Label-Efficient Learning for Radio Frequency Fingerprint Identification

Thayheng Nhem*, Fabian Denoodt*, Maarten Weyn, Michael Peeters†, José Oramas, Rafael Berkvens
IDLab - University of Antwerp - imec
Sint-Pietersvliet 7, 2000 Antwerp, Belgium
† imec - Kapeldreef 75, 3001 Leuven, Belgium
firstname.lastname@uantwerpen.be
* Equal contribution

***Abstract*—Radio Frequency Fingerprint Identification (RFFID) is a novel approach that aims to differentiate devices based on their unique signal transmissions, rather than their given identities. This approach has promising applications in wireless security, spectrum management, and sensing. Current RFFID research uses deep learning due to its success in various domains, which is largely attributed to the availability of massive labeled datasets for training. However, unlike other domains, labeled data in RFFID is limited, and the labeling process is expensive. In this paper, we propose a label-efficient learning approach for RFFID based on Contrastive Predictive Coding (CPC), a pre-training method that learns to predict future samples given the past without labels. Afterward, the model is fine-tuned to identify the device. We evaluate our approach on a fingerprint dataset of 20 devices. Our results show that CPC learns effective representations of RF signals and outperforms fully supervised learning in both classification performance and label efficiency, requiring up to 10 times fewer labels while maintaining competitive accuracy. Finally, we evaluate CPC's robustness against noise and observe competitive performance after fine-tuning.**

***Index Terms*—Radio Frequency Fingerprint Identification, Deep Learning, Contrastive Predictive Coding, Label-Efficient Learning**

I. Introduction

The landscape of wireless communication technologies is expanding rapidly, driven by the increasing demand for wireless connectivity in applications ranging from smart cities to autonomous vehicles and the Internet of Things (IoT) [1]. However, this growth also presents significant challenges in terms of security and spectrum management [2]. Radio Frequency Fingerprint Identification (RFFID) is a promising solution to these challenges. RFFID exploits the unique characteristics of wireless signals to identify and authenticate devices at the physical layer. Each device transmits signals with unique characteristics caused by hardware imperfections [3], [4]. RFFID offers a novel approach to physical layer security that complements traditional cryptography in authentication and authorization systems [5], [6]. In spectrum management, RFFID can identify specific devices or types for dynamic spectrum allocation. Additionally, RFFID enables various sensing applications by differentiating signals from different devices, which can be used for further analysis such as people counting, localization, and tracking [7], [8]. Thus, the potential applications of RFFID are vast, spanning wireless security to sensing.

Although RFFID has shown great potential, the implementation of RFFID in real-world applications remains challenging. Traditional RFFID approaches rely on feature engineering and classical machine learning which require domain expertise and are limited to the known characteristics of the RF signals [6]. Classical machine learning is also limited in capturing complex and non-linear relationships in RF signals. Furthermore, the performance of these models is highly dependent on the quality of the handcrafted features and usually suffers from environmental changes [8].

To address these challenges, researchers have turned to deep learning, which has shown great success in various domains such as computer vision, natural language processing, and speech recognition [9]. In the context of RFFID, deep learning models have outperformed traditional machine learning models [10]. However, these models face the significant challenge of requiring large amounts of labeled data for training [11]. Generating labeled data for RFFID is labor-intensive and costly, as it involves capturing RF signals from different devices and manually labeling them, or requires extensive signal processing work and domain expertise to decode and label the RF signals [12]. This process is time-consuming, labor-intensive, and costly.

There is a pressing need for label-efficient learning approaches in RFFID to reduce the amount of labeled data required for training while maintaining similar performance. In this paper, we propose integrating contrastive learning as a pretraining step to significantly reduce the amount of labeled data needed. The contributions of this paper are as follows:

- We introduce a network architecture that is pretrained using Contrastive Predictive Coding (CPC) for label-efficient learning in RFFID. This work is the first to apply CPC in this context.
- We empirically demonstrate that CPC learns useful representations of the RF signals and outperforms fully supervised learning.
- We evaluate the robustness of CPC against noisy signals with signal-to-noise ratios (SNR) ranging from -10 dB to

This research is partially funded by Research Foundation - Flanders (FWO) PESSO project under grant number: G018522N.

20 dB. Our results show that fine-tuning is a necessary step to achieve competitive robustness compared to fully supervised training.

In Section II, we discuss related works in the context of RFFID in deep learning and contrastive learning. Section III explains the CPC framework, which is integrated for label-efficient RFFID in Section IV. Our proposed pipeline is then evaluated in Section V. Finally, we conclude the paper in Section VI.

Reproducibility: To encourage reproducibility, our code, models, and commands to replicate the experiments are all available via GitHub. The dataset is available upon request.

II. Related Work

A. Deep Learning in RFFID

RFFID has been studied extensively from traditional hand-crafted features to deep learning approaches [13]. The adoption of deep learning models for such RF signal intelligence tasks has shown great success. It offers automatic feature extraction and can capture complex and unexplored relationships in the data which is much needed in RF signals.

Works such as [14], [15] have demonstrated that deep learning performs well in RFFID tasks and can classify devices with high accuracy. However, their models are trained on immense quantities of labeled data. In this work, we pose the question of whether learning with significantly fewer labeled data is possible through self-supervised pre-training.

B. Contrastive Learning

Since data labeling can be costly, recent work has investigated learning from data with fewer labels. One such method is contrastive learning. Contrastive methods learn an encoding function $g(x)=c$ such that after training, a downstream task $T(\cdot)$ can use the obtained representation c to obtain its prediction $\hat{y}$, rather than by using x directly. The encoder function $g(\cdot)$ is typically learned from data without labels. The downstream task should then receive an input representation that is easier to work with, and can therefore learn from only a subset of labeled data, rather than the full dataset needed by conventional fully supervised learning.

Interestingly, these approaches have outperformed their fully supervised counterparts in various domains, including computer vision [16]–[18], speech recognition [17], [19], [20], and natural language processing [21], all while requiring fewer labeled data. Meanwhile, contrastive learning has not yet seen widespread adoption by the RFFID community.

Two predominant contrastive methods are SimCLR [16] and Contrastive Predictive Coding (CPC) [17]. In both methods, given an input x, $g(\cdot)$ learns to produce a representation c such that a classifier can easily detect a "positive" sample x^+ from a "negative" sample x^-. It is important to note that these contrastive methods succeed or fail depending on the quality of how x^+ and x^- are defined. Their definition differs between methods but is typically done by hand, and their definition has frequently focused on image data, which is not directly applicable to RF data.

In the work by [22], a contrastive method is proposed for RFFID by defining x^+ as augmentations of the signal in the form of rotation, flipping, adding Gaussian noise, and shifting. However, these augmentations are inspired by existing methods from the image domain and may not be optimal in the context of RFFID. For example, rotating or flipping a signal may entirely change the underlying information of the signal. This shows the challenges of directly applying image-related augmentations to RF data, where meaningful augmentations are not as easily defined as they are for images.

Meanwhile, CPC does not explicitly define the positive samples x^+ through data augmentations. Instead, CPC obtains representations by assuming temporal data that share mutual information with temporally nearby patches of data, commonly referred to as "slow features" [23]. It, therefore, has the benefit of not being dependent on the quality of human-defined augmentations but is limited to temporal data that obey this "slow features" property.

In this work, we examine whether this property is present in RF data by evaluating whether CPC can learn meaningful representations from this data. To the best of our knowledge, no prior work has explored CPC in the context of RFFID.

III. Background: Contrastive Predictive Coding

In the context of CPC, a temporal signal x is split into patches $x_1, x_2 \dots x_n$, such that for a patch x_t, its positive counterpart x^+, is defined as a successive patch x_{t+k} with $k \in \{1 \dots K\}$ and x^- as a random sample from a set X consisting of random patches. Hence, CPC is tasked to predict the future given the past by learning to discriminate future patches from random ones. This results in CPC's representations to preserve the global information of the signal that is common among neighbors, and discard local noise irrelevant to the task [17]. Note that this scheme of predicting the future given the past is only applicable for data that follows the "slow features" property, such as speech (as described in II-B). For instance, a patch of raw speech lasting a few milliseconds may share information such as the speaker's identity, emotion, and phonemes, which may not necessarily be shared with random patches drawn from other utterances [19]. In this work, we pose the question if this "slow features" data assumption is also present in RF data, and can therefore be used in the context of device identification.

A. Architecture

The encoder function $g(x) = c$ is a neural network consisting of two blocks:

$$g_{enc}(x_t) = z_t \tag{1}$$

$$g_{ar}(z_1, \dots z_t) = c_1 \dots c_t \tag{2}$$

First, a series of patches $x_1 \dots x_n$ are encoded independently from each other into $z_1, \dots z_n$ through a convolution block $g_{enc}(\cdot)$, and afterward, an autoregressive block $g_{ar}(\cdot)$ aggregates the results to obtain n context representations $c_1 \dots c_n$. Finally, a single context vector c is obtained as follows:

$$\text{avg pool}(c_1, \cdots c_n) = c \tag{3}$$

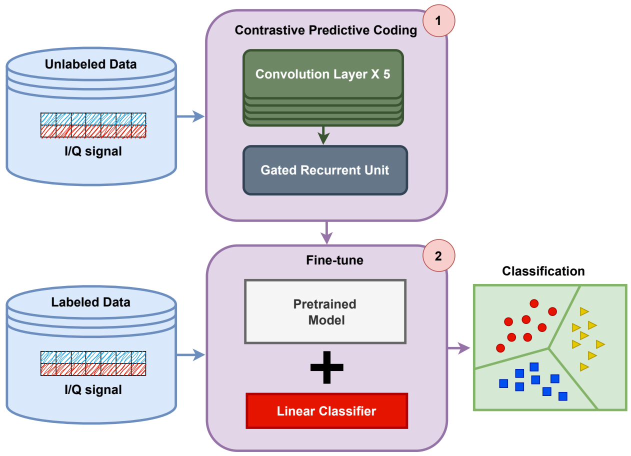

Fig. 1. The proposed pipeline: 1) a backbone architecture is pre-trained with CPC on unlabeled RF data, and 2) afterward, the backbone is extended with an additional fully connected layer and further fine-tuned on a small labeled dataset for device identification.

The resulting average-pooled vector serves as input for the downstream task.

B. Loss Function

CPC must learn to discriminate "positive" future patches from "negative" random ones. This is achieved through K scoring functions $f_k : \mathcal{Z} \times \mathcal{C} \to \mathbb{R}_{>0}$, which rate the similarity between two patches, defined as follows:

$$f_k\left(z_j, c_t\right) = \exp\left(z_j^T W_k c_t\right) \tag{4}$$

Each matrix W_k is learned together with g_{enc} and g_{ar} by optimizing the following InfoNCE objective through stochastic gradient descent:

$$\mathcal{L} = -\sum_{k=1}^{K} \mathbb{E}_{X}\left[\log \frac{f_k\left(z_{t+k}, c_t\right)}{\sum_{z_j \in X} f_k\left(z_j, c_t\right)}\right] \tag{5}$$

Hence, through this loss, CPC enforces f_k to obtain high scores for temporally nearby patches, while discouraging them for random ones. Note that this learning objective only makes sense in conditions where the "slow features" property holds.

IV. Methodology

We discuss how to incorporate CPC in our pipeline for RFFID. An overview is shown in Figure 1. First, a randomly initialized backbone is pre-trained using CPC on unlabeled RF data. Afterward, an additional fully connected layer is appended to the backbone and further fine-tuned for device identification with limited labeled data.

A. Dataset

We individually collect I/Q (In-phase and Quadrature) samples from 20 devices. The setup consists of an access point operating IEEE 802.11 b/g and a USRP (Universal Software Radio Peripheral) B210 [24] receiver. Each device is connected to the access point and sends out random data frames where USRP is used to capture those RF signals. The transmitter is placed five meters apart from the receiver in an office environment. We collected 5000 frames from each device, in total, 100,000 frames. We use the train-val-test split ratio of 70:10:20, resulting in 70,000, 9,900, and 20,100 examples for the train, validation, and test sets, respectively. Each example has a shape of 2×2080. The data is relatively large, allowing us to validate generalization capabilities on training subsets while detecting potential stagnation in performance. Using a smaller subset may already provide enough information to achieve good performance.

TABLE I
ARCHITECTURE FOR THE EXPERIMENTS.

Module	Layer	Output	Kernel	Stride	Padding
$g_{enc}(\cdot)$	Input	2×2080			
	Conv	512×415	10	5	2
	Conv	512×103	8	4	2
	Conv	512×104	4	1	2
	Conv	512×105	4	1	2
	Conv	512×104	4	1	1
$g_{ar}(\cdot)$	GRU	256×104	-	-	-

B. Learning Generic RF Representations via CPC

The representations are obtained through a CPC architecture, consisting of of a series of convolution layers and a final Gated Recurrent Unit, as shown in Table I. The predictions are made on the full 2×2080 inputs, resulting in a 256×104 tensor. Overall, the architecture we propose here is a modification of the original from [17]. The original architecture significantly reduces the temporal resolution. In the context of RFFID, we observed that shrinking the temporal resolution less significantly (from 2080 to 104, a 20-fold reduction) improved performance compared to the stronger temporal shrinkage from [17]'s architecture, which would shrink 2080 to 13 time steps (a 160-fold reduction), resulting in too much crucial local information being lost by CPC's framework. Secondly, we ensured enough overlap in the temporal representations by choosing a stride of half the kernel size. As such, there is a 50% overlap between directly subsequent patches. ReLU and batch norm are applied after each layer, except for the final one.

The encoder is trained using the Adam optimizer for 100 epochs with a learning rate of 2×10^{-4} and a batch size of 64. The maximum number of patches to predict in the future, K, is set to 64. Implementation details regarding drawing negative samples x^- for $f(\cdot)$ remain identical to the procedure in CPC [17].

C. Device Classification

After training the CPC model, the representations from the backbone are used for classification. The representation obtained from an RF signal is first average-pooled over the temporal dimensions, resulting in a fixed-length vector ($c \in \mathbb{R}^{256}$). This vector serves as the input for a classifier, which consists of a single fully connected layer without activation

TABLE II
CLASSIFICATION PERFORMANCE. ALL MODELS USE THE SAME INPUT SIZE AND ARCHITECTURE.

Method	**Accuracy (%)**	**CrossEntropy**
Supervised	96.28 $\pm$ 0.62	0.1 $\pm$ 0.01
Randomly Initialized	5.43 $\pm$ 0.31	2.9 $\pm$ 0
CPC Frozen	98.2 $\pm$ 0.33	0.07 $\pm$ 0.01
CPC Fine-tuned	98.73 $\pm$ 0.43	0.04 $\pm$ 0.02

functions. Both the pre-trained backbone and the classification layer are fine-tuned end-to-end on a smaller labeled dataset using cross-entropy loss, with the Adam optimizer, a learning rate of 2×10^{-3}, and a batch size of 64, for 50 epochs.

V. EXPERIMENTS AND EVALUATION

The experiments are constructed to answer the following three questions: *a)* Is the "slow features" property present in RFFID data, and can CPC obtain meaningful representations for device identification? *b)* How does subset training impact performance in label-efficient learning compared to fully supervised training? *c)* How does noise affect robustness compared to fully supervised training?

We evaluate two CPC models; one where the pretrained backbone is frozen and only the final classifier layer is trained for RFFID, denoted as "CPC Frozen", and a second where the backbone is also trained, denoted as "CPC Fine-tuned". The baselines consist of *1)* a fully supervised model with identical architecture that is trained end-to-end without pretraining and *2)* a frozen backbone with randomly initialized weights, where only the weights from the final classification layer are trained for classification. All methods are trained with the same architecture and hyperparameters. However, the fully supervised model is trained for a duration of 150 epochs rather than 50 to have a fair comparison against CPC's training scheme consisting of 100 epochs of CPC training and 50 epochs of fine-tuning.

A. Classification Performance

To observe whether CPC can learn meaningful representations, we train both the encoder and classifier on the full datasets. The results are shown in Table II. Both CPC models; CPC Frozen and CPC Fine-tuned consistently outperform their fully supervised counterparts, obtaining 98.20% and 98.73% versus the fully supervised model's accuracy of 96.28%. Meanwhile, the randomly initialized backbone obtains an accuracy of 5.43%, which appears no better than random guessing. This large discrepancy with CPC Frozen indicates that CPC does indeed learn good representations for RFFID, indicating that the slow-features assumption, required by CPC's InfoNCE objective, is indeed present in RF data.

B. Label Efficiency

To analyze the impact of training on fewer labeled data, we train the classifier on different dataset sizes; starting from merely 70 training examples up to 70,000 examples. The

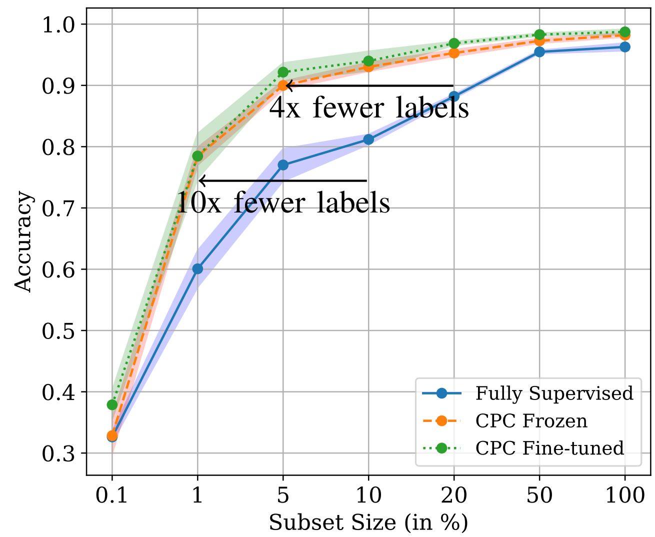

Fig. 2. Accuracy when training with 70, 700, 3,500, 7,000, 14,000, 35,000, and 70,000 training examples, respectively. Both CPC models require significantly fewer training examples while maintaining similar or better performance than fully supervised.

accuracy results for the different training sizes are shown in Fig. 2. We observe that CPC Fine-tuned obtains comparable performance to fully supervised when using up to 10 times fewer labeled data, showing that CPC is an effective approach for label-efficient learning in RFFID. Moreover, at all sizes up to 50%, both CPC Frozen and CPC Fine-tuned show a significant improvement compared to fully supervised. In particular, when training on 1% of the data (or 700 data points), both reach a near 20% improvement, with CPC Frozen and CPC Fine-tuned both exactly 78.46% versus supervised's 60.07%.

Interestingly, fine-tuning CPC's backbone does not appear to hurt label efficiency, which is a remarkable result. The improved generalization from CPC Frozen is somewhat expected as the frozen backbone reduces the degrees of freedom of the model, thereby reducing the space of plausible models and thus making it less prone to overfitting. However, despite CPC Fine-tuned having equal degrees of freedom to fully supervised, and thus an equally complex architecture, it does not appear to be as prone to overfitting, as evidenced by its more consistent test accuracy across the different subset sizes.

C. Noise Robustness

To validate the robustness of the model, we add Additive White Gaussian Noise (AWGN) to the test set at different SNR levels from -10 dB to 20 dB with increments of 5 dB. The impact of induced noise on the inputs is shown in Figure 1. CPC Frozen appears to be less resilient to low SNR conditions compared to the fully supervised model and its fine-tuned counterpart. However, fine-tuning seems to resolve this issue, resulting in robustness equal to that of a fully supervised case. Thus, fine-tuning in CPC appears to be a necessary step to ensure reliable predictions in noisy environments.

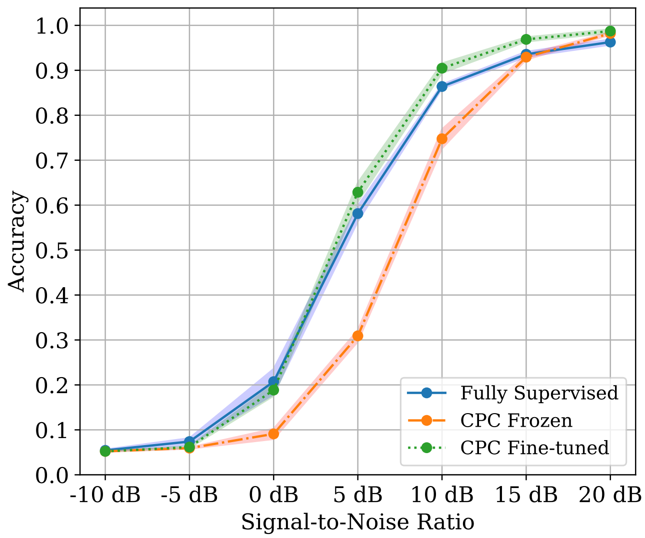

Fig. 3. Performance of each training model under different SNR conditions where Fully Supervised Model is more robust than CPC with only downstream task. However, fine-tuning CPC is more robust against noise above 0 dB.

We hypothesize that the discrepancy between CPC Frozen and fully supervised models is due to CPC learning more generic representations that are usable for different downstream tasks. However, more experiments on different downstream tasks in RF intelligent tasks are required. The preserved information in these representations may contain dimensions that are not necessarily useful for device identification, resulting in a limited capacity for dimensions useful to the task. In contrast, fully supervised models optimize all dimensions with only a single task in mind, potentially allowing for more redundancy in the features. Consequently, the fewer dimensions in CPC Frozen that actually contribute to prediction may be more prone to internal miscalculations in the neural network, as there is less redundancy to recover from mistakes.

The improved robustness from CPC Fine-tuned supports this hypothesis, as the accuracy improves after fine-tuning, likely discarding irrelevant information and allowing more capacity for features useful for device identification.

VI. Conclusion

As a solution to the costly labeling process in RFFID, this paper proposes a label-efficient learning approach for RFFID based on CPC. We show that competitive performance can be maintained when learning from up to 10 times fewer labeled data. Moreover, when trained on similar dataset sizes, our method consistently outperforms the conventional fully supervised model, demonstrating that contrastive predictive coding is a viable method for self-supervised learning in RFFID.

While this work shows that the amount of labeled data can be drastically reduced, the number of devices from which the data was generated has remained identical. As generating data from different devices is costly, future research could explore the model's ability to generalize from a limited set of devices to a broader and more varied set.

References

[1] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[2] P. Angueira, I. Val, J. Montalbán, Seijo, E. Iradier, P. S. Fontaneda, L. Fanari, and A. Arriola, "A survey of physical layer techniques for secure wireless communications in industry," *IEEE Communications Surveys Tutorials*, vol. 24, no. 2, pp. 810–838, 2022.

[3] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, MobiCom '08, (New York, NY, USA), p. 116–127, Association for Computing Machinery, 2008.

[4] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

[5] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. Cavallaro, "Radio frequency fingerprint identification for security in low-cost iot devices," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 309–313, 2021.

[6] J. Zhang, G. Shen, W. Saad, and K. Chowdhury, "Radio frequency fingerprint identification for device authentication in the internet of things," *IEEE Communications Magazine*, vol. 61, no. 10, pp. 110–115, 2023.

[7] N. Bhaskar, "A survey of techniques in passive identification of wireless personal devices and the implications on user tracking," 2020.

[8] A. Jagannath, J. Jagannath, and P. S. P. V. Kumar, "A comprehensive survey on radio frequency (rf) fingerprinting: Traditional approaches, deep learning, and open challenges," *Computer Networks*, vol. 219, p. 109455, 2022.

[9] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[10] L. Xie, L. Peng, J. Zhang, and A. Hu, "Radio frequency fingerprint identification for internet of things: A survey," *Security and Safety*, vol. 3, 09 2023.

[11] S. Alhazbi, A. M. Hussain, S. Sciancalepore, G. Oligeri, and P. Papadimitratos, "Challenges of radio frequency fingerprinting: From data collection to deployment," *ArXiv*, vol. abs/2310.16406, 2023.

[12] S. Hanna, S. Karunaratne, and D. Cabric, "Wisig: A large-scale wifi signal dataset for receiver and channel agnostic rf fingerprinting," *IEEE Access*, vol. 10, pp. 22808–22818, 2022.

[13] S. Abbas, M. A. Talib, Q. Nasir, S. Idhis, M. Alaboudi, and A. Mohamed, "Radio frequency fingerprinting techniques for device identification: a survey," *International Journal of Information Security*, 4 2023.

[14] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 146–152, 2018.

[15] G. Shen, J. Zhang, A. Marshall, M. Valkama, and J. R. Cavallaro, "Toward length-versatile and noise-robust radio frequency fingerprint identification," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2355–2367, 2023.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[17] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019.

[18] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *International conference on machine learning*, pp. 4182–4192, PMLR, 2020.

[19] S. Löwe, P. O'Connor, and B. Veeling, "Putting an end to end-to-end: Gradient-isolated learning of representations," *Advances in neural information processing systems*, vol. 32, 2019.

[20] F. Denoodt, B. de Boer, and J. Oramas, "Smooth infomax – towards easier post-hoc interpretability," *arXiv preprint arXiv:2408.12936*, 2024.

[21] R. Zhang, Y. Ji, Y. Zhang, and R. J. Passonneau, "Contrastive data and learning for natural language processing," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pp. 39–47, 2022.

[22] Y. Peng, C. Hou, Y. Zhang, Y. Lin, G. Gui, H. Gacanin, S. Mao, and F. Adachi, "Supervised contrastive learning for rff identification with limited samples," *IEEE Internet of Things Journal*, vol. 10, no. 19, pp. 17293–17306, 2023.

[23] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural computation*, vol. 14, no. 4, pp. 715–770, 2002.

[24] Ettus, "B210 datasheet." https://www.ettus.com/wp-content/uploads/2019/01/b200-b210_spec_sheet.pdf. Accessed: August 27, 2024.