Energy-Efficient Collaborative Sensing: Learning the Latent Correlations of Heterogeneous Sensors

YUNJI LIANG, School of Computer Science, Northwestern Polytechnical University
XIN WANG, School of Software, Northwestern Polytechnical University
ZHIWEN YU, School of Computer Science, Northwestern Polytechnical University
BIN GUO, School of Computer Science, Northwestern Polytechnical University
XIAOLONG ZHENG, Institute of Automation Chinese Academy of Sciences
SAGAR SAMTANI, Kelley School of Business, Indiana University

With the proliferation of Internet of Things (IoT) devices in the consumer market, the unprecedented sensing capability of IoT devices makes it possible to develop advanced sensing and complex inference tasks by leveraging heterogeneous sensors embedded in IoT devices. However, the limited power supply and the restricted computation capability make it challenging to conduct seamless sensing and continuous inference tasks on resource-constrained devices. How to conduct energy-efficient sensing and perform rich-sensor inference tasks on IoT devices is crucial for the success of IoT applications. Therefore, we propose a novel energy-efficient collaborative sensing framework to optimize the energy consumption of IoT devices. Specifically, we explore the latent correlations among heterogeneous sensors via an attention mechanism in temporal convolutional network (TCN) to quantify the dependency among sensors, and characterize the heterogeneous sensors in terms of energy consumption to categorize them into low-power sensors and energy-intensive sensors. Finally, to decrease the sampling frequency of energy-intensive sensors, we propose a multi-task learning strategy to predict the statuses of energy-intensive sensors based on the low-power sensors. To evaluate the performance of the proposed collaborative sensing framework, we develop a mobile application to collect concurrent heterogeneous data streams from all sensors embedded in Huawei Mate 8. The experimental results show that latent correlation learning is greatly helpful to understand the latent correlations among heterogeneous streams, and it is feasible to predict the statuses of energy-intensive sensors by low-power sensors with high accuracy and fast convergence. In terms of energy consumption, the proposed collaborative sensing framework is able to preserve the energy consumption of IoT devices by nearly 50% for continuous data acquisition tasks.

CCS Concepts: • Human-centered computing → Ubiquitous and mobile computing systems and tools; • Hardware → Power and energy.

Additional Key Words and Phrases: Energy efficiency; Latent correlation learning; Collaboration sensing; Internet of Things; Temporal Convolutional Network; Attention Mechanism; Multi-task learning

Authors’ addresses: Yunji Liang, liangyunji@nwpu.edu.cn, School of Computer Science, Northwestern Polytechnical University, Xi’an, China, 710129; Xin Wang, wangxin1248@mail.nwpu.edu.cn, School of Software, Northwestern Polytechnical University, Xi’an, China, 710129; Zhiwen Yu, zhiwenyu@nwpu.edu.cn, School of Computer Science, Northwestern Polytechnical University, Xi’an, China, 710129; Bin Guo, guob@nwpu.edu.cn, School of Computer Science, Northwestern Polytechnical University, Xi’an, China, 710129; Xiaolong Zheng, xiaolong.zheng@ia.ac.cn, Institute of Automation Chinese Academy of Sciences, Beijing, China, 100190; Sagar Samtani, ssamtani@iu.edu, Kelley School of Business, Indiana University, Bloomington, USA, 47405.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1550-4859/2021/1-ART1 $15.00
https://doi.org/10.1145/3448416

1 INTRODUCTION

The global Internet of Things (IoT) market size is projected to reach up to $1,102.6 billion by 2026 from $190.0 billions of 2018\textsuperscript{1}. The boom of IoT technology is transforming our daily life with numerous IoT devices ranging from smartphones and wearables to robots, autonomous vehicles and drones by seamlessly sensing ambient environments [24, 37]. IoT devices encompass a variety of sensors to sense external or internal environments. The diversity of sensors in IoT devices makes it possible to provide fine-grained and powerful sensing capability for advanced applications. For instance, sensors built-in smartphones can be used to recognize user activities and gestures [16, 51, 58], detect air pollution [38], recognize user emotion [12, 48, 49], detect mental problems [25], secure mobile devices via behavioral biometrics [3, 43, 62]. Moreover, sensors embedded in IoT devices can be used to support multi-modal interaction [27, 32], and to perform rich-sensor tasks [14, 58, 67].

However, the rich-sensor sensing and continuous inference are extremely challenging for resource-constrained IoT devices due to the limited power supply and the restricted computation capability. A recent survey [59] found that heterogeneous sensors in IoT devices depleted the power supply sharply. Although the diversity of heterogeneous sensors makes it suitable for high-impact applications, the huge energy consumption of heterogeneous sensors in IoT devices is becoming a vital problem. To address this problem, extant solutions can be characterized into two categories: energy harvesting and energy saving.

- Energy harvesting aims to harvest energy from ambient environments to eliminate the dependency of on-device batteries, and provides a viable alternative to autonomously power IoT devices [28]. The main energy sources could be kinetic, solar, radio frequency, and thermal [28]. For example, Alsharoa et al. used solar powered drones for energy management of cellular heterogeneous networks [2]; Ma et al. proposed a prototype for gesture recognition with solar powered devices [29]. Energy harvesting techniques address the challenges posed by the frequent energy source replacement, and offer one potential solution for a near perpetual deployment of sensors [36]. However, energy harvesting is still at its initial stage faced with numerous challenges including lower power supply and privacy leakage [28]. On the other hand, IoT devices powered by energy harvesting are low-power hardware which only supports the data collection and communication instead of complex inference tasks.

- Energy saving refers to the solutions that optimize the energy consumption of IoT devices. According to the differences of optimization objectives, energy saving technologies can be applied on different levels. For the rich-sensor sensing tasks, optimized scheduling strategies of sensors are proposed to reduce the energy consumption [26, 64]. For the computation-intensive inference tasks, compressed models [1, 34, 56] and dynamic distribution of inference tasks on distributed IoT devices [11] are studied to offload the computation. For the web applications, compress sensing is applied to minimize the energy consumption by reducing the frequency of network communication [4, 17].

Although numerous solutions have been proposed, the energy issue of IoT devices is still a challenging problem for the following reasons.

\textsuperscript{1}https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307
To enhance the sensing capability of IoT devices, the number of sensors embedded in IoT devices is increasing, resulting in the rising of energy consumption and costs of IoT devices. However, there is no study that investigates the latent correlations among heterogeneous sensors.

The energy consumption of heterogeneous sensors can be impacted by many factors. Extant studies find that physical sensors, network communication, and computation model have significant impacts on the energy supply of IoT devices. However, most studies only focus on one factor to optimize the energy consumption. How to quantify the energy consumption from different perspectives is an open question.

Majority of prior studies are dependent on one specific sensing or inference task. Extant solutions are not scalable and cannot be applied to other IoT devices. A generalized energy-efficient sensing framework is essential for the diversity of emerging IoT devices.

To address this problem, we propose a novel energy-efficient collaborative sensing framework to optimize the energy consumption of heterogeneous sensors built-in IoT devices. The proposed model is a general approach that could be applied into different IoT devices. Specifically, we examine the latent correlations among sensors to reveal the interwinding dynamics of heterogeneous sensors. Based on the latent correlations, we conduct data filtering to rule out the data streams that are unrelated with the given sensor. Next, we provide a generalized energy model to quantify the energy consumption of sensors. Accordingly, all sensors are categorized into energy-intensive sensors and low-power sensors. Finally, to reduce the energy consumption of IoT devices, we propose a multi-task learning model to predict the dynamics of energy-intensive sensors based on the statuses of low-power sensors. The primary contributions of this paper are summarized as follows.

To optimize the energy consumption of IoT devices, we propose a energy-efficient collaborative sensing framework to lower the sampling frequency of energy-intensive sensors. The proposed framework provides one efficient strategy to schedule heterogeneous sensors, and is scalable for rich-sensor IoT devices including smartphones and autonomous vehicles.

To quantify the correlation among heterogeneous data streams, we propose a novel attention mechanism-based temporal convolutional network to learn the latent interaction among multiple streams. To the best of our knowledge, this is the first study to quantify the latent correlations among heterogeneous sensors via attention mechanism.

To reduce the energy consumption of IoT devices, we propose a novel multi-task prediction model to estimate the statuses of energy-intensive sensors based on low-power sensors. The proposed prediction model is able to reduce the sampling frequency of energy-intensive sensors, which holds significant promise in decreasing the power consumption.

The remainder of this paper is organized as follows. In Section 2, we summarize prior studies in the field of collaborative sensing and energy-efficient sensing. We provide the problem definition in Section 3, and present the overview of the proposed collaborative sensing framework in Section 4. In Section 5, we present temporal attention-based convolutional network (ATT-TCN) to quantify the latent correlations among heterogeneous sensors. In Section 6, we provide a general energy model to characterize the energy consumption of IoT devices from multiple aspects. In Section 7, a multi-task transformer model is presented to predict the statuses of energy-intensive sensors with low-power sensors. We conduct extensive experiments to evaluate the performance of the proposed framework and present the experimental results in Section 8. After discussing the proposed solution in Section 9, we conclude this paper in Section 10.
2 RELATED WORK

In this section, we summarize the representative studies in the field of collaborative sensing and energy-efficient sensing for IoT devices.

2.1 Collaborative Sensing

The unprecedented sensing capability of heterogeneous sensors has been applied in several fields ranging from autonomous cars [14, 60, 67] and smart industry [9, 21] to public health [7, 54, 61] and historic site preservation [19, 33]. For example, Sato et al. [50] used machine learning to predict whether a user is tall or short based on a three-axis acceleration sensor during walking. Gupta et al. [14] designed a continuous authentication system for autonomous cars based on behavioral biometrics. They collected the real-time interaction data when users are operating the vehicle, and employed machine learning methods to recognize user identities. The accumulated data of sensors built-in smartphone were collected to predict whether the user is suffering from mental problems such as DeepMood [7, 54] and CrossCheck [61]. In addition, with the popularity of participant sensing [13], a variety of sensing tasks have been outsourced to participants according to their contexts such as location and preferences to perform a specific sensing task by using their mobile devices.

For the surveillance of noise pollution, sensing tasks are assigned to participants to monitor the noise pollution [44] and vehicle air pollution [38] in the urban with smart phones.

Although numerous applications have been developed to provide fine-grained sensory data for the decision making by fusing the outputs of heterogeneous sensors, they typically do not consider the energy consumption problem. However, prior works demonstrated that energy supply was the bottleneck of IoT applications. For example, Murakami et al. analyzed the energy consumption on mobile phones and found that the network communication would increase energy concerns even when the screen was turned off [35]. Pramanik et al. tapped into mobile phones to introduce the functions and working model of sensors, and conducted a comprehensive analysis to characterize their energy consumption. They pointed out sensors in smartphones should not be woken up for a long time, as they significantly consumed the battery energy resources [42]. These observations hold true for IoT devices. To characterize the energy consumption on IoT devices, Azni et al. conducted extensive experiments to analyze the energy consumption when transmitting data from IoT sensors to a remote server, and found that the communication between IoT devices and remote server depletes substantial energy supplies [4].

Apart from communication, the energy supply of IoT devices could be impacted by multiple factors. For example, Liang et al. found that the sampling frequency is an important factor to concern when performing rich-sensor sensing task and claimed the lower sampling frequency would be positive for energy-efficient sensing [26]. In addition, computing load is crucial for IoT devices. The computing load requires large memories and results in significant energy burden for IoT devices. Especially, deep learning methods leverage complex network architecture to learn the latent feature representation from large-scale datasets to achieve the state-of-the-art performance in numerous tasks. However, deep-learning based inference models pose great challenges for resource-constrained IoT devices. Deep learning models are becoming extremely complex with millions of hyper-parameters [53] and can overwhelm the limited computing resource in IoT devices. On the other hand, the training of deep learning models is time-consuming and computationally-intensive, which leads to the significant energy strains [68] and does not work well on resource-constrained IoT devices by debilitating levels of system overhead [23]. Therefore, the energy-efficient sensing and inference on IoT devices is indispensable.

However, there are several notable challenges. First, numerous sensors are integrated in IoT devices to sense more objects. The increasing number of sensors embedded in IoT devices means

high costs and huge energy consumption. Whether it is feasible to sense more objects with limited sensors has not been studied yet. Second, the energy supply is impacted by multiple factors. Extant studies only focus on one specific factor and analyze its impacts on energy consumption. There is no a uniform strategy to quantify the energy consumption of sensors from multiple perspectives.

2.2 Energy-Efficient Sensing in IoT

There are numerous studies about the energy-efficient sensing for IoT devices. As shown in Fig. 1, these studies can be divided into two categories: energy harvesting and energy conservation. Energy harvesting aims to harvest energy from ambient environments to eliminate the dependency of on-device batteries, and provides a viable alternative to autonomously power IoT devices [28]. However, energy harvesting is still at its initial stage and faces notable challenges including lower power supply and privacy leakage [28]. IoT devices powered by energy harvesting are low-power hardware, which can only support the data collection and communication and cannot perform complex inference tasks. In this paper, we mainly focus on the energy conservation technology for IoT devices. The energy conservation problem can be treated as an optimization problem, and extant solutions can be categorized into three groups in terms of optimization goals.

2.2.1 Optimize sampling frequency. Sensor frequency is critically important for the energy consumption and inference performance. To achieve high performance, researchers utilized high sampling frequency to capture high-resolution samples. For example, unmanned aerial vehicles perform data acquisition from on-board instrumentation with over 100 HZ sampling frequency to sense ambient environments [30]. The high sampling frequency of sensors is a double-edge sword. It provides fine-grained samples which may improve the inference performance in terms of accuracy. Moreover, the high sampling frequency implies a large of samples to be processed, which increases storage and computation requirements, resulting in the depletion of energy. Lower sampling frequency would provide a trade-off between accuracy and energy consumption [26]. To avoid the energy depletion caused by positioning services, Pan et al. proposed a solution to predict users’ trajectory using inertial sensors in smart phones [39]. Experimental results show that the proposed scheme not only is able to accurately derive the user’s walking trajectory, but it also effectively preserves the energy supply of mobile phones. Wu et al. identified the most
important sensors to sense the user behaviors, and exploited the importance of sensors to profile the duty-cycles of sensors for activity detection [64].

### 2.2.2 Optimize communication overload

To reduce communication redundancies, compressed sensing (CS) is widely applied in resource-constrained IoT devices. Compressed sensing is an approach to sample the signal at a rate much below the Nyquist sampling rate and to reconstruct the original signals from under-sampling measurements [45]. For the energy-efficient sensing of heterogeneous sensors, prior studies mainly focused on utilizing CS to lower the sampling rate or reduce the number of transmissions [55]. In addition, Sun et al. proposed a CS-based random sensing strategy that explicitly considers the heterogeneous energy consumption of IoT sensor nodes at different locations, in order to accurately attain a desired trade-off between the overall energy consumption and the sensing accuracy [55]. Du et al. designed a compressive sensing-based scheduling scheme to conserve energy of IoT devices by activating only a small subset of sensor nodes in each time slot to sense and transmit for the monitoring of water distribution networks [10]. Instead of assuming the each sample with the same cost, Xu et al. designed a cost-aware compressive sensing framework to deal with the cost-diversity of samples [65].

### 2.2.3 Optimize inference models

To fit deep-learning based models on resource-constrained IoT devices, numerous model compression strategies have been proposed including tensor decomposition [6, 22] and pruning [15, 68] to reduce the complexity of deep learning models. In addition, edge computing is shedding lights on mobile devices oriented deep learning [8, 70]. Via edge intelligence, the complex and energy-intensive deep neural networks can be partitioned into tiny subtasks, and be distributively performed on neighboring devices or edges [20, 66]. For example, Kang et al. designed a light-weight scheduler to automatically balance the computational offload between mobile devices and servers by partitioning the neural network layers [20]. Xu et al. proposed DeepWear to optimize the energy consumption of wearables by offloading the deep learning tasks among mobile devices [66].

### 3 PROBLEM DEFINITION

In this paper, we study the optimization strategy to minimize the energy consumption of heterogeneous sensors embedded in IoT devices. This problem can is formulated as follows. For one given IoT device $D$, sensors in device $D$ are denoted with $\mathcal{S} = \{S_1, S_2, ..., S_i, ..., S_n\}$, where $n$ refers to the number of sensors embedded in device $D$. To characterize the energy consumption of sensors, the energy consumption of sensors can be denoted with $E = \{e_1, e_2, ..., e_n\}$, where $e_i$ indicates the quantified energy consumption for sensor $S_i$. Based on the energy profiling of sensors, sensors in device $D$ can be categorized into two groups $\mathcal{S}_h = \{\forall S_i | S_i \in \mathcal{S} \wedge e_i > e_0\}$ and $\mathcal{S}_l = \{\forall S_i | S_i \in \mathcal{S} \wedge e_i \leq e_0\}$, where $\mathcal{S}_h \cap \mathcal{S}_l = \emptyset$ and $\mathcal{S}_h \cup \mathcal{S}_l = \mathcal{S}$; $e_0$ refers to the threshold value of the energy profiling. To minimize the energy consumption of device $D$, we aim to choose a set of sensors from $\mathcal{S}_l$ to predict the statuses of sensors in $\mathcal{S}_h$ by reducing the sampling frequency of energy-intensive sensors. To conduct energy-efficient collaborative sensing in IoT devices, we need to quantify sensor correlations. Algorithm 1 illustrates the pseudocode of this process.

**Procedure 1: Sensor Correlation.** For sensors $S_i, 1 \leq i \leq n$ embedded in device $D$, we need to find the dependency among sensors by learning their latent correlations. Specifically, for a given sensor $S_i \in \mathcal{S}$, the latent correlation learning aims to quantify the weights $w_{i,j} \in W$ of sensors $S_j, 1 \leq j \neq i \leq n$ for the statuses of $S_i$. The value of $w_{i,j}, 1 \leq j \neq i \leq n$ belongs to the interval $[0, 1]$, which indicates to what extent the two sensors are associated. When $w_{i,j} = 0$, there exists no correlation between sensor $S_i$ and $S_j$. The bigger the value of $w_{i,j}$ is, the tighter correlation exists between sensor $S_i$ and $S_j$. The workflow of the sensor correlation learning can be expressed by Algorithm 1, where $ATT-TCN(\cdot)$ refers to one attention-based neural network to
Algorithm 1 Quantify the Sensor Correlation

**Input:** Sequence outputs of sensors in $S$;  
**Output:** correlation matrix: $W \in \mathbb{R}^{n \times n}$, where $n$ is the number of sensors;  

1: for $i = 1$ to $n$ do  
2: denote the statuses of sensor $S_i$ with $f_{S_i}(\cdot)$;  
3: denote the outputs of others sensors with $f_{S - S_i}(\cdot)$;  
4: quantify the correlations among $S_i$ and $S - S_i$ by $w_i = \text{ATT-TCN}(f_{S_i}(\cdot), f_{S - S_i}(\cdot))$, where ATT-TCN(·) is one attention-based temporal convolutional network.  
5: end for  
6: return $W = [w_1, ..., w_i, ..., w_n]^T$

learn the latent correlations. In ATT-TCN, it uses $f_{S - S_i}(\cdot)$ as inputs and $f_{S_i}(\cdot)$ as outputs to train an attention-based deep neural network, where the trained attention weights are applied to quantify the latent correlations among sensors.

**Procedure 2: Energy Profiling.** For sensors $S_i$, $1 \leq i \leq n$ embedded in device $D$, we need a general energy model to quantify the energy consumption $e_i \in E$ for each sensor $S_i \in D$. Due to the energy consumption could be impacted by numerous factors, the generalized energy model should take as many factors into consideration. Based on the quantified energy indicator $e_i$, $1 \leq i \leq n$, all sensors $S_i$, $1 \leq i \leq n$ can be divided into two subsets $S_h$ and $S_l$, where $S_h$ refers to the set of sensors that are energy intensive, and $S_l$ refers to the sensors that have low power consumption (See Algorithm 2). Based on the energy profiling, we are able to characterize the heterogeneous built-in sensors in terms of energy consumption.

Algorithm 2 Energy Profiling

**Input:** Energy consumption $E = \{e_1, ..., e_i, ..., e_n\}$ of sensors in $S$; $S_h = \emptyset$; $S_l = \emptyset$;  
**Output:** $S_h$, $S_l$;  

1: for $i = 1$ to $n$ do  
2: if $e_i > e_0$ then  
3: $S_h = S_h \cup \{S_i\}$  
4: else  
5: $S_l = S_l \cup \{S_i\}$  
6: end if  
7: end for  
8: return $S_h$ and $S_l$

**Procedure 3: Sensor Prediction.** The sensor prediction aims to choose a minimal set of sensors from $S_l$ to predict the outputs of sensors in $S_h$. The assumption of sensor prediction is based on the fact that decreasing the sampling frequency of energy-intensive sensors is helpful for reducing the overall energy consumption [26]. The workflow of sensor prediction can be formulated as Algorithm 3. To find the function $\mathbb{F}(\cdot)$ with a minimal loss, we propose a multi-task solution to estimate the statuses of $S_l \in S_h$ with a set of sensors chosen from $S_l$. The details of function $\mathbb{F}(\cdot)$ are presented in Section 7.

4 OVERVIEW OF COLLABORATIVE SENSING FRAMEWORK

In this section, we propose a novel energy-efficient collaborative sensing framework for energy conservation in IoT devices. We assume that the energy-intensive sensors drain the power supply...
Algorithm 3 Predict energy-intensive sensors with low-power sensors

**Input:** Sequence outputs of sensors in $S$; $S_h$ and $S_l$; correlation matrix: $W \in \mathbb{R}^{n \times n}$;  
**Output:** function $F(\cdot)$; 

1: \textbf{for} $i = 1$ to $n$ \textbf{do}  
2: \hspace{1em} \textbf{if} $S_i \in S_h$ \textbf{then}  
3: \hspace{2em} obtain the statues of $S_i \in S_h$, denoted with $f_{S_i}(\cdot)$;  
4: \hspace{1em} \textbf{for} $j = 0$ to $n$ \textbf{do}  
5: \hspace{2em} \textbf{if} $w_{i,j} > 0$ and $S_j \in S_l$ \textbf{then}  
6: \hspace{3em} obtain the status of $S_j \in S_l$;  
7: \hspace{2em} \textbf{end if}  
8: \hspace{1em} \textbf{end for}  
9: \hspace{1em} find a function $F(\cdot)$ to minimize the loss between $f_{S_i}(\cdot)$ and $F(\cdot)$;  
10: \hspace{1em} \textbf{end if}  
11: \hspace{1em} \textbf{end for}  
12: \hspace{1em} return $F(\cdot)$

in IoT devices severally and in the rich-sensor sensing tasks decreasing the sampling frequency of energy-demanding sensors will be constructive for the energy conservation. Accordingly, we propose a collaborative sensing framework to predict the outputs of energy-intensive sensors. As shown in Fig. 2, the proposed framework is consisted of three components: latent correlation learning, energy profiling and hybrid sensor prediction.

- **Latent Correlation Learning:** In IoT devices, a variety of heterogeneous sensors are embedded to enrich the sensing capability. The integration of numerous sensors on the same circuit board renders the outputs of a sensor impacted by other sensors. For example, the speaker vibration can be tracked by the accelerometer to recognize the voice information for side attacks [31, 41, 47]. The vibration when interacting with touchscreen is used to recognize user identities for the security of IoT devices [52]. Latent correlation learning aims to find the correlations among heterogeneous sensors, which can help improve energy optimization.
- **Energy Profiling:** Sensors embedded in IoT devices have various working mechanisms with different performance in terms of energy consumption. Energy consumption of sensors is
determined by various factors including work mechanism, network communication, sampling frequency, etc. Energy profiling aims to model the energy consumption of sensors by considering multiple factors. Based on the energy profiling, sensors are categorized into two types: energy-intensive and low-power.

- Hybrid Sensor Prediction: As energy-intensive sensors cost significant energy supply, decreasing the sampling frequency of energy-intensive sensors is constructive to energy conservation. The hybrid sensor prediction aims to replace the energy-intensive sensors with low-power sensors. To achieve this goal, we predict the statuses of energy-intensive sensors with low-power sensors. The hybrid sensor prediction lowers the sampling frequency of energy-intensive sensors. It also shrinks the data volume for data processing.

The overall workflow of the proposed framework is shown in Fig. 2. The sequential outputs for a sensor can be taken as one sequential stream. Since the sampling frequencies for heterogeneous sensors could be different, the raw multiple data streams are aligned according to time slots. Missing data are filled with their previous statuses. For the aligned heterogeneous data streams, latent correlation learning is conducted to reveal the latent correlations among multiple data streams, by which the correlation matrix is constructed to quantify the extent to what the outputs of one given sensor can be impacted by other sensors. The strong correlation among sensors indicates that the status of one sensor changes with other sensors. For the energy profiling, it provides one unified energy model to quantify the energy consumption for sensors. In the energy model, it covers as many factors that are associated with the energy consumption to provide a fine-grained measurement of energy consumption. Based on the quantified energy consumption, sensors in one IoT device can be divided into two subcategories: energy-intensive sensors and low-power sensors. For the hybrid sensor prediction, it optimizes the sampling frequency of energy-intensive sensors by predicting their outputs with low-power sensors. To achieve this goal, we rely on the latent correlation learning to find a set of low-power sensors that have tight correlations with energy-intensive sensors, and provide one prediction model to precisely estimate the statuses of energy-intensive sensors.

5 LEARNING LATENT CORRELATIONS AMONG HETEROGENEOUS SENSORS

To study the latent correlation among sensors, we present a novel attention mechanism integrated into deep neural network to quantify the correlations among heterogeneous sensors.

5.1 Latent Correlation Learning

As shown in Procedure 1, latent correlation learning aims to learn the correlation matrix \( W \in \mathbb{R}^{n \times n} \), where \( w_{i,j} \) refers to the weight of correlation between sensor \( S_i \) and \( S_j \). The correlations of sensor \( S_i \) with other sensors \( S - \{S_i\} \) is denoted with \( w_i = [w_{i,1}, ..., w_{i,n}] \). To learn the vector \( w_i \), we propose an attention-based neural network to capture the latent correlations among heterogeneous sensors. The heterogeneous data streams from sensors \( S \) are aligned according to the maximal sampling frequency. After the data alignment, the missing data in each time slot is filled with its previous status. Next, we train one attention-based neural network with the data flow from sensor \( S_i \) as the output and the data streams from sensors \( S - \{S_i\} \) as inputs.

The architecture of attention-based neural network is shown in Fig. 3, where it consists of three layers: a attention layer, a TCN layer, and a fully-connected layer. The attention layer is responsible for weighting the importance of \( S - \{S_i\} \) to predict \( S_i \). The attention weights are treated as the latent correlations. The TCN layer is the core of latent correlation learning, which intrinsically extracts the relevant characteristics from multiple data flows and maps them to the desired data structure. The TCN layer first performs a dilated convolution to ensure that each hidden layer
in the convolution has the same size as the input sequence and to increase the receptive field. To build deeper neural network, a hopping layer connection for the residual convolution and a $1 \times 1$ convolution operation are added, where the $1 \times 1$ convolution can be used to reduce the dimension. Finally, the fully-connected layer maps the reduced features to one specific sequential data. Through this approach, the latent correlations among $S_i$ and $\mathbb{S}\{S_i\}$ are quantified by the attention weights. When $w_{i,j} = 0$, no association exists between sensor $S_i$ and $S_j$.

5.2 Attention based Temporal Convolutional Network

To quantify the correlation among heterogeneous sensors, we introduce a novel attention mechanism in temporal convolutional network (TCN) to weight the latent interaction among multiple sequential data flows. As shown in Fig. 4, the attention based TCN consists of four components: causal convolution, dilated convolution, residual connection and attention mechanism. Each is described in turn below.
5.2.1 Causal Convolution. Sequential problems can be characterized as the prediction of \( \{y_1, \ldots, y_t\} \) with \( \{x_1, \ldots, x_t\} \). Formally, a sequence modeling network is one mapping function \( f : x_1, \ldots, x_t \rightarrow y_t \) that satisfies the causal constraint that \( y_t \) depends only on \( x_1, \ldots, x_t \) and not on any "future" inputs \( x_{t+1}, \ldots, x_{t+n} \). The goal of learning in the sequence modeling setting is to find a mapping function \( f(\cdot) \) that minimizes the expected loss between the actual outputs \( \{y_1, \ldots, y_t\} \) and the prediction results \( f(x_1, \ldots, x_t) \), where the sequences and outputs are drawn according to some distribution [5].

\[
\hat{y}_1, \ldots, \hat{y}_t = f(x_1, \ldots, x_t)
\]

5.2.2 Dilated Convolution. The convolutional neural network (CNN) obtains a larger receptive field by adding a pooling layer, resulting in the problem of information loss after multiple pooling layers [63]. To address this problem, dilated convolution injects holes into the standard convolution to increase the receptive field. The advantage of dilated convolution is that it does not collect the loss information and increases the receptive field so that each convolution output contains the information of the conduction range. In a TCN model, for a 1-D sequence input \( x \in \mathbb{R}^n \) and a filter \( f : \{0, \ldots, k-1\} \rightarrow \mathbb{R} \), the dilated convolution operation \( F \) on elements of the sequence is defined as Eq. 2, where \( d \) is the dilation factor, \( k \) is the filter size, and \( s - d \cdot i \) accounts for the direction of the past.

\[
F(s) = (x \times df)(s) = \sum_{i=1}^{k-1} f(i) \cdot x_{(s-d-i)}
\]

The Fig. 4(a) illustrates the causal convolution and hollow convolution. The value of each layer at time \( t \) only depends on the statuses at time \( t, t-1, \ldots, 1 \) of the previous layer, reflecting the characteristics of causal convolution. The information extraction of the previous layer by each layer is leaping, and the expansion rate of each layer increases exponentially by 2. Due to the use of hollow convolution, padding is required for each layer (usually filled with 0), and the size of padding is \((k-1) \cdot d\).

5.2.3 Residual Block. In order to build deeper networks, residual blocks are introduced in TCN. Let the input of residual block denoted with \( x \), the nonlinear mapping we aim to obtain by learning within the dotted-line box consists of a series of hollow convolution, weight normalization, activation function, and dropout (See Fig. 4(b)). The output of the target mapping is denoted with \( f(x) \). To preserve the information from the original input, skip connections are introduced to parametrize the deviation from the identity, whose output is denoted with \( x + f(x) \). Residual blocks are a special case of highway networks without any gates in their skip connections, and allow the flow of memory (or information) from initial layers to last layers. Despite the absence of gates in their skip connections, residual networks perform as well as any other highway network in practice. Numerous studies applied the residual connection to build deeper network for significant performance gain. The skip connections in the residual blocks facilitate preserving the norm of the gradient, and leads to stable back-propagation [69].

Compared with RNN and its variants, TCN not only shows competitive performances in terms of accuracy and model size, it presents advantages in several aspects including parallelism, stable gradients, and low memory requirements [5]. First, TCN fixes the vanishing and exploding gradient problems and does not require time backpropagation during the training process. Second, dilated convolution in TCN makes the receptive field grow exponentially with the depth of the network, which makes it possible to build deeper neural networks. Third, TCN has a higher degree of parallelism than RNN, which makes their training and deployment more computationally feasible. Finally, TCN is able to learn the long-distance dependency from sequential input by using a hierarchy of temporal convolutional filters, pooling, and upsampling.
5.2.4 Attention Mechanism. To learn the correlations among heterogeneous sensors, we propose an attention mechanism in TCN to quantify the causal association among sensors. An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors \([57]\). The output is computed as a product of value and weight, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. We apply the attention mechanism in our model to calculate the weight of sensor data before TCN processes the data. First, an attention operation is performed on the incoming multiple sensor data. To implement this, the Input is branched out into three copies Query, Key, and Value \([57]\). Through the calculation of similarity between Query and each Key, then we can get the attention score \(s_{i,j}\), which is the importance of different temporal moments. This attention score \(s_{i,j}\) is then normalized by softmax to form a mask \(\alpha_{j,i}\). Finally, we multiply the Value by the mask \(\alpha_{j,i}\) to have the attention-weighted features \(\text{Input}_i\).

Specifically, the input vector Input is transformed into two feature spaces via \(\text{Query}(\text{Input}) = W_{\text{Query}}\text{Input}, \text{Key}(\text{Input}) = W_{\text{Key}}\text{Input}\), where \(W_{\text{Query}} \in \mathbb{R}^{n \times n}\) and \(W_{\text{Key}} \in \mathbb{R}^{n \times n}\). In this paper, Value is computed from Input with a \(1 \times 1\) convolution layer. Thus we have \(\text{Value}(\text{Input}_i) = W_{\text{Value}}\text{Input}_i\), where \(W_{\text{Value}} \in \mathbb{R}^{n \times n}\). The number of filters of Value is same as the channel size of Input. Query and Key are similar to Value. If \(\alpha_{j,i}\) indicates the extent to what the model attends to the \(i\)th location when synthesizing the \(j\)th region, \(\alpha_{j,i}\) can be quantified by Eq. 3, where \(s_{i,j} = \text{Query}(\text{Input}_j)\text{Key}(\text{Input}_i)^T\).

\[
\alpha_{j,i} = \frac{\exp(s_{i,j})}{\sum_{i=1}^{n} \exp(s_{i,j})} \quad (3)
\]

Then the output of the weighted attention map is \(\text{Att} = (\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_n) \in \mathbb{R}\), where \(\text{Att}_j\) is measured by Eq. 4. Finally, we multiply the input feature map with the weights. The modulated input \(\text{Input}_i\) can be formulated as Eq. 5. The modulated features \(\text{Input}_i\) will be fed into TCN.

\[
\text{Att}_j = \sum_{i=1}^{n} \alpha_{j,i}\text{Value}(\text{Input}_i) \quad (4)
\]

\[
\text{Input}_i = \text{Input}_i \times \text{Att}_j \quad (5)
\]

6 ENERGY PROFILING

In this section, we present a general energy model to quantify the energy consumption of sensors and conduct a case study to characterize the energy consumption of heterogeneous sensors in smartphone.

6.1 Generalized Energy Model

To measure the power consumption of a electronic device, according to the physical principle, the energy consumption can be quantified by Eq. 6, where \(W\) is the corresponding energy consumption, \(P\) is the power, \(t\) is the running time of a sensor, \(U\) is the voltage and \(I\) is the current. For IoT devices, they run with a constant voltage \(U\), thus the energy consumption can be transformed as Eq. 7, where \(\text{Power}\) represents the energy consumption of IoT devices with unit (mAh); \(I\) represents the current (in unit mA) when a sensor is running; \(t\) refers to the running time (in unit hour) of sensors. Eq. 7 indicates that only the current and running time are needed to measure the energy consumption of sensors.

\[
W = P \times t = U \times I \times t \quad (6)
\]
\[ \text{Power} = I \times t \] (7)

To cover multiple factors in the energy model, we investigate the lifecycles of sensors in IoT devices. Generally, the lifecycles of sensors can be summarized as follows: activation wake-up, data processing, network communication and state control. Thus, the generalized energy consumption, denoted with \( \text{Power}_{\text{sensor}} \), for one sensor can be characterized as Eq. 8, where \( \text{Power}_{\text{wake}} \) represents the energy required for the hardware to be activated; \( \text{Power}_{\text{data}} \) represents the energy required to process the sensed data during operation; \( \text{Power}_{\text{network}} \) indicates the energy required to transmit the processed data to a specific server backend. \( \text{Power}_{\text{command}} \) quantifies the energy required to receive the sensor control commands sent from the server backend and execute the corresponding sensor control commands.

\[ \text{Power}_{\text{sensor}} = \text{Power}_{\text{wake}} + \text{Power}_{\text{network}} + \text{Power}_{\text{data}} + \text{Power}_{\text{command}} \] (8)

6.2 Case Study of Energy Consumption Analysis

To evaluate the impacts of data collection tasks on the span time of battery, we conduct a comparative study to profile the energy consumption on Android mobile phones. Specifically, we use a fully charged Huawei Mate 8 and track its energy consumption until the power of battery is drained under different experimental settings. To avoid the impacts of other factors, we switch the mobile phone to the airplane model and turn off all background apps. We also prevent the screen from turning off by setting the turn-off of the monitor (display) to never. We turn off the auto-adjustment setting of screen brightness. To avoid overheating the battery, we place the phone in a refrigerator where the temperature is set as 5°C. We monitor the smartphone’s status until it automatically shuts down due to the low battery power. To compare the impacts of a rich-sensor inference task on the battery time span, we conduct the experiment two rounds in the same environment settings. In the first round, we track the time span when no sensing task is performed. In the second round, a mobile application developed by our team for data collection is launched to perform a continuous sensing task.

In the case study, we use the battery historian tool, an official API provided by Google, to view the device’s power consumption during the lifecycle. The visualized results are shown in Fig. 5. As shown in Fig. 5(a), when no sensing task is conducted on the phone, the span time of battery is around 24 hours. By contrast, when the developed app is performed, the span time of battery decreases up to 12 hours (See Fig. 5(b)). This indicates that the continuous rich-sensor sensing tasks take huge energy supplies and severely impair the performance of battery. According to the comparative studies, we observe that continuous sensing with heterogeneous sensors will drain the battery supply quickly.

6.3 Energy Consumption Analysis of Sensors

In the generalized energy model for IoT devices, we introduce four factors that typically impact the time span of battery: activation, data processing, network communication, and state control. Among them, the power consumption of sensor activation is determined by the hardware itself, which can be considered as constants. As the state control energy consumption of a sensor does not correspond to the demand in our perception task, we set the state control power consumption for all sensors as 0 mAh. Thus, we need to experimentally measure the energy consumption for network communication and data processing respectively.

To ensure the consistency of the experimental setting up, we perform the experiment in a same experimental setting up as elaborated in section 6.2. This experiment is divided into two stages. In stage one, we need to collect the fundamental data of mobile phones when no specific sensing task
or app is running. The fundamental data could be used to evaluate the lowest energy consumption of smart phones. Specifically, we switch the test device to flight mode and place it in a refrigerator where the temperature is set as 5°C for one hour. In stage two, we launch our software, and select the sensor to measure from the corresponding drop-down list and press the button to choose the type of energy consumption we want to measure. After the application setting, we placed the smartphone in the refrigerator to run for one hour. The energy consumption of the software was measured by the Battery Historian tool. This process is repeated until all types of power consumption of sensors have been collected.

The results of the calculation are the energy consumption of all sensors on different features. We plot the detailed summary of energy consumption results of all sensors in Fig. 6. We find that there are significant differences in energy consumption among different sensors in terms of Power\textsubscript{sensor}. Among all sensors, MageticField costs the highest energy supply, followed by Orientation. While RationVector, Game RationVector and Geo RationVector show similar performances in terms of power consumption. This is due to the fact that RationVector, Game RationVector and Geo RationVector are virtual sensors which are implemented by multiple accelerometer sensors. On the contrary, we observe that air pressure sensor, accelerometer and light sensor are low-power sensors with less energy consumption. Based on the energy profiling of sensors, we can categorize sensors into two groups according to the energy consumption: energy-intensive sensors and low-power sensors. We also perform an energy optimization task to reduce the sampling frequency of energy-intensive

![Fig. 5. Comparison of battery power consumption via Google battery historian](image-url)
sensors by predicting their statuses according to the latent correlations among sensors. We detail the optimization strategy in the next section.

Fig. 6. Profiling energy consumption of sensors built-in Huawei Mate 8

7 Prediction of Energy-Intensive Sensors Status

To reduce the energy consumption, we predict the outputs of energy-intensive sensors with a set of low-power sensors by decreasing the sampling frequency of energy-intensive sensors. In this section, we propose a multi-task solution to predict the realtime outputs of energy-intensive sensors.

7.1 Multi-Task Transformer Model

According to the definition of Sensor Prediction elaborated in Section 3, we need to predict the outputs of energy-intensive sensors $S_i \in \mathbb{S}_h$ based on statuses of a set of low-power sensors from $\mathbb{S}_l$. As the outputs of each sensor $S_i$ are three-dimensional sequential data, the proposed model should be able to predict the outputs of sensor $S_i$ on each axis respectively. To address this problem, we propose a multi-task transformer model to estimate the statuses of sensor $S_i$ on the three axes with the minimal loss.

The block diagram of the proposed multi-task transformer model is shown in Fig. 7. The model consists of four components: one transformer encoder and three transformer decoders. For the transformer encoder, it takes the aligned multiple heterogeneous data flows as the input, denoted with $X$. To evaluate how the sampling frequency of the energy-intensive sensors will impact the battery supply, we introduce a parameter $\lambda \in [0, 1]$ to control how many samples from energy-intensive sensors are applied in input $X$. When $\lambda = 0$, it means the samples from energy-intensive sensors are excluded from $X$. The higher $\lambda$ is, the more samples are chosen from energy-intensive sensors, which will boost the energy consumption of IoT devices. After the transformer encoder block, the latent representation of multiple heterogeneous data flows are denoted with $h_L$, where $L$ refers to the number of encoders stacked in the transformer encoder block.

The learnt latent representations are shared among three transformer decoders. Each decoder takes the $Output_X$, $Output_Y$ or $Output_Z$ of energy-intensive sensors as input, and completes the mapping operation between $h_L$ and $Output$ through the encoder-decoder attention. In the transformer decoder block, numerous transformer decoders can be stacked as well to construct deeper network for learning complex representation. Finally, the three transformer decoders are
applied to predict the statuses of energy-intensive sensors on each axis respectively. Through the prediction, we aim to reduce the sampling frequency of energy-intensive sensors to optimize the battery supply.

![Fig. 7. Multi-task transformer model for the prediction of energy-intensive sensors](image)

We use transformer architecture to encode the latent interaction among heterogeneous data streams. Transformer relies on the self-attention mechanism to compute representations of input and output without using sequence aligned RNNs or convolution. As shown in Fig. 8, the transformer is essentially an encoder-decoder structure, and the output of the encoder will be used as the input of the decoder. In transformer encoder, the data goes through a self-attention module to learn a weighted feature vector $h_L$, which is formulated as Eq. 9.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

(9)

After getting $h_L$, it will be sent to the next module of the encoder, namely Feed Forward Neural Network. This fully connected component has two layers. The activation function of the first layer is ReLU, and the second layer is a linear activation function, which is formulated as Eq. 10.

$$FFN(h_L) = max(0, h_LW_1 + b_1)W_2 + b_2$$

(10)

In the transformer decoder, the encoder-decoder attention is introduced to learn the relationship between the current data and the encoded feature vector. After decoding by the decoder, the decoded feature vector passes through a fully connected layer with a softmax activation function to obtain an output vector reflecting the probability of each data.

The multi-task transformer model has two advantages. First, it uses the attention mechanism to reduce the distance between any two positions in the sequence to a constant. Second, compared with RNN and CNN, the transformer model applies parallel computing to speed up data processing.
for large data volume problems. This not only improves the performance of the model but also conforms to the current hardware environment.

7.2 Loss Function

Given one sample of sensor $S_i$ at time $t$ denoted with $(x_t, y_t, z_t)$, the corresponding output of our proposed multi-task model for sensor $S_i$ at time $t$ can be denoted with $(\widehat{x}_t, \widehat{y}_t, \widehat{z}_t)$. To precisely predict the output of sensor $S_i$, we co-train the multi-task transformer model to minimize the loss function defined in Eq. 11.

$$
loss = \sum_{t=1}^{n} (||x_t - \widehat{x}_t||) + \sum_{t=1}^{n} (||y_t - \widehat{y}_t||) + \sum_{t=1}^{n} (||z_t - \widehat{z}_t||)
$$

(11)

To minimize the loss, we optimize this problem by the popular optimizer adaptive moment estimation (Adam). The reasons to choose Adam are three-fold: (i) it tunes the learning rate during the training automatically; (ii) it provides faster convergence compared with stochastic gradient descent algorithm; (iii) it is computationally efficient with less memory requirements.

8 EVALUATIONS

To evaluate the proposed energy-efficient collaborative sensing framework for IoT devices, we describe the data collection and present the experimental results about latent correlation learning and energy-intensive sensor prediction.

8.1 Test bed

In our experiment, we chose a Huawei Mate 8 smartphone as the test bed for the following reasons.

- As shown in Table 1, there are up to 14 sensors embedded in the Huawei Mate 8. The diversity of built-in sensors provides a perfect platform to learn the latent interaction among heterogeneous data streams.
- The Android operation system is open source. It is possible to collect the raw data of sensors with high precision and accuracy [40]. The fine-grained data streams can be explored to predict the statuses of energy-intensive sensors.
- Although there are many rich-sensor devices including drones and autonomous vehicles, we choose mobile phones as the test bed in view of its availability and the simplicity of operation procedures.
Table 1. List of built-in sensors in Huawei Mate 8

<table>
<thead>
<tr>
<th>Sensor Type</th>
<th>Sensor Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion Sensors</td>
<td>Accelerometer, Orientation, Groscope, Gravity, LinearAcc, RotationVector, GameRV, GeoRV</td>
</tr>
<tr>
<td>Position Sensors</td>
<td>MagneticField, Hall, SignificantMotion, StepCounter, Light, Airpress</td>
</tr>
</tbody>
</table>

8.2 Data Collection

In order to evaluate our proposed solution, we design a data acquisition application to collect data from multiple heterogeneous sensors. As shown in Fig. 9(a), the data acquisition application consists of two components: client side and server side. On the client side, the graphical user interfaces (GUI) are used to support the user interaction including phone registration, data collection, data uploading. In addition, we also use GUI to setup and retrieve the configuration information of sensors. The screenshots of our app are shown in Fig. 9(b). The server end is responsible for processing the collected data, and obtaining the sensor energy consumption information in the test bed.

Two voluntary participants are recruited from our research team. Each subject is assigned a smartphone (Huawei Mate 8) for the data collection. The developed app is pre-installed on the mobile phones, and the participants are trained about how to use the app for data collection. We encourage the participants to perform the data collection as much as possible during their daily life. Before starting the data collection, participants are able to set the maximal sampling frequency. The default maximal sampling frequency is up to 2000 Hz. When participants launch the data collection task, the real-time outputs of sensors packaged with timestamps are transmitted to the server side via WiFi. We conduct the data collection within three days. The total duration of data collection lasts 28 hours with over 10 million samples. The distribution of the collected samples is shown in Table 2. Obviously, there are significant differences among active sensors and inert sensors in terms of data volume. The active sensors generate a large number of samples with sampling frequency up to 1.1 KHz, and result in huge computation overload for mobile devices.

Fig. 9. Architecture of data acquisition application
Table 2. Overview of the dataset from Huawei Mate 8

<table>
<thead>
<tr>
<th>Sensor Type</th>
<th>Numbers of Samples</th>
<th>Sampling Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>63430</td>
<td>0.017 Hz</td>
</tr>
<tr>
<td>Airpress</td>
<td>2052060</td>
<td>0.55 KHz</td>
</tr>
<tr>
<td>MagneticField</td>
<td>2052060</td>
<td>0.55 KHz</td>
</tr>
<tr>
<td>GameRV</td>
<td>2052060</td>
<td>0.55 KHz</td>
</tr>
<tr>
<td>Orientation</td>
<td>2052060</td>
<td>0.55 KHz</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>2052060</td>
<td>0.55 KHz</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>2052060</td>
<td>0.55 KHz</td>
</tr>
<tr>
<td>Gravity</td>
<td>4134140</td>
<td>1.1 KHz</td>
</tr>
<tr>
<td>RotationVector</td>
<td>4134140</td>
<td>1.1 KHz</td>
</tr>
<tr>
<td>LinearAcc</td>
<td>4134140</td>
<td>1.1 KHz</td>
</tr>
</tbody>
</table>

8.3 Data Preprocessing
As heterogeneous sensors work in different models with different frequencies, we conduct data alignment to fit heterogeneous data streams with the proposed models for latent correlation learning and status prediction. First, we remove the samples collected from Hall, SignificantMotion, StepCounter, and Light due to the extreme low sampling frequency. For other heterogeneous data streams, we conduct data alignment and data filling according to the timestamp of samples. As shown in Fig. 10, we choose the sensor with highest sampling frequency, denoted as $S_1$, as the benchmark and use its corresponding timestamps to split the time duration. After the time span splitting, we get $n$ time slots, where $n$ is the number of samples collected from sensor $S_1$. The other heterogeneous data streams are fit into the specific time slots according to their timestamps.

![Fig. 10. The pipeline of data preprocessing](image)

8.4 Latent Correlation Analysis
As the outputs of sensors in our test bed are three-dimensional sequence data, we analyze the correlations between one specific axis of a given sensors and the three-dimensional statuses of other
sensors respectively. Specifically, to quantify the association among sensor $S_j$ and $\mathcal{S} - \{S_j\}$, we use the sequential statuses of sensor $S_j$ on x axis as the output sequence, and use the three-dimensional statuses of $\mathcal{S} - \{S_j\}$ as the input sequence. We feed the input sequence to the attention-based TCN model to fit the output sequence with the minimal loss. The attention weights are taken as the importance of sensors to predict the statuses of sensor $S_j$ on x axis. We iteratively conduct this procedure to learn the correlation matrix among sensors.

In our experiment, we conduct grid parameter searching to find the optimized settings for the network architecture of ATT-TCN. For the given network architecture, we randomly choose 80% of the collected data as the training dataset, and use the 20% as the test dataset. We repeat the latent correlation learning procedure, and obtain the attention weights in each training processing. According to the attention weights, we construct the correlation matrix, where the horizontal axis and the vertical axis indicate the order of sensors. Specifically, the order of sensors is Airpress, MagneticField, Orientation, Accelerometer, GameRV, GeoRV, Gravity, LinearAcc, RotationVector, Groscope.

![Fig. 11. Visualization of latent correlation among multiple heterogeneous sensors](image)

The latent correlations among multiple heterogeneous sensors in Huawei Mate 8 are shown in Fig. 11. According to the first column, we find that the statuses of Airpress on X axis are tightly associated with the statuses of MagneticField with the attention weight as 1. According to the third column, the statuses of Orientation on Y axis can be impacted by sensor Gravity. The correlation between each sensor is represented by a column. We observe our model can learn the latent correlation between different heterogeneous sensors without the help of external information. According to Fig. 11, we summarize the correlations among sensors in Table 3. For the MagneticField sensor, the outputs of axis Z are tightly associated with Orientation, accelerometer and rotation vector. After clarifying correlations among these sensors, we perform optimization procedure to lower the energy consumption of energy-intensive sensors.

Table 3. Latent correlations learnt from Huawei Mate 8

<table>
<thead>
<tr>
<th>Energy-Intensive Sensor</th>
<th>Low-Power Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
</tr>
<tr>
<td>MageticField</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Z</td>
</tr>
<tr>
<td>Orientation</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Z</td>
</tr>
</tbody>
</table>

8.5 Status Prediction of Energy-Intensive Sensors

8.5.1 Model Performance. Based on the latent correlation learning and energy profiling, we can extract the association knowledge among energy-intensive sensors and low-power sensors. This is the fundamental base to predict the statuses of energy-intensive sensors based on the association knowledge. Next, we conduct experiments to evaluate the performance of multi-task transformer model for status prediction. Based on the association knowledge shown in Table 3, we predict the statuses of MageticField and Orientation respectively based on the associated low-power sensors.

After the energy profiling for each sensor in smart phones, as shown in Fig. 6, we find that MageticField and Orientation sensors cost more power supply compared with other sensors. For the status prediction, we need to find the sensors that have tight correlations with MageticField and Orientation, and predict their statuses based on the other low-power sensors. To evaluate the impacts of sampling frequency on the prediction performance, we define one parameter variable $\lambda$, which defines how many samples are extracted from the energy-intensive sensors for the status estimation.

After the data preprocessing, we mix low-power sensors and energy-intensive sensors according to the ratio of $\lambda$ as the input of prediction model. The value of $\lambda$ is assigned as 0.5, which means that half of the raw data from energy-intensive sensors are mixed with the data from low-power sensors as the input of the prediction model. Meanwhile, the statuses of energy-intensive sensors are taken as the output. The training process aims to find a model to minimize biases between the predicted results and the statuses of energy-intensive sensors.

We implemented our model in Keras 2.3.1 with TensorFlow as a backend. The experiments were performed on an Intel Iris Plus Graphics 650 GPU with 8 GB memory. We set up a 6-layer encoder and a 6-layer decoder to implement the multi-task transformer model. The constructed transformer has 512 dimensions. The feed-forward size is 2048 and the dropout rate is 0.1. The number of multi-head attention is 5. The prediction results for MageticField are shown in Fig. 12, which presents the changes of losses with the increase of epoches. We compare the loss on each axis between the multi-task transformer model and the classical transformer model. According to Fig. 12, we observe that the proposed multi-task transformer model is able to predict the statuses of MageticField with less loss. Meanwhile, the proposed model converges quickly with fewer training epoches.

8.5.2 Effects of $\lambda$. In the experiment mentioned above, the value of $\lambda$ is fixed. To evaluate the effects of $\lambda$ on the model performance, we study the model performance by changing the value of $\lambda$. In our study, we use the low-power sensors to predict the statuses of MageticField and Orientation with the increase of $\lambda$ ranging from 0.1 to 0.6. The experimental results are shown in Table 4. Accordingly, when $\lambda = 0.1$, the prediction accuracies for MageticField and Orientation are up to 95%.
This demonstrates that it is feasible to predict the statuses of energy-intensive sensors based on the low-power sensors. Obviously, with the increase of $\lambda$, the prediction accuracies for MagneticField and Orientation show significant increase. On the contrast, the battery supply is quickly depleted with the increase of $\lambda$. For example, if $\lambda$ is assigned as 0.1, we can save battery power up to 62.6 mAh; while when $\lambda = 0.6$, the energy saving is around 26.6 mAh. This observation implies that the energy-intensive sensors working with high sampling frequency will drain the power supply of IoT devices quickly. However, how to set the sampling frequency of one sensor is challenging. When setting up the sampling frequency, we need to consider both the prediction accuracy and the energy issue. According to Table 4, although we can extend the time span of battery when $\lambda = 0.1$, the prediction accuracy for MagneticField will be hampered. When $\lambda = 0.5$, we are able to balance the prediction accuracy and energy saving.

8.5.3 Comparison with Benchmarks. To evaluate the performance of the proposed multi-task transformer model for status prediction, we compare the proposed solution with several benchmarks. For fairness, we adopt the multi-task learning framework by changing the transformer model with other sequential models. Specifically, the transformer model is replaced by RNN, LSTM, GRU, TCN, and ATT-TCN respectively. We choose these baselines for two reasons. First, RNN and its variants including LSTM and GRU are widely adopted to cope with sequence data, and achieve great success in numerous tasks. In this paper, we mainly cope with multiple sequence inputs. Therefore, it is reasonable to choose these methods as baselines. Second, we choose recent progress in sequence modeling(TCN and ATT-TCN) as baselines. Several studies show that TCN and ATT-TCN show competitive performance with less training complexity.

We compare these methods in terms of accuracy and loss. The experimental results are shown in Fig. 13. In terms of the prediction accuracy, transformer-based solution outperforms other baselines with up to over 10% margin gain. The significant improvement in accuracy indicates that our
Table 4. Effects of $\lambda$ on model performance

<table>
<thead>
<tr>
<th>Sensor Name</th>
<th>$\lambda$</th>
<th>Accuracy(%)</th>
<th>Energy Saving (mAh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MagenticField</td>
<td>0.1</td>
<td>95.13</td>
<td>62.6</td>
</tr>
<tr>
<td>MagenticField</td>
<td>0.2</td>
<td>95.85</td>
<td>55.6</td>
</tr>
<tr>
<td>MagenticField</td>
<td>0.3</td>
<td>97.50</td>
<td>48.7</td>
</tr>
<tr>
<td>MagenticField</td>
<td>0.4</td>
<td>98.36</td>
<td>41.7</td>
</tr>
<tr>
<td>MagenticField</td>
<td>0.5</td>
<td>99.76</td>
<td>34.8</td>
</tr>
<tr>
<td>MagenticField</td>
<td>0.6</td>
<td>99.99</td>
<td>26.6</td>
</tr>
<tr>
<td>Orientation</td>
<td>0.1</td>
<td>90.73</td>
<td>49.1</td>
</tr>
<tr>
<td>Orientation</td>
<td>0.2</td>
<td>92.66</td>
<td>43.7</td>
</tr>
<tr>
<td>Orientation</td>
<td>0.3</td>
<td>95.32</td>
<td>38.2</td>
</tr>
<tr>
<td>Orientation</td>
<td>0.4</td>
<td>96.98</td>
<td>32.8</td>
</tr>
<tr>
<td>Orientation</td>
<td>0.5</td>
<td>98.88</td>
<td>27.3</td>
</tr>
<tr>
<td>Orientation</td>
<td>0.6</td>
<td>99.53</td>
<td>20.5</td>
</tr>
</tbody>
</table>

(a) Trends of accuracy with the increase of epoches  
(b) Trends of training loss with the increase of epoches

Fig. 13. Performance comparison with baselines

The proposed model can be used to precisely predict the statuses of energy-intensive sensors based on low-power sensors. Meanwhile, the training loss of our proposed solution is minimal. In addition, according to Fig. 13, we observe that our proposed solution converges quickly, which is important for the training of deep learning models with lower computation overload.

9 DISCUSSION

9.1 Accuracy of latent correlation learning

To learn the latent correlations among heterogeneous sensors, we propose a novel attention-based neural network to quantify the latent interactions of multiple heterogeneous sequences. From the perspective of statistics, correlation refers to the strength and direction of a linear relationship between two random variables [18], and can be quantified by numerous statistics solutions including the maximum information coefficient (MIC) [46]. However, prior solutions do not fit well for the

analysis of multiple heterogeneous sequential inputs. To show the efficiency of the proposed solution for latent correlation learning, we conduct stationary analysis to characterize the static correlation coefficients among sensors. The results of the MIC analysis are shown in Fig. 14.

![Static correlation coefficients based on maximum information coefficient](image)

**Fig. 14.** Static correlation coefficients based on maximum information coefficient

We investigate whether the latent association learnt by ATT-TCN is consistent with that by MIC. According to Fig. 14, the association between RotationVector and MagneticField with MIC is 0.70; While the attention weight between them is 0.75. For the Gyroscope and Gravity, the quantified association with MIC is 0.30; That is 0.31 for ATT-TCN. We observe that the results based on MIC are consistent with the findings based on the attention-based latent correlation learning. This indicates that the proposed solution can quantify the static correlations among sensors. In addition, the proposed solution is an automatical and efficient solution to learn the latent correlations among heterogeneous data streams with no required external information and less computational overload.

### 9.2 Limitations

To conduct rich-sensor sensing tasks in IoT devices, we propose a novel energy-efficient collaborative sensing framework to reduce the energy consumption of sensing via the latent correlation learning and status prediction. Specifically, we propose a novel attention-based method to quantify the latent correlations among heterogeneous streams by the attention weights. To decrease the sampling frequencies of energy-intensive sensors for energy saving, we propose a multi-task transformer model to predict the statuses of energy-intensive sensors based on low-power sensors. To the best of our knowledge, this is the first work to conduct energy-efficient collaborative sensing via learning the latent correlations and reducing sampling frequency of energy-intensive sensors. However, this study is subject to the following limitations.

- For simplicity, we choose the Huawei Mate 8 as the test bed to evaluate the performance of the proposed solution. The proposed solution can be applied to other rich-sensor IoT platforms including drones, autonomous vehicles, and mobile robots. In addition, the collaborative sensing framework is suitable for sensing tasks that will transmit the sensing data to the cloud for data processing.
- We primarily focus on the optimization of energy-intensive sensors by decreasing their sampling frequency. However, many other factors including network communication and...
model complexity will impair the battery supply as well. In the proposed solution, we take advantage of the high parallelism of TCN and multi-task transformer model to reduce the complexity of neural networks. This makes it possible to deploy deep learning models in resource-constrained IoT devices. The communication load can have two potential solutions. First, the non-real-time applications, the data can be stored locally and the accumulated data can be transmitted to the server periodically. For the real-time applications, compressive sensing is promising to reduce the volume of data to be transmitted.

10 CONCLUSIONS

With the boom of IoT applications, numerous rich-sensor IoT devices (e.g., smartphone, autonomous vehicles, drones, etc.) perform a variety of sensing tasks. However, the power supply of IoT devices makes it infeasible for long-term continuous sensing. To improve the energy management, we propose an energy-efficient collaborative sensing framework. Specifically, we design an attention-based TCN to quantify the latent correlations among heterogeneous sensors, propose a generalized energy model to characterize sensors by their energy consumption, and utilize a multi-task model to predict the statuses of energy-intensive sensors based on low-power sensors. The proposed solutions are evaluated on a large-scale dataset collected from Huawei Mate 8. From the perspective of model training, we observe that the attention mechanism in TCN can be used to learn the latent correlations among multiple sequential inputs, and the transformer-based multi-task solution can be applied to predict the statuses of energy-intensive sensors. This ultimately proved helpful in reducing the sampling frequency of energy-intensive sensors for the optimization of energy supply. For the system perspective, the proposed energy-efficient collaborative sensing framework is feasible to optimize the power supply in IoT devices for performing rich-sensor sensing and inference tasks.

ACKNOWLEDGMENTS

This work is supported by the National Major Program for Technological Innovation 2030—New Generation Artificial Intelligence under grant No. 2018AAA0100500, by the natural science foundation of China under grant No. 61902320, and by the fundamental research funds for the central universities under grant No. 31020180QD140.

REFERENCES


