CoRAST: Towards Foundation Model-Powered Correlated Data Analysis in Resource-Constrained CPS and IoT

Yi Hu*, Jinhang Zuo†, Alanis Zhao*, Bob Iannucci‡ and Carlee Joe-Wong*
*Carnegie Mellon University, †Caltech & UMass Amherst, ‡Google
{yihu,aazhao,cjoewong}@andrew.cmu.edu, jhzuo@cs.umass.edu, biannucci@google.com

***Abstract*—Foundation models (FMs) emerge as a promising solution to harness distributed and diverse environmental data by leveraging prior knowledge to understand the complicated temporal and spatial correlations within heterogeneous datasets. Unlike distributed learning frameworks such as federated learning, which often struggle with multimodal data, FMs can transform diverse inputs into embeddings. This process facilitates the integration of information from various modalities and the application of prior learning to new domains. However, deploying FMs in resource-constrained edge systems poses significant challenges. To this end, we introduce CoRAST, a novel learning framework that utilizes FMs for enhanced analysis of distributed, correlated heterogeneous data. Utilizing a server-based FM, CoRAST exploits existing environment information to extract temporal, and cross-feature correlations among sensor data. This enables CoRAST to offer context-aware insights for localized client tasks through FM-powered global representation learning. Our evaluation on real-world weather dataset demonstrates CoRAST's ability to exploit correlated heterogeneous data through environmental representation learning to reduce the forecast errors by up to 50.3% compared to the baselines.**

***Index Terms*—Cyber-physical systems, foundation models, heterogeneous data analysis, Internet of Things (IoT), time series.**

I. Introduction

The data generated by edge devices in Cyber-Physical Systems (CPS) and the Internet of Things (IoT) is inherently rich in spatial and temporal correlations, offering significant opportunities for enhancing edge intelligence. Those systems' distributed computing and sensing capabilities enable the monitoring of a physical environment through different modalities (e.g., audio/video signals, or environmental metrics like temperature and air pressure). Effectively utilizing this rich and interrelated data pool, collected from a diverse set of edge sensors, can substantially benefit a wide range of complex downstream tasks that require detailed data interpretation (Figure 1). For example, augmenting audio data with visual cues (e.g., lip-reading) from a camera focusing on a speaker can significantly improve the accuracy of speech recognition [1].

However, it is challenging to leverage this rich, distributed data effectively. Extensive research has been done on multivariate time series for forecasting or classification, and the major approach involves treating each variable as an independent univariate time series, addressing temporal and inter-variable correlations separately [2]–[4]. There is a lack of a unified strategy for integrating heterogeneous data types, such as video streams and time series data. Moreover, the prevalent practice of centralizing data processing can overlook essential spatial details. For instance, when one sensor captures 1D acceleration data at a particular location, while another records 3D gyroscope data elsewhere, the central system may fail to fully leverage the spatial context of these data points. This limitation underscores the need for strategies that can effectively integrate and interpret heterogeneous data sources while preserving and utilizing their spatial characteristics to enhance analysis and decision-making.

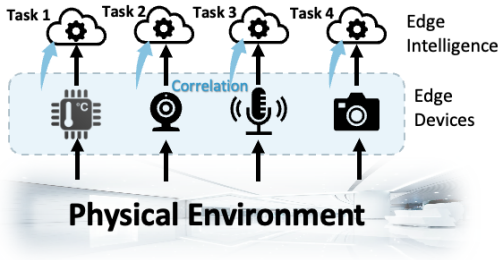

Fig. 1. Correlated Data in CPS and IoT

Foundation models (FMs) emerge a promising solution to complex data analysis and information fusion. Research in this domain is rapidly gaining popularity, as evidenced by recent surveys [5]–[8]. Employing transformers, these models adeptly convert diverse inputs into embeddings, enabling the *integration of various data types and facilitating the application of prior learning or training to new domains*. Large Language Models (LLMs) like GPT and BERT extend well beyond their initial applications in text and image processing to encompass temporal and multimodal data analysis. Research by Zhou et al. [9] exemplifies how pre-trained LLMs can match or even outperform the state-of-the-art methods in time series analysis. Advanced multimodal models [10], [11] further integrate multimodal data – from images and text to audio and video – for a diverse range of downstream tasks.

Nonetheless, integrating FMs into CPS and IoT applications poses major **challenges**, primarily due to the extensive

This work was funded by the NSF grant CNS-2106891 and the Wei Shen and Xuehong Zhang Presidential Fellowship at Carnegie Mellon University.

resources required to run these large models, often beyond the capabilities of edge devices. Many CPS and IoT applications require edge devices themselves to make decisions or predictions. For instance, in smart agriculture, soil moisture sensors may need to determine optimal irrigation levels based on collected data. These devices, e.g., low-power sensors, may not have the communication capabilities needed to continually send collected data to a central location and receive back a decision. Moreover, sending some data, e.g., audiovisual data, to a central location for processing may introduce privacy issues. We aim to answer the question: *how can we leverage the knowledge from a foundation model, previously trained on historical data, for distributed systems with resource-constrained edge devices?* Research challenges include:

Resource Constraints on Edge Devices. Edge devices or sensors within CPS or IoT environments often face severe memory and computational limitations, making it difficult to host FMs directly. For example, the base BERT model [12] has 108 million parameters. Training even moderately sized FMs typically requires several gigabytes of memory, presenting a challenge for updating these models on edge devices. This is true even when considering distilled or compressed versions of FMs designed to reduce size and resource requirements [13]. High-end edge computing devices, such as NVIDIA Jetsons, might struggle with the storage and memory demands required for training large models, let alone less capable IoT devices (e.g., microcontrollers) that are typically equipped with only a few megabytes of SRAM. Thus, involving FMs in training on locally collected environmental data may require significant involvement of more well-resourced servers that can host FMs.

Decentralized and Correlated Local Tasks. CPS and IoT applications often need distributed decision-making on the edge due to privacy concerns, real-time responsiveness, and unreliable network connections. This distributed intelligence calls for the development of local models tailored to the specific edge data models and distributions. Traditional federated learning (FL), even with personalization, often falls short when it comes to learning with correlated and heterogeneous local data. For example, adjacent edge devices with different sensor arrays may collect correlated yet distinct types of data, calling for unique local models for tasks like anomaly detection. This level of data specificity and the need for model customization go beyond what personalized federated learning currently offers. In light of the fact that many edge devices cannot directly host FMs, we must design new methods to integrate information from FMs hosted at a central server into distributed decision-making on such correlated data, without leading to excessive communication overhead.

Contributions: To address these research challenges, we introduce **CoRAST**, a novel general learning framework designed to utilize FMs for **A**nalyzing **S**patially and **T**emporally **CoR**related heterogeneous data within resource-constrained edge computing systems. Diverging from traditional approaches based off improving a single application objective, CoRAST adopts a holistic strategy, emphasizing the interconnectedness of data from a shared physical environment, thereby improving the learning for various downstream tasks. Leveraging FMs to understand the correlation among various sensor data and tasks with prior environmental knowledge, CoRAST provides context-aware insights for client-specific tasks. We summarize our contributions:

- We introduce CoRAST, the first FM-based learning framework for analyzing correlated heterogeneous data in CPS and IoT. The framework leverage a server-based FM to synthesize global insights and client-specific models that apply these insights to local tasks.
- Our study offers insights into designing FM-based architectures for managing correlated heterogeneous data.
- CoRAST improves distributed learning on a real-world weather dataset, reducing forecasting errors with its FM-based global learning approach.

We review related work in Section II and present CoRAST in Section III. The proposed framework is evaluated in Section IV. The paper concludes in Section V.

II. Related Work

FM for Correlated and Multimodal Data Analysis: Research on FM for correlated data analysis is rapidly advancing with models such as GPT4TS [9], LLM4TS [14], and AnomalyTrans [15], which utilize pre-trained models and transformers for tasks like time series analysis, forecasting, and anomaly detection. These models enhance their performance through targeted fine-tuning. GTA [4] employing graph structures for uncovering hidden data associations with novel attention mechanisms. Penetrative AI [16] exploits textualized expert knowledge to guide the reasoning of the physical world using existing language models, but it has limitations in detailed data analysis and comprehensive environment learning. Additionally, the adoption of multimodal large models, including CLIP [11] and Macaw-LLM [10], demonstrates FMs' potential to unify data modalities such as in text-image translation by converting diverse inputs into embeddings.

Advancements in Representation Learning: TS2Vec[17] introduces a universal contrastive learning approach to time series representations that effectively differentiates between positive and negative samples at various time scales with contextual information. In federated learning, CreamFL [18] pushes the boundaries of federated multimodal learning by sharing learned modality-specific representations. FedGKT [19] explores a novel strategy for knowledge transfer using local representation and global logits. In contrast, CoRAST explicitly incorporates FMs to learn representations capturing correlations between distributed and multi-modal data.

III. CoRAST

CoRAST leverages a server-based FM backbone to exploit both existing data and prior knowledge of the physical environment. It aims to learn useful representations of the environment in a self-supervised manner. These representations learned by CoRAST serve to provide valuable context for CPS and IoT applications operating within the same physical setting. On the client side, CoRAST integrates this environmental

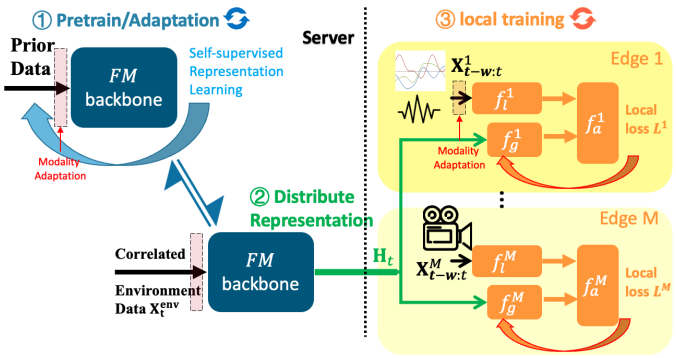

Fig. 2. CoRAST framework. ① Global Representation Learning – A server-based FM is pre-trained and fine-tuned on historical environment data for global representation learning in a self-supervised manner. ② Representation Distribution – Using the environment data that correlate with client local data, the FM produces and distributes contextual representations to edge clients aiding downstream local tasks. ③ Local Learning with Global Context – Clients integrate global contexts with their local data for independent local learning, leveraging broad environmental insights with local datasets.

understanding, as learned from the server FM, with local data to address specific edge computing tasks.

Figure 2 provides an overview of the framework. We consider a distributed system of M edge clients indexed by m, each equipped with various sensors for distinct local tasks. Each client m maintains a dataset $\mathbf{X}_t^m$ and aims to optimize a local model to minimize its unique loss function L_t^m, which may reflect its specific local task (e.g., predicting various environmental phenomena).

An edge server with the sufficient resources to run an FM, holds a dataset $\mathbf{X}_t^{\text{env}}$ correlating with the clients' local data. The FM, trained in a self-supervised or unsupervised way, generates global representations $\mathbf{H}_t$ of the environment to be distributed to the edge clients. This paper primarily focuses on the genral framework's design, leaving the detailed FM architecture and its adaptation to specific systems, applications, and data types for future exploration. Our framework is agnostic to the specific architecture of this FM.

Upon receiving the global representation $\mathbf{H}_t$ from the server, edge clients merge this information with their local data to refine their models. To illustrate the idea, consider a scenario where the environment of interest is a single room monitored by various sensors with distinct functions (e.g., an infrared camera that detects human at the front door, a microphone designed to pick up specific keywords, and temperature sensors programmed to regulate the room's air conditioning). The server might have access to the video streams from a surveillance camera or an aggregation of sensor data to train the FM. The server can pre-train the FM with all the existing environmental data and generate useful environment representations $\mathbf{H}_t$ to each client. This representation embeds vital extraneous global information not directly observable by local sensors (e.g., the presence of a person in the room), but crucial for enhancing local task performance.

By training the local model on both the specific local data and the enriched global representation, the model is equipped with a broader understanding of the environment. This comprehensive input significantly elevates the model's potential to excel in its designated task.

A. Server-Based Representation Learning

The goal of the server-based representation learning is to encode the intricate relationships and correlations among multimodal data gathered from a specific environment. Correlation describes how data relate to each other. If two things are independent, they are not correlated. In the realm of information theory, discerning the correlational dynamics within data can effectively diminish system entropy – a metric quantifying uncertainty. Should two variables X and Y exhibit correlation, their joint entropy $H(X,Y)$ would be less than the sum of their separate entropies $H(X) + H(Y)$ [20]. This implies that correlated data inherently reduce system uncertainty, potentially enhancing model performance when such correlations are adeptly leveraged.

Modeling correlations has been challenging. Traditional methods use meta-data, such as spatial and temporal information, or entropy analysis to discern correlations [21]. However, they fall short in complex scenarios where the correlation changes over time, or when dealing with high-dimensional, multimodal data. In recent years, the shift towards deep learning approaches, particularly the use of large foundation models, enable autonomously derive insights from extensive datasets and learn valuable embeddings for multimodal data.

In CoRAST, we employ a FM backbone for representation learning. This learning process is conducted in a self-supervised manner, aiming to extract correlation-centric, high-level concepts that help improve performance across a range of downstream tasks. One notable strategy is contrastive representation learning, which improves the model by differentiating among varied data samples [22]. TS2Vec[17], for example, applies this method to time series data, capturing contextual information across various temporal scales effectively.

Yet, correlations in CoRAST's setting appear as temporal (time-based changes), spatial (spatial distribution), and cross-modal (across different data types). Addressing these correlations effectively demands careful architectural design. We briefly discuss two advanced techniques that enhance the ability to interpret complex system inputs and diverse data types and thus handle more complex correlations.

a) Graph Neural Networks (GNNs): GNNs stand out for their ability to model relational data. In GNNs, node representations are iteratively updated through a series of message passing with neighbours [23]. The graph structures capture complex relationships within data, such as inter-temporal dependencies–where nodes represent distinct time points, or inter-variable connections–where nodes represent different variables [2]. Edges thus represent temporal and inter-variable correlations. For datasets of heterogeneous data types, heterogeneous GNNs is a viable solution that supports processing input features of varying sizes.

b) Attention Mechanism: Given a task, an attention score is computed, indicating the relevance between pairs of inputs

and generates embeddings based on that score [24]. In our context, this process enables models to interpret the interconnections between different data features and their temporal connections over time. Transformers heavily rely on attention to encode input sequences. When integrated with GNNs, it leads to the Graph Attention Networks (GAT) [25] that can discern the complex interrelationships present within graph-structured data, enabling the extraction of meaningful representations out of spatially and temporally correlated data.

B. Client Local Training

Each client m's local model contains three parts: a local data processing module f_l^m, a global information extraction module f_g^m that deciphers the encoded global representation $\mathbf{H}$, and an affiliate function f_a^m that integrates outputs from both modules to produce the final outputs $\mathbf{Y}^m$. Mathematically, $\mathbf{Y}^m = f_a^m\left(f_l^m(\mathbf{X}^m), f_g^m(\mathbf{H})\right)$.

The local loss L^m can be determined by the specific local tasks of each client, which might include, for instance, the use of cross-entropy loss $L_{CE}(\mathbf{Y}^m, y^m)$ for classification tasks, where y^m represents the target label; or the prediction error $L_{MSE}(\mathbf{Y}^m, \mathbf{X}_f^m)$ on the discrepancy between the predicted and actual future values of the local measurements $\mathbf{X}_f^m$.

In addition to flexibly supporting a wide range of local tasks, the client local model also allows multimodal data inputs that may exhibit temporal or cross-modal correlations. The data processing unit can be customized to capture time-dependent patterns or inter-feature correlations. For instance, temporal convolutional layers [26] can be used to grasp the temporal dynamics of the data in CPS and IoT systems, and attention mechanisms with graph structures can be adopted to learn the interconnections between two types of data.

C. Continual Learning

The CoRAST framework is designed to foster continual learning, allowing for the adaptation and incorporation of new data over time. At its core, the framework maintains a separation between the process of learning environmental representations and the specific client local tasks. This division allows environmental representation learning to universally augment both present and future CPS and IoT tasks within a given environment. Consequently, both the server model and the local client models can efficiently integrate updates or new data without the need for a complete system-wide retraining.

The framework allows different update frequencies for the server model and the client models, a practical feature due to the usual delay in client data becoming available to the server. One learning scheme is to have different update intervals, T_s for the server model and T_c for the client models, with T_s being significantly longer than T_c. We don't focus on the specifics of how the model adjusts to new data. In the intervals between server model updates, clients can manage incoming new data using the representations generated from previously learned server model. Clients are required to seek updated global representations from the server only when the server model is updated, an event that occurs much less frequently.

TABLE I
EXPERIMENT SETTINGS

Settings	Client Data	Server Data
1-centralized	[Tdew, rh, sh]	[Tdew, rh, sh]
1-distributed	[Tdew], [rh], [sh]	[Tdew, rh, sh]
2-centralized	[Tdew, Tpot, rh, p, sh]	[Tdew, Tpot, rh, p, sh]
2-distributed	[Tdew, Tpot], [rh, p], [sh]	[Tdew, Tpot, rh, p, sh]

TABLE II
MODEL SETUP

Model		# of parameters
Client (TCN)	1-centralized	13984
	2-centralized	15214
	1-distributed	12970, 12970, 12970
	2-distributed	13450, 13450, 12970
Server (TS2Vec)		337152
Representation		256

D. Resource Usage

CoRAST optimizes resource usage by enabling distributed clients to leverage the knowledge of a pre-trained FM without local execution of the large model. Instead, clients only need to run moderately-sized local models, ensuring a lightweight computational footprint. Communication overhead predominantly involves one-directional data flow from the server to the clients, mainly for the purpose of transmitting representations. Currently, we consider a single global representation for all the clients, with the representation's size dependent on the volume and variety of data available to the server.

For training with new data spanning a length of T, the global representation is a $d \times T$ matrix for a d-dimensional representation model. This representation is distributed to all clients at intervals of T_s, the update interval of the server model. During inference, a d-dimensional vector representation is broadcast to all clients at each inference point.

A future direction would involve developing customized representations tailored to individual client needs, thereby encoding only the pertinent partial information. This would further optimize computational and communication efficiency, for instance, by utilizing smaller representations for simple tasks like automatically adjusting air conditioning based on temperature. More discussion in Section V.

IV. EXPERIMENTS

We evaluate our CoRAST distributed learning framework for managing complex correlated heterogeneous data distribution in edge computing systems. We use the real-world benchmark weather dataset from [27] and distribute the dataset's measurement data across clients. The dataset includes a variety of measurements such as dew temperature (Tdew), potential temperature (Tpot), air pressure (p), and relative humidity (rh), among others. It is characterized by temporal patterns, such as variations in temperature and humidity throughout the day, as well as inter-feature correlations, such as the relationship between the pressure and temperature.

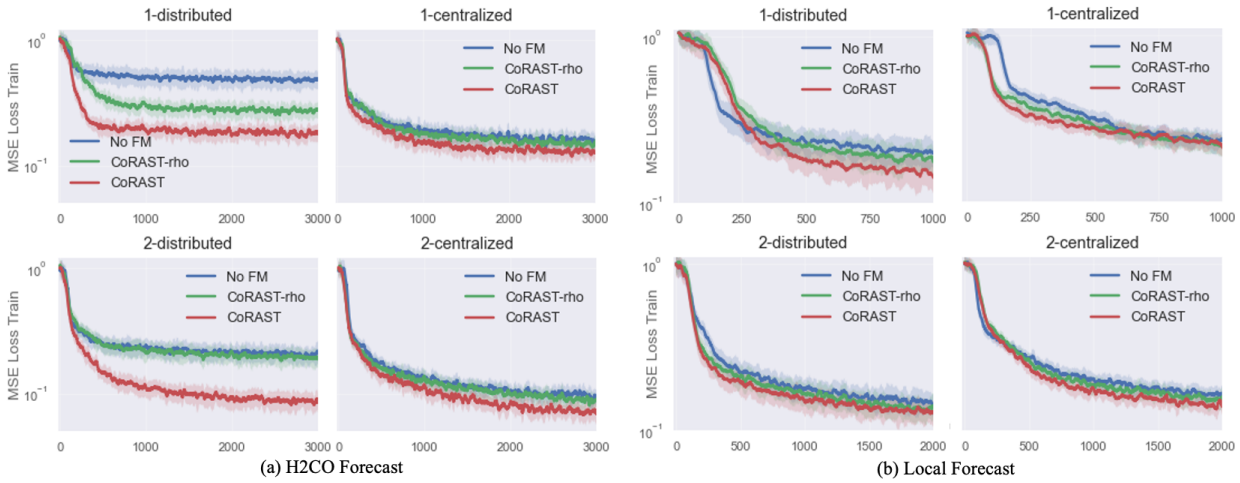

Fig. 3. MSE Train Loss with the standard deviation represented by a shaded area around the loss curves. CoRAST has the lowest training loss in all settings when clients share the same local training task or have distinct local training tasks.

To demonstrate CoRAST's ability to process correlated heterogeneous data, each client is assigned a distinct subset of data types without any overlap or data sharing. The specific experimental settings are detailed in Table I, where each set of square brackets denotes the data types assigned to a particular client. There are two settings: Setting 1 (centralized/distributed) and Setting 2 (centralized/distributed). Setting 2 provides clients with additional climate information (i.e., Tpot and p) compared to Setting 1.

In distributed settings, three clients each gather non-overlapping sets of climate measurements, while in centralized settings, one central client collects all data. Given that clients collect different types of data with possible dimension differences (Setting 2), traditional FL struggles due to the local models' diversity, making it hard to aggregate these into an effective global model with potentially different architectures.

For all experimental setups, the server accesses all the client data to train the FM for representation learning. In the centralized cases, the learning process is equivalent to training a lightweight local model with a FM-based learning approach pre-trained on the same data. We explore two additional scenarios: (1) **No FM**, where FM is not utilized, and learning is purely local, akin to traditional multi-variate time-series analysis. (2) **CoRAST-rho**, where the server uses only the density measure `rho` – a variable not directly observed but significantly correlated with client data – to evalute the framework without direct data sharing from clients.

Setup: The dataset is divided into training, validation, and test sets in a 7:1:2 ratio. The server's backbone model employs TS2Vec [17], consisting of 3 contrasting layers, each containing 64 hidden units. It is pre-trained on the training set using a self-supervised approach with hierarchical contrastive loss and finally generates a 256-dimensional global representation for each time step.

On the client side, local data is divided into sequences of 128 and processed through a 3-layer temporal convolutional network (TCN) [26], with a kernal size of 3 and 64 hidden units. Meanwhile, a fully connected network is applied to the server-derived representation corresponding to the same time step. These outputs are combined and fed into an affiliated output layer to produce final results. Table II summarizes the number of parameters of the client models, the server model, and the dimension of the representation used in the experiment.

The server model trains at an initial learning rate of 0.001, and client models at 0.0001, using the Adam optimizer and Cosine Annealing scheduler. Training halts if performance drops on the validation set for three consecutive evaluations (i.e., an early stopping criterion with a patience of 3).

a) Aligned Objective - H2CO Forecast: We first focus on a common objective across all clients: predicting future H2CO levels, though none directly observe H2CO. This variable is highly correlated with client data. Training losses alongside test errors, are depicted in Figure 3(a) and summarized in Table III. CoRAST enhances learning outcomes across all scenarios. When focusing on a singular global objective, CoRAST significantly improves distributed learning by adding additional environmental insights into learned representations. This approach facilitates a more effective aggregation of client data than traditional forecasting techniques, as evidenced by Table III's reductions in forecast error in comparison to the centralized case lacking FM integration. Furthermore, the improvements of CoRAST-rho over no FM show the value of the global representation in conveying critical environmental information, previously inaccessible to client-based learning, thereby improving local model performance. Setting 2 generally gives lower MSE than Setting 1, indicating the value of including more weather data in the prediction.

b) Multimodal Tasks - Local Forecast: We explore multimodal local objectives with clients predicting future values of their specific observed variables, which vary in type. For example, in the 2-distributed setting, three clients forecast variables [Tdew, Tpot], [rh, p], and [sh], respectively. As a result, distinct local models targeting specific client data need to be learned. Figure 3(b) and Table IV show the average

TABLE III
H2CO FORECAST - TEST MSE. CORAST HAS THE LOWEST TEST LOSS IN MOST SETTINGS WHEN ALL CLIENTS HAVE THE SAME LOCAL TASK. CORAST-RHO TAKES ADVANTAGE OF LIMITED GLOBAL INFORMATION TO OUTPERFORM THE BASELINE WITHOUT A FM.

Setting	No FM	CoRAST-rho	CoRAST
1-centralized	0.195	0.182	**0.171**
1-distributed	0.391	0.303	**0.201**
2-centralized	0.075	**0.055**	0.061
2-distributed	0.151	0.146	**0.063**

TABLE IV
LOCAL FORECAST - TEST MSE IN DISTRIBUTED SETTINGS WITH DISTINCT LOCAL TASKS FOR CLIENTS. CORAST HAS THE LOWEST TEST LOSS IN MOST SETTINGS. CORAST-RHO TAKES ADVANTAGE OF LIMITED GLOBAL INFORMATION TO OUTPERFORM THE BASELINE WITHOUT A FM.

Setting	Variable	No FM	CoRAST-rho	CoRAST
1-distributed	Tdew	0.072	**0.056**	0.075
	rh	0.297	0.318	**0.276**
	sh	0.077	0.072	**0.059**
2-distributed	Tdew	1.953	**1.796**	1.853
	rh	0.223	0.216	**0.211**
	sh	1.945	**1.912**	1.930
	p	0.151	0.159	**0.145**
	Tpot	2.633	2.544	**2.438**

training loss across all variables and test MSE for each variable. The results show that CoRAST improves the training process and model accuracy in both centralized and distributed cases, which highlights CoRAST's capability to enhance the distributed learning of interrelated tasks by utilizing correlated environmental data through FMs. Comparing CoRAST and CoRAST-rho reveals the distinct influences that environmental variables exert on each other, emphasizing the complexity of correlations within environmental data.

V. CONCLUSION AND FUTURE WORK

This paper introduces CoRAST, a general FM-based learning framework specifically designed for analyzing correlated heterogeneous data in CPS and IoT. CoRAST uses a server-based FM to generate global insights through advanced representation learning and enables client models to enhance task performance. It supports asynchronous updates for continuous learning without requiring system-level retraining, and is flexible in managing diverse local data and customizing client models. Our evaluation demonstrates CoRAST improves distributed environmental forecasting on a real-world dataset.

Future work will focus on refining the architecture to better support temporal, spatial, and cross-modal correlations. This includes adapting to diverse training datasets and the discrepancy in data available at training versus inference. Moreover, the current method of using a uniform global representation fails to meet the diverse needs of individual clients. Introducing client-specific adaptors for tailored representations adds complexity but is essential for personalized learning. These adaptors, reliant on client feedback for performance optimization, introduce a new interaction dynamic.

REFERENCES

[1] D. Ryumin, D. Ivanko, and E. Ryumina, "Audio-visual speech and gesture recognition by sensors of mobile devices," *Sensors*, 2023.

[2] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," in *2020 IEEE Intl. Conf. on Data Mining (ICDM)*, 2020.

[3] Z. Z. Darban, G. I. Webb, S. Pan, C. C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," 2022.

[4] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in iot," *IEEE Internet of Things Journal*, 2021.

[5] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, 2023.

[6] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," 2024.

[7] M. Jin, Q. Wen, Y. Liang, C. Zhang, S. Xue, X. Wang, J. Zhang, Y. Wang, H. Chen, X. Li, S. Pan, V. S. Tseng, Y. Zheng, L. Chen, and H. Xiong, "Large models for time series and spatio-temporal data: A survey and outlook," 2023.

[8] J. Wu, W. Gan, Z. Chen, S. Wan, and S. Y. Philip, "Multimodal large language models: A survey," in *2023 IEEE BigData*, 2023.

[9] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One Fits All: Power general time series analysis by pretrained lm," in *NeurIPS*, 2023.

[10] C. Lyu, M. Wu, L. Wang, X. Huang, B. Liu, Z. Du, S. Shi, and Z. Tu, "Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration," 2023.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Intl. Conf. on machine learning*, 2021.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[13] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[14] C. Chang, W.-Y. Wang, W.-C. Peng, and T.-F. Chen, "LLM4TS: Aligning pre-trained llms as data-efficient time-series forecasters," 2024.

[15] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," 2023.

[16] H. Xu, L. Han, Q. Yang, M. Li, and M. Srivastava, "Penetrative ai: Making llms comprehend the physical world," in *Proceedings of the 25th Intl. Workshop on Mobile Computing Systems and Applications*, 2024.

[17] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *Proceedings of the AAAI Conf. on Artificial Intelligence*, 2022.

[18] Q. Yu, Y. Liu, Y. Wang, K. Xu, and J. Liu, "Multimodal federated learning via contrastive representation ensemble," 2023.

[19] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," *NeurIPS*, 2020.

[20] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, 1948.

[21] N. Nguyen Thi Thanh, K. Nguyen Kim, S. Ngo Hong, and T. Ngo Lam, "Entropy correlation and its impacts on data aggregation in a wireless sensor network," *Sensors*, 2018.

[22] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *IEEE Access*, 2020.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[25] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, 2017.

[26] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018.

[27] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," 2023.