VibroFM: Towards Micro Foundation Models for Robust Multimodal IoT Sensing

Tomoyoshi Kimura†, Jinyang Li†, Tianshi Wang†, Yizhuo Chen†, Ruijie Wang†, Denizhan Kara†
Maggie Wigness*, Joydeep Bhattacharyya*, Mudhakar Srivatsa††, Shengzhong Liu‡,
Mani Srivastava§, Suhas Diggavi§, Tarek Abdelzaher†
†*University of Illinois at Urbana-Champaign,* **DEVCOM Army Research Laboratory*
‡*Shanghai Jiao Tong University*, ††*IBM T. J. Watson Research Center*
§*University of California, Los Angeles*
{tkimura4, jinyang7, tianshi3, yizhuoc, ruijiew2, kara4}@illinois.edu,
{maggie.b.wigness, joydeep.bhattacharyya}.civ@army.mil, msrivats@us.ibm.com,
shengzhong@sjtu.edu.cn, {mbs, suhas}@ee.ucla.edu, zaher@illinois.edu

2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS) | 979-8-3503-6399-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/MASS62177.2024.00014

***Abstract*—The paper argues for the feasibility and utility of *micro foundation models* (μFMs), a key direction for future smart IoT/CPS systems that exploits advances in self-supervised pre-training to support multiple downstream tasks. We demonstrate key beneficial properties such as latent representation independence from the downstream task, robustness to domain shifts, and ability to learn from unlabeled data. Importantly, we demonstrate the emergence of these properties after pre-training with only *moderate* amounts of unlabeled data, earning the name μFMs. To make the argument, evaluate model efficacy, and surface some of the underlying challenges, this paper describes a vibration-based μFM, called VibroFM, pre-trained with moderate amounts of *unlabeled acoustic and seismic sensing data*, to support target classification and tracking applications. VibroFM is pre-trained in an environment-agnostic fashion using *unlabeled* sensor data. It can then be fine-tuned to a given deployment using a small amount of in-situ labeled data. The paper shows that VibroFM (i) improves the robustness of several downstream tasks, (ii) efficiently adapts to different environmental conditions (using only small amounts of fine-tuning), and (iii) allows few-shot generalization to unseen targets. We further show that VibroFM can execute in real time on *embedded sensor nodes*. We compare the robustness and performance of VibroFM to conventional supervised deep neural networks, showing the advantages of the former. Combined with the feasibility of executing μFMs in resource-limited settings and the sufficiency of only moderate amounts of data for their pre-training, we conclude the importance of micro foundation models as a promising research direction for the IoT/CPS community.**

***Index Terms*—Foundation Model, Self-Supervised Learning, Internet of Things**

I. Introduction

The paper defines and argues for the importance of a class of domain-specific foundation models we call *micro foundation models* (μFMs) needed to overcome some of the training and robustness challenges of intelligent IoT/CPS systems. We argue for the feasibility of pre-training μFMs from scratch on resources available to a broad range of institutions, in contrast to the prohibitive amounts of resources needed to train the next generation of, say, large language models (LLMs) or vision language models (VLMs). To make these arguments, we present and evaluate VibroFM, a new vibration-based μFM developed and trained by the authors to support target classification and tracking applications based on acoustic and seismic sensing.

The success of foundation models (FMs) [1] in the areas of natural language processing and computer vision has led to generalizations of the foundation model concept to other domains, where significant amounts of unlabeled data exist that can be used for self-supervised pre-training. One such domain is *IoT applications*. While many of today's foundation models, such as GPT-4 [2] and LLava [3], are very large, calling for amounts of pre-training resources that exceed the capacity of most research institutions, the paper shows that properties of foundation models, such as robust generalization and independence from downstream tasks, emerge at a much smaller scale, thus imparting application benefits at realistic cost. To illustrate this point, VibroFM (originally presented by the authors as a workshop publication [4]) is (i) extended with a new (lighter) encoder, (ii) used to investigate the impact of pre-training data volume on resulting model quality, and (iii) adapted in the evaluation to multiple downstream inference tasks and new targets, concluding that the pursuit of μFMs is both technically feasible and operationally beneficial, as key foundation model properties emerge at only a moderate training scale.

We choose acoustic and seismic sensing data modalities because of the core IoT-centric challenges that these modalities exemplify. Namely, such modalities are particularly sensitive to environmental factors, conflating target signatures with environmental effects. Even in the same application domain, such as target tracking, a target (*e.g.*, some vehicle on a road) may generate different acoustic and seismic signatures depending on a variety of environmental factors, such as the type of terrain (paved road, gravel, sand, etc.), background noise (rain, wind, construction, traffic, etc.) and various natural and/or human disturbances. Training an inference task (*e.g.*, a target classifier) to handle all such contingencies is a daunting undertaking. These challenges have no direct correspondence in several mainstream AI contexts such as text inputs (for

2155-6814/24/$31.00 ©2024 IEEE
DOI 10.1109/MASS62177.2024.00014

LLMs), where a label, such as "Ford Mustang", always denotes the same car regardless of the context in which the car is described. The confounding effects of the environmental context are also less prominent in vision, where the car will have similar visual features regardless of the type of road it is on and regardless of the background landscape. In that sense, IoT time-series data modalities are arguably more challenging as both the target and environment features get superimposed onto the same input stream.

The challenge with disentangling time-series data (*e.g.*, acoustic and seismic data) into the underlying signatures of different targets and environmental factors often frustrates traditional *supervised* learning solutions. Such solutions (for intelligent IoT applications) are label-hungry. Labeled data must be collected not only on a sufficient set of targets but also in a sufficiently representative set of environmental conditions. In the absence of sufficient amounts of labeled data, supervised DNN training techniques suffer from overfitting, thereby dramatically reducing the robustness of run-time inference [5]. In contrast, by obviating the need for labeled data in pre-training (requiring small amounts of labels for fine-tuning), we show that VibroFM improves inference robustness and adaptation to domain shifts, environmental noise, and new targets.

The rest of the paper is organized as follows. We discuss micro foundation models in more detail in Section II, followed by a description of our running case study and experimental set-up in Section III. Section IV presents evaluation results of model robustness, run-time efficiency, and effect of pre-training data size. Section V discusses the main takeaway points from this paper. Section VI covers related work. The paper concludes with Section VII.

II. μFMs: Scope, Advantages, and Pre-training

In this section, we define our concept of *micro foundation models*, describe pre-training (and fine-tuning) approaches, and identify some of the challenges and research opportunities in that space.

A. Definition

Foundation models are task-independent neural network models trained in a self-supervised fashion on large amounts of unlabeled data to encapsulate knowledge in a given field. Large language models (LLMs) are a common example of foundation models, but the concept extends to other application domains, such as security [6], [7], networking [8]–[10], and meteorology [11], to name a few. We assume that the reader is already familiar with the concept of foundation models. A great description of this concept is found in the original paper that popularized it [1]. For the purposes of this paper, we define *micro* foundation models as foundation models that satisfy the following constraints:

- *Domain-specific:* We consider domain-specific models. An example would be a foundation model for medical image analysis, urban traffic monitoring, human activity recognition, network security, or a similarly targeted domain. In an IoT/CPS context, this allows the model to specialize in knowledge pertinent to the target domain only, thus improving tractability.
- *Modality-specific:* The model uses, as input, only a small pre-specified range of sensing modalities. This is especially important for IoT/CPS applications, where the number of possible sensing modalities can be vast, thus calling for specialization to allow effective pre-training.
- *Self-supervised:* They are pre-trained in a self-supervised manner (i.e., using *unlabeled* data). This is a core property of all foundation models and is a key enabler that allows us to circumvent the need for scarce labeled data.
- *Task-agnostic:* Their pre-training is agnostic to downstream (inference) tasks. This is another core property that separates foundation models from machine learning solutions customized for an individual task. For example, a μFM for urban traffic monitoring might need to enable (i) classification of different objects in the urban environment, (ii) localization of these objects, (iii) speed estimation, etc. The same underlying model (with the possible exception of a small task-specific head or layer) should support all these tasks. We call this property *task-agnostic* as opposed to *task-independent* because we want to allow for some application-specific bias in training. For example, if the set of application tasks has to do generally with foreground objects, it may be OK for pre-training to ignore background features. In other words, some *inductive bias*, informed by the application domain, is acceptable.
- *Moderately-sized:* They use a moderate number of model parameters (in the millions, not billions) and correspondingly moderate amounts of data for pre-training. Beyond that, we are intentionally vague on the notion of *moderate* as it might be application-specific.

B. Advantages

Unlike supervised training techniques that directly teach a neural network *how to perform a particular inference task*, pre-training a foundation model aims to teach the neural network *a better internal representation of domain-specific and modality-specific data*. The internal representation encodes higher-level semantics or "knowledge" of the domain, extracted from the specified data modality as input. As mentioned above, three further features characterize the pre-training of (micro) foundation models. First, pre-training is *self-supervised*; no labeled data are needed. Second, it is *task-agnostic*; it does not know the downstream inference task(s) and, as such, can in principle support several different tasks, deployments, or environments. Finally, specific to μFMs, we explore pre-training with *moderate* amounts of data. We demonstrate the emergence of useful model properties despite the moderate pre-training data scales. The feasibility of pre-training with moderate amounts of data without data labels and without knowing the exact downstream task(s) makes the approach attractive to IoT applications. First, unlabeled data are a lot easier to collect in IoT settings than labeled data (due to the lack of interpretability of many data modalities, such

as vibration or RF signatures, and thus difficulties labeling collected data after the fact). Self-supervised pre-training is therefore highly advantageous. Second, the independence of pre-training from downstream tasks makes the approach easily customizable to changes in model use and deployment conditions. We show that the pre-trained model can be fine-tuned with only a minimal amount of labeled data for a specific downstream deployment, allowing for more robust task performance than baseline (supervised) approaches. Third, the moderate size of micro foundation models makes them compatible with the computational limitations of IoT devices. Rapid advances in machine learning have led to increasingly larger DNNs [12]. However, many IoT devices remain limited by their resource constraints [13]. These devices, from simple sensors to complex wearables, often lack the necessary processing power, memory, and energy efficiency to support the operation of large-scale DNNs in real time. This discrepancy poses significant challenges for deploying advanced DNNs in IoT applications [14]. We show that the pre-trained model we use is capable of real-time execution on a Raspberry-Pi class of devices. It is also shown to have a higher fine-tuning efficiency and a lower memory consumption, while offering more robust performance, compared to its supervised counterparts.

C. Model Pre-training

Pre-training foundation models can be conceptualized as an act of *encoding* or mapping input domain data into a multi-dimensional latent space that is better organized semantically. This is usually referred to as input data *embedding*. The improved semantic organization simplifies solving downstream inference tasks. For example, if input measurements of similar phenomena land closer in some dimension of the latent space then it becomes easier to identify a phenomenon simply based on its embedding location. While many techniques were proposed recently for self-supervised pre-training of foundation models, two are particularly widespread: (i) learning to reconstruct masked [15], [16] (or distorted [17]) inputs and (ii) contrastive learning [18]–[21]. They differ in the way they train the model useful concepts from the domain, without the need for labeled data.

Specifically, *masking/distortion* removes or distorts parts of the input, and then rewards the model for correct reconstruction of these parts. Clearly, a model that learns correct reconstruction from partial data must have encapsulated some knowledge about the target domain. Some language models (e.g., BERT [15]) are trained by input masking and reconstruction.

Unlike masking, *contrastive learning* teaches the model what "similarity" means in the target domain (by contrasting similar and dissimilar sample pairs), such that similar inputs are grouped closer together in a latent space. To do so without labels, it often relies on semantics-invariant input transformations that convert individual input samples to "similar" ones (without necessarily knowing what the sample labels are) to contrast with random pairs (that are likely less similar). An example of such transformations in vision is image resizing; an image and its resized version are more similar than two random images. An example in time-series data is adding simulated noise. The result of rewarding the model for putting similar samples closer together in the latent space is a well-organized learned latent representation, where proximity implies semantic similarity.

D. IoT/CPS-Specific Pre-training Challenges

Several peculiarities of data collected in IoT/CPS applications call for re-thinking of the mainstream foundation model pre-training pipelines, described above. For example, IoT/CPS data typically represent sensor measurements of physical phenomena. These phenomena have better representations in the frequency domain [22], thus favoring contrastive learning and masking solutions that are optimized for frequency domain data. Examples of such optimizations have been recently described for contrastive learning [21] and masking [23], respectively.

Spectrograms are commonly used representations of frequency domain data. While superficially similar to images, spectrograms differ in many respects from other visual inputs, calling for customized solutions beyond mainstream image-based pre-training. For example, observing a visually similar object (e.g., a car) at different locations in an image does not usually affect its classification, whereas observing a similar visual pattern in different locations of the spectrogram usually implies differences in class because the underlying signal frequency range is different. Also, in spectrograms, some frequency ranges may be more important than others depending on the temporal dynamics (and thus spectral frequencies) of phenomena in the underlying application domain and the nature of environmental noise.

For another example of differences of IoT/CPS data, such data are often multimodal. Unlike commonly explored modalities in mainstream AI, such as text, images, and video, IoT data may feature other modalities including accelerometer, gyroscope, or geophone data. These modalities call for new notions of sample similarity and entail different latent space architectures to capture both modality-specific and cross-modality information [24]. The optimization of foundation model pre-training pipelines to the needs of IoT/CPS applications is thus a key research topic for the intelligent (IoT/CPS) systems community. Some previous work has started to address the topic [23], [24], but the field is new and more research is needed, which is beyond the scope of this paper. Below, we merely show that an optimized pre-training pipeline is able to demonstrate beneficial foundation model properties at moderate model and training data scales.

E. A Vibrometry μFM

To experiment with an example μFM (we call VibroFM), we train it from acoustic and seismic data using a contrastive learning framework, called FOCAL [24], recently proposed by the authors for (pre-training) intelligent multimodal sensing applications. FOCAL pre-trains *an encoder* to extract a structured latent representation of the input multimodal data. This

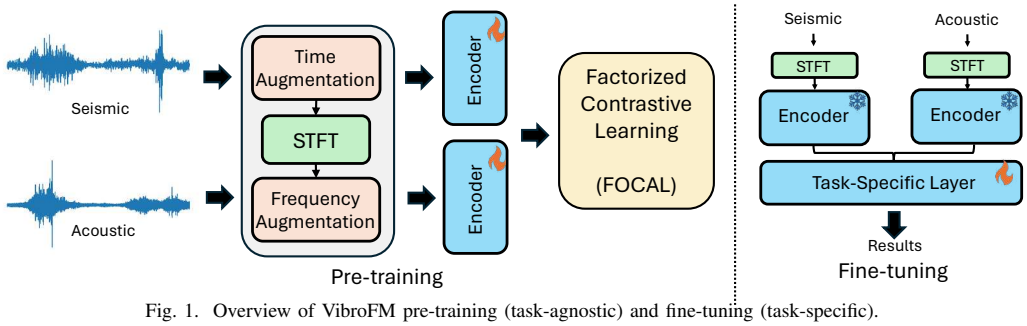


Fig. 1. Overview of VibroFM pre-training (task-agnostic) and fine-tuning (task-specific).

TABLE I

TRAINING CONFIGURATIONS: BELOW, WE DETAIL THE TRAINING PARAMETERS INCLUDING THE BATCH SIZE (NUMBER OF SAMPLES PER BATCH), THE OPTIMIZER FOR UPDATING MODEL PARAMETERS, THE INITIAL LEARNING RATE (LR), AND THE LR SCHEDULER FOR DYNAMIC LR ADJUSTMENTS, ALONGSIDE ITS LR DECAY RATE. THE TABLE ALSO LISTS THE TOTAL TRAINING EPOCHS AND THE DATA AUGMENTATIONS APPLIED.

Stage	Batch Size	Optimizer	Initial LR	LR Scheduler	LR Decay	Epochs	Augmentations
Supervised	128	AdamW [25]	1e-4	Cosine [26]	0.2	500	Mixup, Phase Shift
Pre-train	256	AdamW [25]	0.0001	Cosine [26]	0.05	6000	Permutation, Negation, Time Warp, Horizontal Flip, Magnitude Warp, Scaling, Phase Shift
Fine-tuning	256	Adam [27]	1e-3	Cosine [26]	0.2	200	Mixup, Phase Shift

latent representation separates shared and private subspaces. The shared subspace contains common information shared across the different sensing modalities. The private subspace holds additional modality-exclusive information by contrasting different augmented views. An orthogonality constraint is applied among the private subspaces, as well as between each private subspace and the shared subspace to enforce information independence among these subspaces.

An overview of VibroFM training is shown in Figure 1. We first utilize FOCAL to pre-train VibroFM with three popular DNN encoders (DeepSense [28], SWIN-Transformer, abbreviated as SW-T [29], and TSMixer [30]) on a multi-modal Moving Object Detection [24] (MOD) dataset. During pre-training, we randomly select time and frequency augmentations to create multiple views for modality-exclusive contrastive learning. We use STFT (Short Time Fourier Transform) to convert each sample into the frequency domain and then extract the embedding of each modality. The training configurations used are presented in Table I. We also use the same setup to pre-train a larger-scale version of VibroFM, denoted as VibroFM-Large, with additional data collected. A brief size comparison of the two models is shown in Table II.

For testing, we perform a two-day deployment experiment in a real-world neighborhood as a case study to examine the performance of VibroFM. The pre-training data did *not* include any data from that deployment. To experiment with the robustness of the pre-trained model, we freeze the pre-trained model and append a single linear layer for fine-tuning. We *fine-tune* this linear layer on part of the labeled data collected in the new deployment and *test* the fine-tuned model's performance under the same or different deployment conditions. We would like to note that only the linear layer is trained at the fine-tuning stage. During fine-tuning, we apply mixup [31] augmentation in the time domain and phase shift augmentation in the frequency domain. We also separately train supervised DNNs for the three backbone encoders as the benchmarks. The supervised model contains an additional fusion layer to fuse the modality embeddings for classification. Training configurations for fine-tuning and supervised benchmarks are shown in Table I. We also use a supervised model initially trained on the MOD dataset and later fine-tuned on its final classification layer, mirroring VibroFM's fine-tuning approach. We call it the supervised fine-tuned baseline.

III. TESTING μFM PROPERTIES: A CASE STUDY

Experiments with VibroFM were conducted at an outdoor research facility located on (repurposed) state park grounds. Sensors were deployed and vehicles were driven nearby. Figure 2 shows a satellite view of the test facility and the locations of sensor nodes. Nodes 1 & 4 utilized the RaspberryShake[1] 4D, model 4B Rev 1.4, while Nodes 2 & 3 utilized the RaspberryShake 1D, model 4B Rev 1.5. Each node featured a geophone and a microphone array, collecting seismic and acoustic vibration signals from nearby objects. In each run, a specific target navigated the neighborhood, passing the sensors in some arbitrary order within a short time window. Four distinct target types were used: (i) a Polaris[2] off-road vehicle, (ii) a Warthog[3] all-terrain unmanned ground robot, (iii) a Husky unmanned outdoor field robot[4], and (iv) a standard civilian automobile.

A. Datasets

For pre-training, we first consider the MOD dataset released in [24]. MOD consists of multi-modal acoustic and seismic signals collected from sensors deployed in different urban and rural environments that varied in terrain (paved, gravel, dirt,

[1]https://raspberryshake.org/
[2]https://www.polaris.com/
[3]https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/
[4]https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/

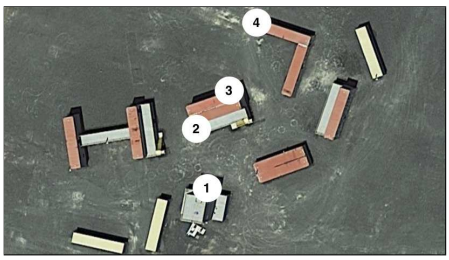


Fig. 2. The satellite view of the case study neighborhood with labeled nodes.

TABLE II
COMPARISON OF DATASET SIZES FOR EVALUATION SET, MOD, AND VIBROFM-LARGE.

Metric	Evaluation Set	MOD	VibroFM-Large
Labeled	Yes	No	No
Hours	7.02	21.18	66.11
Number of Samples	12635	38116	119000

rooftop, etc) and environmental conditions (quiet, windy, etc), recording the passage of a variety of target types, mostly focusing on civilian automobiles, bikes, and humans. We follow the same setup as [24] with a 0.2-second overlapping ratio between 2 seconds samples of 8000Hz acoustic 100Hz seismic data. We partition MOD into a set of unlabeled data used to pre-train the FM and a set of labeled data for supervised training and fine-tuning.

To experiment with the impact of training data size, we increase the scale of the pre-training dataset by acquiring additional seismic and acoustic signals from four different civilian cars, collected across three domains distinct from those in the MOD dataset. We process these new data to match the format of the MOD dataset and integrate it with the MOD dataset, expanding the pre-training set to threefold its original size. We call this expanded set VibroFM-Large. We call the data collected during experimental evaluation the Evaluation Set. Statistical comparison between each set is presented in Table II. We use MOD and Evaluation Set to evaluate VibroFM's run-time robustness and efficiency compared to supervised methods. Then, we use VibroFM-Large to analyze VibroFM at a larger scale (pre-trained with more data) on additional downstream tasks.

B. The Encoders

We choose FOCAL [24] as our self-supervised training framework to pre-train VibroFM . We train and test VibroFM with three different backbone encoders:

- **DeepSense** [28] is a DNN classifier designed for time-series sensory inputs. It applies convolution layers on modality spectrograms to extract general features and then utilizes recurrent layers (stacked GRU) to further extract global temporal relationships.
- **SWIN-Transformer (SW-T)** [29] is a variant of Vision Transformer (ViT) [32], proposing to extract a hierarchical representation through downsampling and shifting window operations.
- **TSMixer** [30] is a popular lightweight neural network for various industrial time-series forecasting tasks. It mainly leverages multi-layer perception (MLP) blocks on non-overlapping time series patches to learn multivariate and inherent temporal representations. The MLP layers (called Mixer layers) learn correlations across the patches, between the hidden feature within each patch, and between different channels.

IV. EVALUATION RESULTS

Below, we examine VibroFM performance in terms of robustness and label efficiency after fine-tuning with some target domain labeled data and then compare the training efficiency of the supervised and the foundation models.

A. Model Retraining/Fine-tuning

For purposes of comparing with supervised solutions, we divide the Evaluation Set into training, validation, and testing data with a ratio of 8:1:1. We train supervised models using different amounts of labeled samples (label ratio) from the training data of the Evaluation Set (100%, 50%, 10%, 1%) and use the same amount of data for fine-tuning VibroFM. We then evaluate their respective performance on the withheld testing data. Table III summarizes the performance of the retrained models on the Evaluation Set, under different label ratios. When the amount of labeled data used is high (100% or 50%), the supervised approaches work well. In fact, they slightly outperform VibroFM (that tunes its last layer only). However, as the amount of labeled data decreases (10% and 1%), the supervised approaches degrade substantially, whereas VibroFM suffers a much lower penalty in performance, suggesting a higher label efficiency.

B. Generalization to New Targets

Next, we show that VibroFM also generalizes well to unseen targets (absent in pre-training data). While Polaris, Warthog, and Civilian classes are present in MOD, the Husky class is not and is therefore not seen by VibroFM during pre-training. We analyze VibroFM's performance (fine-tuned with 100% label ratio and SW-T as the backbone encoder) for each class and show the confusion matrix in Figure 3.

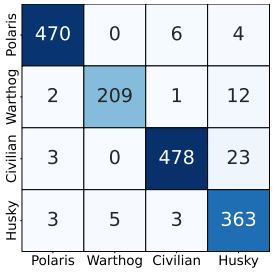
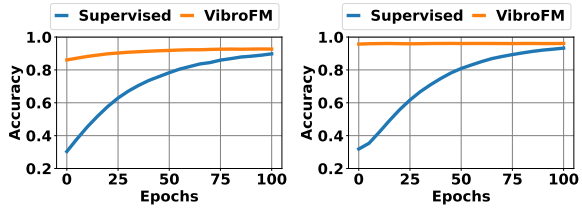


Fig. 3. Test confusion matrix of VibroFM with SW-T as the backbone fine-tuned on Evaluation Set.

Note that, VibroFM can correctly classify the Husky class even though it was not exposed to the Husky data during pre-training. This suggests that the foundation model is learning

TABLE III
FINE-TUNING RESULTS ON THE EVALUATION SET. MODELS ARE TRAINED/FINE-TUNED ON THE EVALUATION SET.

Label Ratio		100%		50%		10%		1%	
Encoder	Model	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DeepSense [28]	Supervised	**0.9684**	**0.9637**	**0.9425**	**0.9328**	0.8078	0.7714	0.5247	0.5019
	Supervised-fine-tuned	0.7933	0.7578	0.7762	0.7379	0.7383	0.6892	0.5974	0.5392
	VibroFM	0.9330	0.9293	0.9204	0.9154	**0.8976**	**0.8893**	**0.8078**	**0.7876**
SW-T [29]	Supervised	**0.9842**	**0.9840**	**0.9608**	**0.9589**	0.7434	0.7107	0.3660	0.2802
	Supervised-fine-tuned	0.6372	0.5829	0.6327	0.5778	0.6056	0.5592	0.5607	0.5037
	VibroFM	0.9526	0.9473	0.9558	0.9524	**0.9425**	**0.9372**	**0.8312**	**0.8176**
TSMixer [30]	Supervised	**0.9722**	**0.9700**	**0.9216**	**0.9117**	0.7124	0.6912	0.5556	0.4936
	Supervised-fine-tuned	0.7705	0.7430	0.7636	0.7356	0.7427	0.7151	0.6625	0.6245
	VibroFM	0.8382	0.8233	0.8363	0.8225	**0.8217**	**0.8067**	**0.7377**	**0.7158**



(a) DeepSense (b) SW-T

Fig. 4. Accuracy curves of supervised training and VibroFM fine-tuning.

a generalized representation of surrounding moving objects through the task-agnostic objectives during pre-training and can adapt well to new targets in downstream tasks.

C. Training Efficiency

In this section, we compare the training efficiency of the supervised models and the fine-tuning efficiency of VibroFM (which we refer to as "training" efficiency as well, for the sake of brevity, below). We define the training efficiency as the convergence speed or the number of training epochs needed for convergence. We compare the convergence speed of supervised training and fine-tuning by observing the training accuracy curves in Figure 4 during the first 100 epochs. On both backbone encoders, VibroFM (fine-tuning) converges much faster compared to the supervised model. This shows that the pre-trained representation is useful for the downstream task and can easily transfer knowledge to achieve high performance quickly. On the other hand, since the supervised models are trained from scratch, they begin at a lower accuracy and with more parameters to train. Thus, the supervised algorithm approaches VibroFM performance only towards the end of the 100 epochs. We do not consider the supervised-fine-tune benchmarks since they are dominated by the others.

D. Run-time Execution and On-device Inference

In this section, we deploy our models to a RaspberryShake 4D (RS4D), model 4B Rev 1.4. The RS4D device has 8GB of RAM on a Raspberry Pi single-board computer with an ARM Cortex-A72 processor. We first evaluate the computation overhead during inferencing, followed by an on-device training/fine-tuning analysis.

TABLE IV
COMPARING INFERENCE COMPUTATION OVERHEAD BETWEEN SUPERVISED AND VIBROFM ACROSS DIFFERENT MODELS.

Encoder	Model	Size (MB)	Parameters (M)	Infer Speed (s)
DeepSense	Supervised	13.787	3.6123	0.145332
	VibroFM	25.27	6.6220	0.101111
SW-T	Supervised	47.433	12.4342	0.190434
	VibroFM	44.955	11.7725	0.184065
TSMixer	Supervised	6.444	1.6892	0.075898
	VibroFM	7.463	1.9523	0.070949

We evaluate the trained and fine-tuned model on the RS4D Pi device. Detailed information regarding the sizes and number of parameters (expressed in millions) of these models is provided in Table IV.

The table also shows the average time taken by each model to make an inference from a single data sample (a two-second window of input sensor data). All models can compute an inference from the two-second sample in less than 0.2 seconds. SWIN-Transformer, due to its expensive attention operations, is the largest and the slowest. TSMixer, composed of mostly lightweight multi-layer perception layers, has the smallest size and quickest inference speed. Importantly, the foundation model based approach is similar in inference efficiency to supervised classifier.

E. On-device training/fine-tuning

Next, we train and fine-tune these models on the RS4D Pi device and profile their speed. Specifically, we record the average time required to process one batch during both training and fine-tuning phases across a range of batch sizes, from 1 to 128. The results, shown in Table V, reveal a significant gap between the Supervised models and VibroFM in terms of batch processing time. VibroFM, which only tunes its final linear layer, demonstrates a superior training speed — achieving a 50% and up to nearly 90% reduction in time compared to its Supervised counterparts. This efficiency makes VibroFM an ideal approach for on-device learning, particularly in scenarios requiring rapid adaptation to dynamic environments. Faster processing time can also significantly lower the energy consumption on incoming tasks, which is non-trivial for resource-constrained IoT devices. Besides, we use Pytorch Profiler [33] to analyze the memory consumption required to train and

TABLE V
AVERAGE BATCH PROCESSING TIME DURING TRAINING (SUPERVISED) AND FINE-TUNING (VIBROFM) ON RASPBERRY PI DEVICE.

Encoder	Model	Average Processing Time for each batch size (second)								Average Time Reduction
		1	2	4	8	16	32	64	128	
DeepSense	Supervised	0.6499	0.8850	1.2151	1.9488	3.4106	6.9621	13.3596	29.6567	-88.49%
	VibroFM	0.1052	0.1374	0.1724	0.2452	0.3481	0.5901	1.0795	2.0166	
SW-T	Supervised	1.2364	1.5483	2.2258	3.5723	6.1268	11.2664	21.5920	42.6260	-64.84%
	VibroFM	0.2639	0.4035	0.6932	1.2597	2.4553	4.5907	9.2683	18.6447	
TSMixer	Supervised	0.3526	0.5386	0.8981	1.5925	3.0116	5.8583	11.5925	24.8614	-54.94%
	VibroFM	0.1215	0.2092	0.3825	0.7470	1.4690	2.8076	5.6527	12.9797	

fine-tune the model. We present the peak memory allocation for various batch sizes in Table VI. In contrast to Supervised models that train the entire model, VibroFM requires only a minimal amount of memory to fine-tune its final linear layer. The benefit of the Foundation Model for runtime execution is two-fold: first, its memory usage is significantly lower than that of fully supervised models; second, it achieves a much faster fine-tuning speed than traditional supervised training.

F. Effect of Pre-Training Data Scale

Next, we examine VibroFM when additional data is used for pre-training. Specifically, we pre-train VibroFM at a larger scale on the VibroFM-Large dataset described in Section III-A. To maintain a model size suitable for edge device deployment, we keep the model architecture and parameters consistent with the previous experiments and only focus on increasing the scale of the pre-training dataset. We plot the accuracy of VibroFM-Large against VibroFM with DeepSense as the backbone encoder in Figure 5. VibroFM-Large, leveraging additional unlabeled data achieves better generalization performance than VibroFM.

G. Additional Downstream Tasks

We also explore an additional downstream task — distance classification, using discrete distance labels derived from the sensor and vehicle GPS value for each sample. Two domains are analyzed. Domain A contains data gathered under varying environmental conditions, exhibiting domain shift effects for both VibroFM and VibroFM-Large. Conversely, Domain B corresponds to the dataset exclusively seen by VibroFM-Large. We train and fine-tune the models using the distance labels and present the results in Figure 6. In Domain A, both VibroFM variants demonstrate resilience against data from a different domain, outperforming the Supervised models. Within Domain B, VibroFM-Large achieves the best performance, indicating that pre-trained knowledge significantly enhances the performance of downstream tasks within the seen domain. The results underscore that pre-trained embeddings capture comprehensive task-agnostic knowledge, making them suitable for deployment in a wide range of downstream tasks. Such adaptability is particularly useful in IoT applications, allowing for the seamless adaptation and transfer of a single foundation model, pre-trained with unlabeled data, to multiple tasks with minimal computation cost.

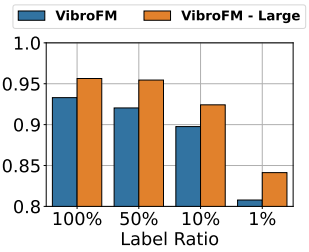


Fig. 5. Test accuracy of VibroFM and VibroFM-Large with DeepSense against different label ratios.

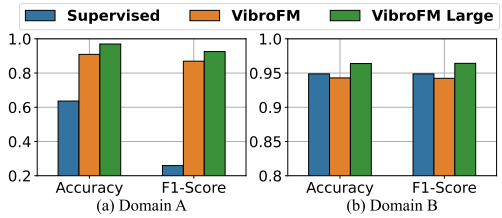


Fig. 6. Distance classification with DeepSense on two domains.

V. DISCUSSION

The results reported in this paper suggest that the task-agnostic nature of pre-training of self-supervised models endows them with greater robustness, making them ideally suited for IoT application deployment across various environments with only limited fine-tuning needed to achieve high-quality and efficient inference. Unlike traditional supervised models, these pre-trained models exploit *unlabeled* data, while offering enhanced resilience against domain shifts. The pre-trained models also show exceptional generalization abilities to unseen targets. These characteristics are particularly useful in dynamic IoT sensing scenarios where different sensor deployments (even within the same application) may be subjected to vastly different conditions. Importantly, the above beneficial properties are attained at moderate pre-training data sizes, thus supporting the case for μFMs.

The high label efficiency of pre-trained models further facilitates their rapid deployment to a wide array of downstream tasks, where label scarcity is a critical challenge. Merely training a single linear layer in VibroFM for fine-tuning can easily reach optimal performance within a few epochs. Besides, VibroFM has a relatively small memory requirement with a much higher throughput during fine-tuning, compared

TABLE VI
PEAK MEMORY ALLOCATION IN MB FOR DIFFERENT BATCH SIZES DURING TRAINING (SUPERVISED) AND FINE-TUNING (VIBROFM).

Encoder	Model	Peak Memory Consumption for each batch size (MB)							
		1	2	4	8	16	32	64	128
DeepSense	Supervised	33.4778	87.1689	98.5412	121.2858	166.7751	257.7537	439.7108	858.3438
	VibroFM	0.4977	0.5793	1.1293	2.2292	4.4311	8.8349	17.6424	35.2574
SW-T	Supervised	68.1482	89.0742	130.9264	214.6307	342.1466	675.292	1341.5829	2679.6642
	VibroFM	4.0575	7.1023	13.1918	25.3708	49.729	98.4452	194.865	389.73
TSMixer	Supervised	20.2296	34.0536	61.7017	116.9978	227.5901	448.7746	891.1436	1775.8816
	VibroFM	2.0768	4.1399	8.2661	16.5186	33.0234	66.0331	132.0525	264.0913

to fully-supervised learning. This efficiency not only enhances the practicality of μFMs in dynamic settings but also opens opportunities for on-device training, making it feasible to train them on resource-constrained IoT devices.

Finally, one should acknowledge that this initial study, while promising, offers only anecdotal evidence. More research is needed to experiment with other application domains, modalities, tasks, pre-training techniques, and conditions in terms of data volume. The authors hope that this initial work might encourage a broader and more systematic investigation into the μFM concept for smart and distributed sensing applications.

VI. RELATED WORK

Deep Learning has catalyzed significant advances in inference from IoT sensing data [34], with DNNs becoming integral to a wide range of IoT applications [35], [36]. However, domain-specific challenges still lead to many limitations in building robust DNNs for IoT sensing. Deployed DNNs must handle unpredictable interference in the field that greatly alters the statistical distribution of collected sensor data. The altered distribution, or *domain shift* [37], can significantly degrade DNN performance, leading to inaccurate results. FMs [1] have gained increasing popularity, most notably in language [2], [38], [39] and vision [3], [16]. Contrastive Learning (CL) [24], [40], [41] has been a popular form of learning to extract a robust embedding space during pre-training. The main idea is to pull similar samples closer while pushing other samples further apart in the embedding space. Unimodal CL frameworks like [40], [42] apply random augmentations to learn transformation invariant information. Multi-modal CL frameworks [41], [43] enforce cross-modal consistency. Improving resilience against domain shifts has been widely studied in recent years [44], [45], improving the efficiency of unsupervised domain adaptation for IoT applications. These solutions primarily consider classifiers trained in a supervised manner. Others have also worked on Federated Learning-based domain generalization [46], [47]. Numerous works analyze domain generalization in vision [48], but less has been explored for IoT applications.

VII. CONCLUSIONS

In this paper, we advertise the importance of μFMs, exemplified by a vibration-based Foundation Model the authors pre-trained using a self-supervised learning framework, FOCAL, comparing against conventional supervised models in the context of IoT sensing. Model evaluation has demonstrated that it requires minimal domain-specific tuning to achieve significantly improved robustness and generalization, compared to fully-supervised models. The μFM was further shown to allow efficient real-time inference and fine-tuning on resource-constrained IoT devices. Our results highlight promising opportunities for μFMs in the IoT landscape. Future work will focus on developing more capable μFMs for a broader class of IoT systems, sensing modalities, and applications.

VIII. ACKNOWLEDGEMENTS

Research reported in this paper was sponsored in part by DEVCOM ARL under Cooperative Agreement W911NF-17-2-0196 (ARL IoBT CRA), and in part by NSF CNS 20-38817, and the Boeing Company. It was also supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The views and conclusions contained in this document are those of the authors, not the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[4] T. Kimura, J. Li, T. Wang, D. Kara, Y. Chen, Y. Hu, R. Wang, M. Wigness, S. Liu, M. Srivastava, S. Diggavi, and T. Abdelzaher, "On the efficiency and robustness of vibration-based foundation models for iot sensing: A case study," in *Proceedings of the International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things (FMSys)*, ser. FMSys '24, 2024.

[5] T. Wang, D. Kara, J. Li, S. Liu, T. Abdelzaher, and B. Jalaian, "The methodological pitfall of dataset-driven research on deep learning: An iot example," in *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*. IEEE, 2022, pp. 1082–1087.

[6] J. G. Almaraz-Rivera, J. A. Cantoral-Ceballos, and J. F. Botero, "Enhancing iot network security: Unveiling the power of self-supervised learning against ddos attacks," *Sensors*, vol. 23, no. 21, p. 8701, 2023.

[7] Z. Zhang, S. Bu, Y. Zhang, and Z. Han, "Market-level integrated detection against cyber attacks in real-time market operations by self-supervised learning," *IEEE Transactions on Smart Grid*, 2024.

[8] Z. Wang, Z. Li, J. Wang, and D. Li, "Network intrusion detection model based on improved byol self-supervised learning," *Security and Communication Networks*, vol. 2021, pp. 1–23, 2021.

[9] S. Zhang, O. T. Ajayi, and Y. Cheng, "A self-supervised learning approach for accelerating wireless network optimization," *IEEE Transactions on Vehicular Technology*, 2023.

[10] M. S. Towhid and N. Shahriar, "Encrypted network traffic classification using self-supervised learning," in *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*. IEEE, 2022, pp. 366–374.

[11] G. Mai, W. Huang, J. Sun, S. Song, D. Mishra, N. Liu, S. Gao, T. Liu, G. Cong, Y. Hu *et al.*, "On the opportunities and challenges of foundation models for geospatial artificial intelligence," *arXiv preprint arXiv:2304.06798*, 2023.

[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[13] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained iot nodes: Opportunities and challenges," *IEEE Design & Test*, vol. 36, no. 2, pp. 7–40, 2019.

[14] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu, L. Su, and T. Abdelzaher, "Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 278–291.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[17] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

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

[19] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in neural information processing systems*, vol. 33, pp. 5812–5823, 2020.

[20] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 8765–8775, 2020.

[21] D. Liu, T. Wang, S. Liu, R. Wang, S. Yao, and T. Abdelzaher, "Contrastive self-supervised representation learning for sensing signals from the time-frequency perspective," in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–10.

[22] S. Yao, A. Piao, W. Jiang, Y. Zhao, H. Shao, S. Liu, D. Liu, J. Li, T. Wang, S. Hu *et al.*, "Stfnets: Learning sensing signals from the time-frequency perspective with short-time fourier neural networks," in *The World Wide Web Conference*, 2019, pp. 2192–2202.

[23] D. Kara, T. Kimura, S. Liu, J. Li, D. Liu, T. Wang, R. Wang, Y. Chen, Y. Hu, and T. Abdelzaher, "Freqmae: Frequency-aware masked autoencoder for multi-modal iot sensing," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 2795–2806.

[24] S. Liu, T. Kimura, D. Liu, R. Wang, J. Li, S. Diggavi, M. Srivastava, and T. Abdelzaher, "Focal: Contrastive learning for multimodal time-series sensing signals in factorized orthogonal latent space," in *Advances in Neural Information Processing Systems*, 2023.

[25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[26] ——, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2016.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

[28] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *International Conference on World Wide Web*, 2017.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021.

[30] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '23, 2023, p. 459–469. [Online]. Available: https://doi.org/10.1145/3580305.3599533

[31] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[33] "Pytorch profiler," https://pytorch.org/docs/stable/profiler.html.

[34] M. Srivatsa, T. Abdelzaher, and T. He, Eds., *Artificial Intelligence for Edge Computing*. Springer, 2023.

[35] B. Salehi, G. Reus-Muns, D. Roy, Z. Wang, T. Jian, J. Dy, S. Ioannidis, and K. Chowdhury, "Deep learning on multimodal sensor data at the wireless edge for vehicular network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7639–7655, 2022.

[36] V. Radu, C. Tong, S. Bhattacharya, N. D. Lane, C. Mascolo, M. K. Marina, and F. Kawsar, "Multimodal deep learning for activity and context recognition," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 1, no. 4, pp. 1–27, 2018.

[37] A. Mathur, T. Zhang, S. Bhattacharya, P. Velickovic, L. Joffe, N. D. Lane, F. Kawsar, and P. Lió, "Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2018, pp. 200–211.

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[39] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[40] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021.

[41] X. Ouyang, X. Shuai, J. Zhou, I. W. Shi, Z. Xie, G. Xing, and J. Huang, "Cosmo: Contrastive fusion learning with small data for multimodal human activity recognition," in *International Conference on Mobile Computing And Networking (MobiCom)*, 2022.

[42] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020.

[43] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *European Conference on Computer Vision (ECCV)*, 2020.

[44] J. Li, M. Jing, H. Su, K. Lu, L. Zhu, and H. T. Shen, "Faster domain adaptation networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5770–5783, 2021.

[45] Y. Zhao, D. Saxena, and J. Cao, "Memory-efficient domain incremental learning for internet of things," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 1175–1181.

[46] L. Zhang, X. Lei, Y. Shi, H. Huang, and C. Chen, "Federated learning for iot devices with domain generalization," *IEEE Internet of Things Journal*, 2023.

[47] Y. Huang, M. Du, H. Zheng, and X. Feng, "Incremental unsupervised adversarial domain adaptation for federated learning in iot networks," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2022, pp. 186–190.

[48] I. Achituve, H. Maron, and G. Chechik, "Self-supervised learning for domain adaptation on point clouds," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 123–133.