-encoder module for capturing relevant features from the data, and the classification task concludes with the output from the classifier module, which is a fully connected neural network.

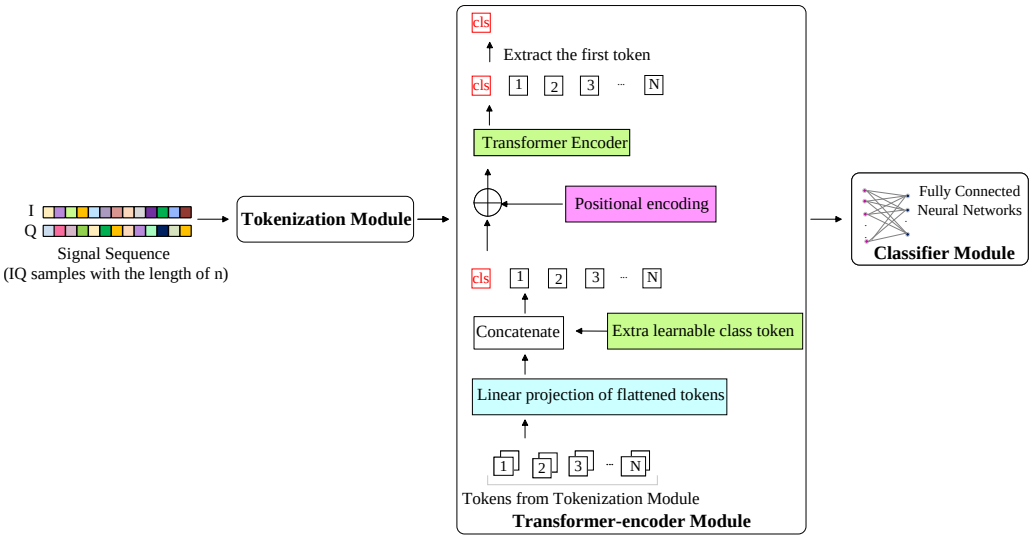

Figure 1. The overall architecture of our proposed Transformer-based model for the AMR task includes three main components: a tokenization module that converts the signal into tokens, a Transformer-encoder module, which captures information and extracts relevant features through self-attention mechanism, and a classifier module for the final classification step.

Utilizing various techniques to generate tokens, we investigated four distinct Transformer architectures for the AMR task, namely TransDirect, TransDirect-Overlapping, TransIQ, and TransIQ-Complex. These methods are explained further in the following subsections.

3.1. TransDirect

The detailed architecture considered for TransDirect is shown in Figure 2. The process starts with the tokenization of data, a crucial step in implementing the Transformer network. This approach is similar to ViT, where image patches are generated from segments of the input as tokens.

In the context of TransDirect, tokenization involves dividing the *IQ* samples, which have a sequence length of n, and consists of two channels into multiple shorter subsequences that are called tokens. The I and Q samples from the signal sequence can be expressed as $I = [i_1, i_2, \ldots, i_n]$ and $Q = [q_1, q_2, \ldots, q_n]$, with n denoting the length of the signal. In this arrangement, both the I and Q sequences are divided into shorter segments of length l, using a sliding window technique with a step size of w. As the tokens (segments) are designed to be non-overlapping, the stride length w is set equal to the segment length

(token size) l, ensuring distinct and consecutive tokens. Consequently, the segmented subsequences for the Is are given by $I_1 = [i_1, i_2, \ldots, i_l]$, with subsequent sequences like $I_2 = [i_{1+w}, i_{2+w}, \ldots, i_{l+w}]$, leading up to $I_N = [i_{1+(N-1)w}, i_{2+(N-1)w}, \ldots, i_{l+(N-1)w}]$. A similar segmentation applies to the Q sequence. As a result, the overall count of tokens using this method is determined by $N = \frac{n}{l}$.

Until this point, the model converts the two-dimensional vector at each time step into N tokens with two channels. These tokens are then processed through the Transformer-encoder module, according to Figure 1, where they first get transformed into linear sequences by the Linear Projection Layer, acting as the starting point of the Transformer-encoder's processing. Moreover, an additional trainable "class token" is also included with the tokens, increasing the total count of tokens to $N + 1$. Positional encoding is then applied to the tokens before they are input into the Transformer-encoder, enabling the model to understand the relative positioning of the tokens. The tokens then pass through N_l Transformer encoder layers, which utilize the self-attention, with a hidden size (embedding size) of m and attention heads of N_{head}. As the core part of the Transformer, the attention mechanism maps a query and a set of key–value pairs to the output, establishing connections between various positions within a sequence to integrate information across the entire input data. After passing through N_l Transformer encoder layers, the output is an $(N + 1) \times d$ matrix, where d is the embedded dimension, calculated as twice the number of samples per token l, reflecting the dual-channel structure of the input data. The first token from this output is then extracted and fed into the classifier module. This module consists of fully connected layers with a set number of hidden layers of N_{hidden}. The final result of the classification comes from the output of the classifier module, generating a c-dimensional vector, where each represents the probability of a particular modulation type.

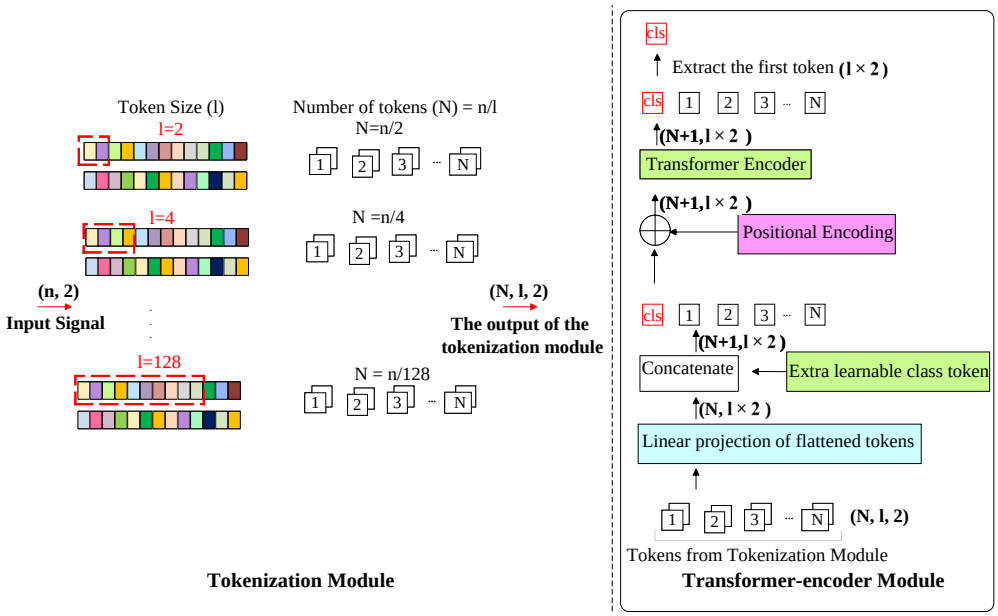

Figure 2. TransDirect architecture. In the tokenization module of this architecture, *IQ* samples are segmented into shorter sequences referred to as tokens, each having a size of l.

3.2. TransDirect-Overlapping

The overall architecture of TransDirect-Overlapping is illustrated in Figure 3. Similar to TransDirect, the original I and Q sequences are segmented into shorter tokens of length l through the use of a sliding window method in the tokenization module. However, unlike the previous method, this technique introduces an overlap between consecutive tokens. Specifically, the step size w is configured to be $l/2$, resulting in a 50% overlap between adjacent tokens. This overlapping strategy allows for a more comprehensive analysis through the reuse of data across tokens. The sub-sequences for the I component are, thus, initiated with $I_1 = [i_1, i_2, \ldots, i_l]$,

with the following sub-sequences such as $I_2 = [i_{1+(l/2)}, i_{2+(l/2)}, \ldots, i_{l+(l/2)}]$ and continuing to $I_N = [i_{1+(N-1)(l/2)}, i_{2+(N-1)(l/2)}, \ldots, i_{l+(N-1)(l/2)}]$, and this segmentation approach is similarly applied to the Q component. Accordingly, the total number of tokens generated by this method is calculated as $N = \frac{n-l}{w} + 2$. Despite the different tokenization techniques, TransDirect-Overlapping retains the same Transformer-encoder module and classifier as TransDirect, offering an improved method for initial data segmentation via a sliding window technique that ensures data overlap, thereby maximizing data utilization. Like the TransDirect tokenization approach, the output dimensionality of the tokenization module remains $(N, l, 2)$. However, TransDirect-Overlapping sets itself apart by increasing the number of tokens to $N = \frac{n-l}{w} + 2$, while maintaining the embedding dimension similar to TransDirect. This dimension equals twice the number of samples per token l, a feature unchanged from TransDirect due to the consistent number of channels (2), presenting I and Q components.

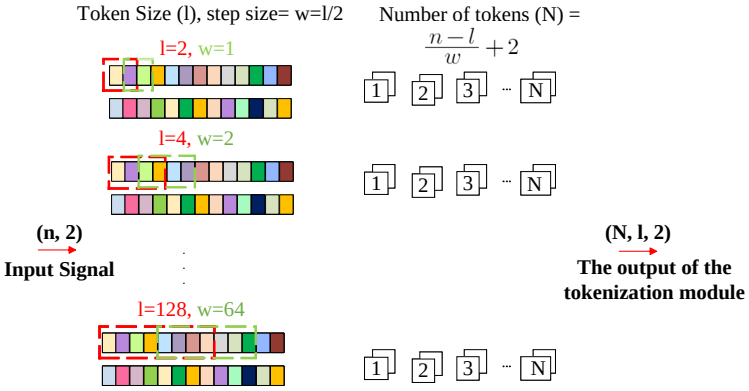

Figure 3. Tokenization module of TransDirect-Overlapping architecture, which divides IQ samples into tokens, each with a length l, where each token overlaps the preceding one by $l/2$.

3.3. TransIQ

In the TransIQ design, as illustrated in the Figure 4, we enhanced the model's feature-extraction capacity by expanding its receptive field in the tokenization phase. Following the tokenization approach of earlier techniques, we first slice the IQ sequence into shorter segments using a sliding window technique in the tokenization module. These segments are then reorganized into a matrix of signal sequences, and one-dimensional convolutional encoding is performed.

The introduction of a CNN layer between the tokenization module and the Transformer-encoder plays a crucial role in enhancing the feature representation of the input data and facilitates the extraction of detailed features from each token. The CNN layer, through its convolutional filters (kernels), is inherently designed to extract spatial features from the data. Each filter specializes in identifying specific patterns within the input. By applying these filters across the signal sequences, the CNN layer can highlight important features from the input data. This process effectively prepares a rich, detailed feature map for each token. On the other hand, the Transformer-encoder utilizes a self-attention mechanism to weigh the importance of different features within the input. To elaborate more, after the CNN layer enriches the feature representation, the Transformer encoder assesses these features, determining their relevance to the task of modulation classification. The self-attention mechanism allows the model to focus on the most relevant features by assigning higher weights to them. This capability stems from the Transformer's design to dynamically allocate attention based on the input's context, enabling it to detect which features (enhanced by the CNN layer) are crucial for understanding the input sequence's characteristics. So, the CNN layer ensures that the Transformer encoder receives a high-resolution feature map, where important characteristics are already highlighted. Subsequently, the Transformer can efficiently evaluate these features' relevance, adjusting its internal weights to prioritize tokens crucial for accurate modulation classification. So, leveraging CNNs for improved

feature representation and Transformers to dynamically adjust the importance of those features, TransIQ efficiently allocates suitable weights to each feature segment, focusing on key positions in the input temporal data.

So, the tokens generated from the sliding window are fed into a convolutional layer with a kernel size of k with the same padding and N_c number of output channels. The output of the convolution layer is fed to a ReLU activation function. Up to this stage, the model transforms the two-dimensional vector at each time sample into an N_c (output channel) -dimensional feature representation using different kernels in the convolutional layer. This matrix is then processed through N_l Transformer encoder layers, which operates on an $(N+1) \times d$ matrix using the self-attention mechanism. Here, $N+1$ represents the total number of tokens, and d represents the embedding dimension of the Transformer. In this scenario, d equals the product of the token size and the number of output channels of the CNN layer. The first token from the output of the Transformer encoder is then fed into the classifier module to perform the modulation classification.

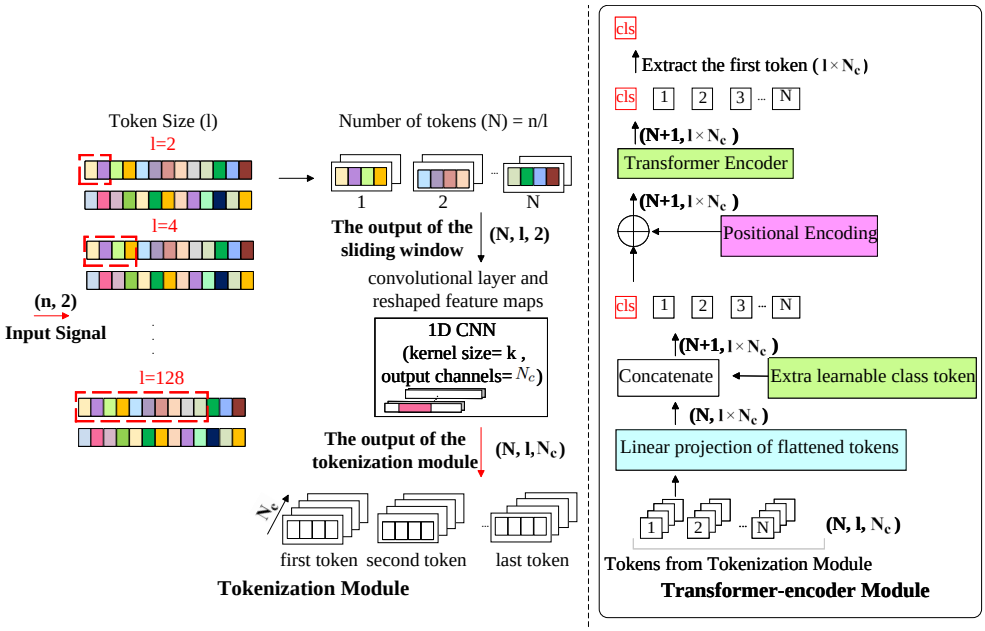

Figure 4. TransIQ architecture. In the tokenization module of this architecture, the input signal is segmented into tokens. Each token then undergoes one-dimensional convolutional before being processed by the Transformer-encoder module.

3.4. TransIQ-Complex

We have developed a novel strategy for working with complex (I and Q) information in our tokenization module, inspired by the DeepFIR [20] model. This model has shown its effectiveness in AMR tasks by using a complex CNN layer with complex weights and biases. Considering the composite nature of the input data, featuring both real and imaginary components, the incorporation of a complex convolutional neural network (CNN) layer within our tokenization module emerges as a key strategy for enhanced data handling. This enhancement aligns with the TransIQ architecture, as illustrated in Figure 4, while integrating two essential modifications. Primarily, the model processes the input signal as a singular channel entity, diverging from the conventional dual-channel data approach. Specifically, although the input signal originally consists of two channels, one for the I and another for the Q components, utilizing a complex convolutional layer allows for the integration of these channels into a unified one-channel input, thereby resizing the signal's dimension to $n \times 1$. Following initial segmentation into shorter tokens of length l, the dimensionality of each token after applying sliding window is maintained at $l \times 1$. A second modification is the inclusion of a complex convolutional layer in place of a standard CNN in the architecture shown in Figure 4. Subsequently, each token undergoes processing via a

complex convolutional layer that individually extracts features from the singular-channel data, generating an output with N_c output channels. Thus, this tokenization method transforms the one-dimensional vector of the signal sequence into a multidimensional feature representation.

4. Experimental Results and Discussion

4.1. Ablation Study

For evaluating the performance of our proposed models, we utilized the RadioMl2016.10b dataset and our CSPB.ML.2018+ dataset. These datasets were randomly divided, allocating 60% for training, 20% for validation, and 20% for testing. Our experimental setup was based on the PyTorch framework, and we employed the Cross-Entropy (CE) loss function. Furthermore, all experiments were performed on a GPU server equipped with 4 Tesla V100 GPUs with 32 GB of memory. The Adam optimizer was used for all experiments. Each model underwent training for 100 epochs, starting with a learning rate of 0.001. For experiments involving token sizes larger than 16 samples, a reduced learning rate of 0.0001 was employed with a batch size of 256. Classification accuracy was also evaluated using the F1 score, a key performance metric for classification problems that combines precision and recall into a single measure by calculating their harmonic mean.

In all our experiments, we employed the same structure unless changes from this configuration are specifically mentioned. Our implementation comprises a Transformer encoder architecture with four encoder layers, two attention heads, and a feed-forward network (FFN) with a dimensionality of 64. In the classifier module, we utilize a fully connected neural network consisting of a single hidden layer with 32 neurons with the ReLU activation function and dropout, followed by an output layer adjusted to the number of modulation types in each dataset.

Our investigation began with the TransDirect architecture, evaluating the impact of token size on model performance, focusing on how the token length affects classification accuracy. For the experiments, we used token lengths of 8, 16, 32, and 64, which corresponds to creating 128, 64, 32, and 16 tokens, respectively, from the original signal sequence length of 1024 in the CSPB.ML.2018+ dataset. According to our results detailed in Table 2, we noticed that accuracy increases by about 3% when the token size doubles from 8 to 16, reaching 56.29% at its peak and then dropping by about 4% when the sample size per token is 64. The embedded dimension, in this structure, equals the number of samples per token times the number of channels, which is 2 for the I and Q channels. For example, with the TransDirect model using a token size of 16, each token consists of 16 samples across two channels. Subsequently, in the Linear Projection Layer, these tokens are converted to the linear sequence of 32. This leads to an embedding dimension of 32, calculated by multiplying the 16 samples by the two channels. Therefore, as the number of samples per token increases from 8 to 64, while maintaining fixed numbers of encoder layers and head attentions, the total number of parameters grows from 17.2 K to 420 K. This demonstrates that the TransDirect reaches optimal performance when using a token size of 16 samples, understanding the significance of token size and the model's complexity in enhancing classification accuracy while taking into account the computational constraints of IoT devices. This experiment shows a trade-off between the optimal token size and model complexity when tokens are input directly to the Transformer encoder.

We conducted additional tests using the TransDirect-Overlapping architecture, where each token overlaps the previous one by 50%. This variation in tokenization led to a doubling of the number of tokens compared to the initial setup, TransDirect. Specifically, when changing the number of samples per token from 8 to 64, the total number of tokens ranged from 256 (for 8 samples per token) to 32 (for 64 samples per token). Despite this, the total number of model parameters stayed the same as in the TransDirect technique, because the embedding dimension does not change. Table 2 further illustrates a consistent trend in model accuracy, showing an increase of approximately 6%, from 53.98% with a token

size of 8 to 60.08% when the token size is increased to 32, and then starting to decrease by approximately 3% as the token size increases to 64.

We expanded our investigation into the TransIQ model through a series of comprehensive experiments. We adjusted the network's depth, kernel configurations, and the number of output channels to achieve an optimal balance between accuracy and model size. Starting with the previously implemented Transformer-encoder configuration of two heads and four layers, we explored token sizes ranging from 8 to 32, integrating a convolutional layer with eight output channels. The highest accuracy we achieved was 63.72%, with the optimal token size set to 16 samples. This represents an approximately 7% improvement in accuracy compared to the best performance of the TransDirect and an approximately 3% increase over the TransDirect-Overlapping model. The improved performance is due to the inclusion of a convolutional layer, which enhances the model's capability to extract features from each token, distinguishing it from the TransIQ and TransIQ-Overlapping models.

In another set of experiments, we opted for a token size of eight to constrain the model's size. We then explored different configurations for the head and layer within our TransIQ, seeking to enhance efficiency by minimizing the parameter count. As the total number of parameters is proportional to the CNN's complexity and the embedding dimension of the Transformer encoder, we evaluated two versions: the first variant, named the large variant of TransIQ, features a convolutional layer with eight output channels, four heads, and eight layers, achieving the highest accuracy on the dataset at 65.80%. The small variant of TransIQ, on the other hand, utilizes a token size of eight and a Transformer-encoder architecture with two heads and six layers. This model has a total of 179 K parameters and achieves an accuracy of 65.39%.

We further explored the impact of integrating a complex convolutional layer by employing the complex layer in the TransIQ-Complex model. This model, distinct from typical CNNs, incorporates complex weights and biases in its CNN layer. As the token size expanded from 8 to 16, we observed an accuracy improvement of approximately 1.6%. However, considering the increase in model complexity and the marginal gains in accuracy, the use of a complex layer proved to be minimally beneficial in scenarios where the model size is a critical consideration.

Table 2. A comparative analysis of different methods of tokenization in terms of F1 score and number of parameters on CSPB.ML.2018+ dataset. The best result for each tokenization strategy is in bold.

Methods	Tokenization	F1 Score	Number of Parameters
TransDirect	8 samples	53.15	17.2 K
	16 samples	**56.29**	44.1 K
	32 samples	56.20	128 K
	64 samples	52.24	420 K
TransDirect-Overlapping	8 samples	53.98	17.2 K
	16 samples	59.43	44.1 K
	32 samples	**60.08**	128 K
	64 samples	57.67	420 K
TransIQ	8 samples with 8 output channels (head = 2, layer = 4)	63.69	128 K
	8 samples with 16 output channels (head = 2, layer = 4)	63.25	420 K
	16 samples with 8 output channels (head = 2, layer = 4)	63.72	420 K
	32 samples with 8 output channels (head = 2, layer = 4)	62.68	1.5 M
	8 samples with 8 output channels (head = 4, layer = 8)	**65.80**	229 K
	8 samples with 8 output channels (head = 2, layer = 6)	65.39	179 K
TransIQ-Complex	8 samples with 8 output channels (head = 2, layer = 4)	61.19	420 K
	16 samples with 8 output channels (head = 2, layer = 4)	**62.79**	1.5 M
	32 samples with 8 output channels (head = 2, layer = 4)	59.33	5.6 M

Referencing Table 2, an analysis comparing the F1 score and parameter count across different token sizes (8 to 64) for four methods of tokenization reveals detailed insights. A

minimal increase in accuracy, a 0.83% improvement, is observed when comparing TransDirect to TransDirect-Overlapping for a token size of eight, but there is a more significant improvement, nearly 10%, from TransDirect-Overlapping to the TransIQ model. For the token size of 16, both the TransDirect and TransDirect-Overlapping architectures have the same number of parameters of 44.1 K, but TransDirect-Overlapping achieves an approximately 3% higher accuracy due to its overlapping tokenization technique. Furthermore, adopting the TransIQ model with the same token size, and with a convolutional layer with eight output channels, increases the parameter count to 420 K. This change results in a 4% accuracy improvement over TransDirect-Overlapping and an impressive 7% improvement compared to TransDirect with the same token size. Conversely, the TransIQ-Complex architecture, despite increasing its parameters to 1.5 M with the token size of 16, experiences a roughly 1% drop in accuracy compared to the TransIQ with the same token size. This indicates that, while token size plays a crucial role in achieving higher accuracy, simply increasing model complexity does not ensure improved performance. Identifying the optimal balance requires thorough experimentation, as evidenced by the data presented in Table 2.

4.2. Comparison with Other Baseline Methods

To evaluate the performance of our proposed methods, we conducted a quantitative analysis comparing them against models with varying structures on two different datasets, RadioML2016.10b and CSPB.ML.2018+. Our initial analysis focused on the RadioML2016.10b dataset, evaluating models including DenseNet [21], CNN2 [22], VTCNN2 [23], ResNet [7], CLDNN [21], and Mcformer [11]. Among these models, DenseNet, CNN2, VTCNN2, and ResNet are built upon a CNN, and the CLDNN integrates both RNN and CNN architectures. In this comparison, we also included another baseline model, Mcformer, which is built on the Transformer architecture. However, since the Mcformer model was not developed using the PyTorch framework and our inability to access its detailed architecture, we were unable to replicate the results and determine the number of parameters. Therefore, we referenced the results for Mcformer from [11].

In the initial comparison, we tested two variants of our TransIQ model against baselines on the RadioML2016.10b dataset. The TransIQ-large variant featured a token size of eight samples, a convolutional layer with eight output channels, and a Transformer encoder with 4 heads and 8 layers, and the TransIQ-small variant, with the same token size and convolutional layer but a Transformer encoder with 2 heads and 6 layers.

The experimental results are shown in Table 3. As demonstrated in this table, our proposed model outperforms all baseline models in terms of accuracy on this dataset. The table illustrates that our proposed models achieve roughly a 9% better accuracy compared to the DenseNet model, despite having 14-times fewer parameters in the TransIQ-large variant and 18-times fewer in the TransIQ-small variant. Moreover, both TransIQ model variants demonstrate superior accuracy to ResNet, with only a minimal increment in the number of parameters, making their parameter counts relatively comparable. Our analysis, summarized in Table 3, underscores the TransIQ architecture's impressive capability in the AMR task, combining high accuracy with reduced parameter needs. To further validate the robustness and potential of our model, we expanded our evaluation to include another dataset characterized by more channel effects, aiming to test our model's capability under more challenging scenarios.

Consequently, we assessed the performance of our proposed models on the CSPB.ML.2018+ dataset, which has signals of a longer length and more channel effects compared to the RadioML2016.10b dataset. The results, detailed in Table 4, show that, while the parameter counts for baseline models increase significantly with this dataset, the complexity of our proposed models remains consistent. This consistency in complexity is attributed to the same embedding dimension. This demonstrates our model's adaptability and efficiency, maintaining their complexity unchanged even when applied to datasets featuring signals of longer lengths.

Table 3. Comparative analysis of F1 scores and parameter Counts across various models on the RadioML2016 [16] dataset. The highest F1 scores are in bold among the baselines and proposed models.

Methods	F1 Score	Number of Parameters
DenseNet [21]	56.93	3.3 M
CLDNN [21]	61.14	1.3 M
VTCNN2 [23]	61.53	5.5 M
CNN2 [22]	60.94	1 M
ResNet [7]	64.62	107 K
Mcformer [11]	**65.03**	-
TransIQ (large variant)	**65.75**	229 K
TransIQ (small variant)	**65.61**	179 K

Figure 5 shows how accuracy changes for different methods on the CSPB.ML.2018+ dataset when the SNR changes. It is observed that classification accuracy increases with higher SNR levels across all neural network architectures. We note that the larger variant of TransIQ consistently outperforms the baseline models across all SNR values. The small variant of TransIQ also shows competitive performance against the baselines. In particular, the smaller variant has 179 K parameters, significantly less than the millions of parameters found in all baseline models except ResNet. Remarkably, the large variant of TransIQ achieves better accuracy than the ResNet model, while they are comparatively similar in terms of the number of parameters. The superiority of our model becomes particularly apparent especially in situations with a low SNR ($SNR < 0$). Although the TransIQ-large variant outperforms the small one, this performance improvement comes with the increased computational complexity of 50 K. However, both architectures have significantly fewer parameters compared to the other baselines. This highlights the suitability of our proposed models for IoT applications.

Table 4. Comparative analysis of F1 scores and parameter counts across various models on the CSPB.ML.2018+ dataset. The highest F1 scores are in bold among the baselines and proposed models.

Methods	F1 Score	Number of Parameters
DenseNet [21]	57.87	21.6 M
CLDNN [21]	61.26	7.1 M
VTCNN2 [23]	47.29	42.2 M
CNN2 [22]	52.57	3.8 M
ResNet [7]	**65.48**	164 K
TransIQ (large variant)	**65.80**	229 K
TransIQ (small variant)	65.39	179 K

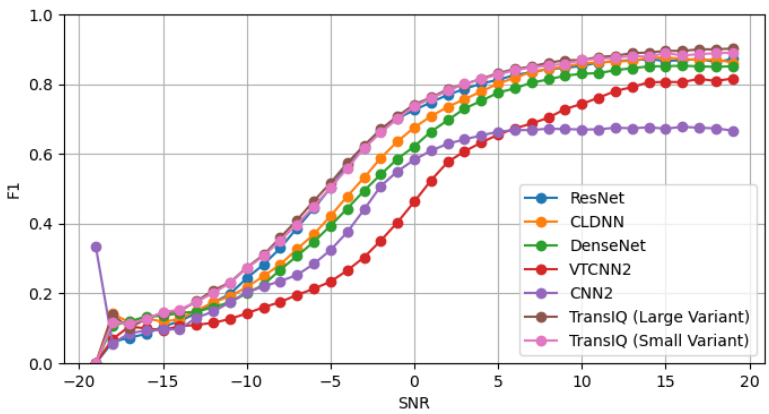

Figure 5. Modulation-recognition accuracy comparison between the two variants of TransIQ and other baseline models on the CSPB.ML.2018+ dataset with a change in the SNR, where the SNR ranged from −19 dB to +20 dB.

The confusion matrix, illustrated in Figure 6, shows the performance of the top-performing baseline model (ResNet), as well as TransIQ variants on the CSPB.ML.2018+ dataset. This matrix is a critical tool for evaluating model accuracy, detailing the relationship between actual and predicted class instances through its structured design. It is an $N * N$ matrix, where N represents the number of classes for classification. Specifically, each row in the matrix corresponds to actual class instances, while each column reflects the model's predictions. The numbers along the diagonal of the matrix represent the true positives for each class, meaning the number of times the model correctly predicted each class. On the other hand, off-diagonal elements represent misclassifications. According to Figure 6, here, the confusion matrix is an $8 * 8$ matrix, aligning with the eight different modulation types in the CSPB.ML.2018+ dataset. For instance, in the confusion matrix of the ResNet model, the value located at the intersection of the first row and the first column is 0.97. This indicates that 97% of the instances truly belonging to BPSK modulation were correctly classified as BPSK by the model. To illustrate, the element in the second row and first column of ResNet's confusion matrix has a value of 0.02, indicating that 2% of the instances that are actually QPSK modulation were incorrectly predicted as BPSK by the model.

Further analysis of it reveals that ResNet still has significant confusion between QAM modulations. Some higher order PSK modulations are also classified as QAM. TransIQ is better able to correctly classify PSK modulations and shows a small amount of confusion among higher-order PSK. TransIQ is also much better able to discern between 16QAM and other higher order QAM modulations. Both variants of TransIQ showed similar confusion between 64- and 256-QAM.

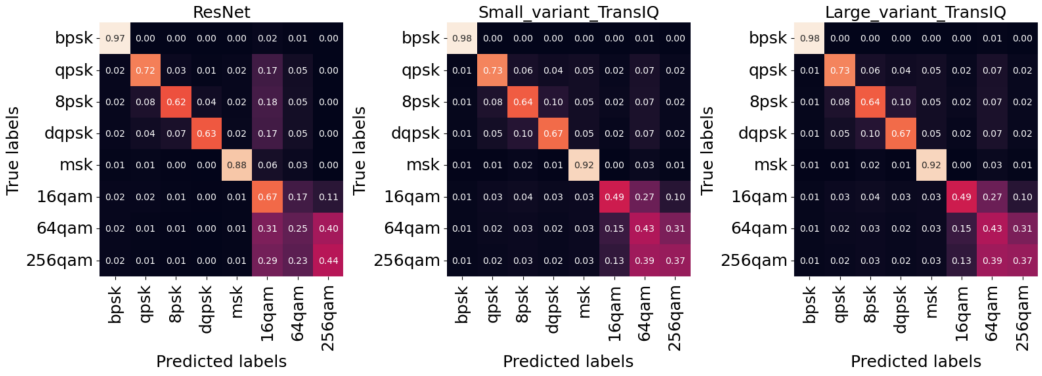

Figure 6. Confusion matrix for ResNet versus proposed models (TransIQ-large variant and TransIQ-small variant) on CSPB.ML.2018+ dataset. Each row represents the actual class, and each column corresponds to the class as predicted by the mentioned models.

4.3. Latency and Throughput Metrics

In assessing the performance of TransIQ (large and small variant), two metrics of latency and throughput were analyzed as well. These analyses offer insights into the models' effectiveness and suitability for real-world scenarios. Latency is defined as the duration required for the model to process a single input, measured in seconds. On the other hand, throughput refers to the model's capability to process a certain number of inputs per second.

Our experiments to measure latency and throughput were carried out using the NVIDIA Jetson AGX Orin Developer Kit, created by NVIDIA Corporation, headquartered in Santa Clara, California, United states. This kit is equipped with an Ampere architecture GPU with 2048 CUDA cores and is also powered by an ARM-based CPU with 12 cores. The system includes the memory of 64 GB, providing sufficient resources for demanding AI tasks. To carry out our measurements, we utilized the CPU resources provided by the kit, in the MAXN performance mode and on NVIDIA JetPack version 6.0-b52. The measured latency and throughput for the two variants of TransIQ are presented in Table 5. The table illustrates that the latency for TransIQ (small variant) is 3.36 ms/sample, while the large variant has a higher latency of 5.93 ms/sample. This increased latency for the large variant

was anticipated due to its greater complexity, with 229 K parameters compared to the 179 K parameters of the small variant, requiring more time to make predictions. In terms of throughput, the small variant demonstrates a throughput of approximately 297 samples per second. This high rate indicates its capability to manage large volumes of predictions with minimal latency, making it well-suited for real-time applications with a bit lower accuracy. Conversely, the large variant has a throughput of approximately 168 samples per second. Although this is lower compared to the small variant, the large variant is suitable for AMR applications requiring higher prediction accuracy, at the cost of slower processing input data.

Table 5. Measured latency and throughput of two variants of TransIQ on NVIDIA Jetson AGX Orin Developer Kit.

Models	Latency (ms/Sample)	Throughput (Sample/s)
TransIQ (small variant)	3.36	297.61
TransIQ (large variant)	5.93	168.63

5. Future Investigation

While this paper provides encouraging results for a Transformer-based architecture (TransIQ), there are many possible future research directions. We introduced a new approach to process the signal data that come in *IQ* samples. This opens the door to further research on how to represent signals differently. In the future, we plan to combine various methods of signal representation, including amplitude/phase and time–frequency. This combination could give us more detailed information and help improve how well we can recognize and understand these signals.

A key area for future research will also be to explore strategies for maintaining high recognition accuracy even with limited data samples. Moving forward, we aim to delve into the development of a Transformer-encoder designed to achieve high accuracy in scenarios characterized by small sample sizes. This direction not only addresses the challenge of data scarcity, but also enhances the method's applicability across various contexts where collecting large datasets is impractical. Additionally, the model's robustness may diminish due to the absence of crucial labels. Therefore, exploring semi-supervised or unsupervised learning methods presents a promising avenue for future research. These approaches could potentially enhance the model's ability to learn from limited or unlabeled data, thereby improving its overall effectiveness and reliability.

In future work, we plan to explore the application of Transformer-based architectures within distributed environments, especially in scenarios where IoT devices are widely dispersed. Investigating the development and implementation of Transformers in such environments represents a significant avenue for advancing the application of Transformer techniques in IoT networks. An important aspect of this research will focus on handling multi-receiver datasets, where each receiver is subject to its unique channel effects. This direction aims to address the challenges of integrating sophisticated deep learning models like Transformers with the complex dynamics of distributed IoT systems, enhancing their ability to efficiently process and analyze data across various network conditions.

6. Conclusions

In wireless communication, AMR plays a crucial role in efficient signal demodulation. Traditional AMR methods face challenges with complexity and high computational demands. Recent advancements in deep learning have offered promising solutions to these challenges. This study highlights the adaptability of the Transformer architecture, originally developed for language processing, to the field of signal modulation recognition. By adapting the Transformer's encoder with various tokenization strategies, ranging from non-overlapping and overlapping segmentation to the integration of convolutional layers, this research introduces a novel architecture, TransIQ, for AMR tasks. A thorough examination of the TransIQ model, utilizing the RadioML2016.10b and CSPB.ML.2018+ datasets,

was presented. Moreover, the experiments underlined the necessity of balancing model size and performance, especially for deployment in resource-constrained environments like IoT devices. Comparative analysis with baseline models showed that TransIQ exhibits superiority in terms of accuracy and parameter efficiency.

Author Contributions: Conceptualization, N.R. and K.W.; methodology, N.R., G.M. and V.K.; software, N.R. and K.W.; investigation, N.R. and G.M.; resources, K.W.; writing—original draft preparation, N.R. and K.W.; writing—review and editing, N.R. and K.W.; visualization, N.R., K.W. and N.M.P. ; supervision, G.S. and H.T.; project administration, G.S. and H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: RadioML2016 is publicly available at https://www.deepsig.ai/datasets/ (accessed on 1 August 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AMR	automatic modulation recognition
IoT	Internet of Things
DN	deep learning
SNR	signal-to-noise ratio
RNNs	recurrent neural networks
NLP	natural language processing
ViT	vision Transformer
CNN	convolutional neural network
RF	Radio Frequency

References

1. Zhang, T.; Shuai, C.; Zhou, Y. Deep Learning for Robust Automatic Modulation Recognition Method for IoT Applications. *IEEE Access* **2020**, *8*, 117689–117697. [CrossRef]
2. Zhou, Q.; Zhang, R.; Zhang, F.; Jing, X. An automatic modulation classification network for IoT terminal spectrum monitoring under zero-sample situations. *EURASIP J. Wirel. Commun. Netw.* **2022**, *2022*, 25. [CrossRef]
3. Rashvand., N.; Hosseini., S.; Azarbayjani., M.; Tabkhi., H. Real-Time Bus Arrival Prediction: A Deep Learning Approach for Enhanced Urban Mobility. In Proceedings of the 13th International Conference on Operations Research and Enterprise Systems—ICORES, Rome, Italy, 24–26 February 2024; pp. 123–132. [CrossRef]
4. Khaleghian, S.; Tran, T.; Cho, J.; Harris, A.; Sartipi, M. Electric Vehicle Identification in Low-Sampling Non-Intrusive Load Monitoring Systems Using Machine Learning. In Proceedings of the 2023 IEEE International Smart Cities Conference (ISC2), Bucharest, Romania, 24–27 September 2023; pp. 1–7.
5. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
6. Gholami, S.; Leng, T.; Alam, M.N. A federated learning framework for training multi-class OCT classification models. *Investig. Ophthalmol. Vis. Sci.* **2023**, *64*, PB005.
7. O'Shea, T.J.; Roy, T.; Clancy, T.C. Over-the-air deep learning based radio signal classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. [CrossRef]
8. Krzyston, J.; Bhattacharjea, R.; Stark, A. High-capacity complex convolutional neural networks for I/Q modulation classification. *arXiv* **2020**, arXiv:2010.10717.
9. Rajendran, S.; Meert, W.; Giustiniano, D.; Lenders, V.; Pollin, S. Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 433–445. [CrossRef]
10. Ramjee, S.; Ju, S.; Yang, D.; Liu, X.; Gamal, A.E.; Eldar, Y.C. Fast deep learning for automatic modulation classification. *arXiv* **2019**, arXiv:1901.05850.
11. Hamidi-Rad, S.; Jain, S. Mcformer: A transformer based deep neural network for automatic modulation classification. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2016; pp. 1–6.
12. Usman, M.; Lee, J.A. AMC-IoT: Automatic modulation classification using efficient convolutional neural networks for low powered IoT devices. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 21–23 October 2020; pp. 288–293.

13. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
14. Rajagopalan, V.; Kunde, V.T.; Valmeekam, C.S.K.; Narayanan, K.; Shakkottai, S.; Kalathil, D.; Chamberland, J.F. Transformers are Efficient In-Context Estimators for Wireless Communication. *arXiv* **2023**, arXiv:2311.00226.
15. Wang, P.; Cheng, Y.; Dong, B.; Hu, R.; Li, S. WIR-Transformer: Using Transformers for Wireless Interference Recognition. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 2472–2476. [CrossRef]
16. O'shea, T.J.; West, N. Radio machine learning dataset generation with gnu radio. In Proceedings of the GNU Radio Conference, University of Colorado, Boulder, CO, USA, 12–16 September 2016 ; Volume 1.
17. Spooner, C.M. Dataset for the Machine-Learning Challenge [CSPB.ML.2018]. Available online: https://cyclostationary.blog/2019/02/15/data-set-for-the-machine-learning-challenge/ (accessed on 1 August 2023).
18. All BPSK Signals – Cyclostationary Signal Processing. Available online: https://cyclostationary.blog/2020/04/29/all-bpsk-signals/(accessed on 1 August 2023).
19. 3GPP. *Study on Channel Model for Frequencies from 0.5 to 100 GHz*; Technical Report (TR) 38.901, version 14.3.0 Release 14, 3rd Generation Partnership Project; 3GPP: Valbonne, France, 2020.
20. Restuccia, F.; D'Oro, S.; Al-Shawabka, A.; Rendon, B.C.; Ioannidis, S.; Melodia, T. DeepFIR: Addressing the Wireless Channel Action in Physical-Layer Deep Learning. *arXiv* **2020**, arXiv:2005.04226.
21. Liu, X.; Yang, D.; Gamal, A.E. Deep Neural Network Architectures for Modulation Classification. In Proceedings of the 2017 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 29 October–1 November 2017; pp. 915–919. [CrossRef]
22. Tekbıyık, K.; Ekti, A.R.; Görçin, A.; Kurt, G.K.; Keçeci, C. Robust and Fast Automatic Modulation Classification with CNN under Multipath Fading Channels. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–6. [CrossRef]
23. Hauser, S.C.; Headley, W.C.; Michaels, A.J. Signal detection effects on deep neural networks utilizing raw IQ for modulation classification. In Proceedings of the MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM), Baltimore, MD, USA, 23–25 October 2017; pp. 121–127.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.