ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

DeepMetaIoT: A Multimodal Deep Learning Framework Harnessing Metadata for IoT Sensor Data Classification

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Inan, M.S.K., Liao, K., Shen, H., Jayaraman, P.P., Montori, F., Georgakopoulos, D. (2025). DeepMetaIoT: A Multimodal Deep Learning Framework Harnessing Metadata for IoT Sensor Data Classification. IEEE INTERNET OF THINGS JOURNAL, 12(20), 42352-42363 [10.1109/jiot.2025.3595556].

Availability:
This version is available at: https://hdl.handle.net/11585/1039273 since: 2026-01-26

Published:
DOI: http://doi.org/10.1109/jiot.2025.3595556

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

18 October 2026

DeepMetaIoT: A Multimodal Deep Learning Framework Harnessing Metadata for IoT Sensor Data Classification

Muhammad Sakib Khan Inan, Kewen Liao, Haifeng Shen, Prem Prakash Jayaraman, Federico Montori, and Dimitrios Georgakopoulos

***Abstract*—Internet of Things (IoT) sensor data, which capture time series physical measurements such as temperature and humidity, often lack proper classification. This limits their effective understanding, integration, and reuse. While sensor metadata—textual descriptions of the measurements—is sometimes available, it is frequently incomplete or ambiguous. As a result, classification often depends solely on the time series data. Leveraging both time series sensor readings and textual metadata for automated and accurate classification remains a challenge due to the heterogeneity and inconsistency of these data sources. In this paper, we propose DeepMetaIoT, a multimodal deep learning framework that integrates time series and textual data for classification. DeepMetaIoT employs a cross-residual architecture comprising a time series encoder and a text encoder based on a pre-trained large language model, enabling effective fusion of both modalities. Experimental results on real-world IoT sensor datasets show that DeepMetaIoT consistently outperforms state-of-the-art machine learning and deep learning baselines.**

***Index Terms*—IoT sensor data, sensor metadata, sensor time series, deep learning, multimodal, fusion, classification**

I. Introduction

IOT sensors are ubiquitous and found in many real-world environments, such as smart cities, smart buildings, and smart healthcare systems. They generate a vast and continuously increasing volume, variety, and velocity of sensor data[1] [1]. IoT applications analyze time series sensor readings[2] to solve real-world problems, providing tremendous economic benefits [1, 2]. A report by the McKinsey Global Institute projected that by 2030, IoT sensors could unlock $5.5 trillion to $12.6 trillion in global value [3].

The emergence of open IoT platforms like ThingSpeak [4] has enabled IoT practitioners worldwide to publicly share meaningful data collected from IoT sensors. Sensor data sharing promotes the re-usability of sensor data from already deployed IoT systems across the globe, supporting a wide range of IoT applications in domains such as environmental monitoring, healthcare, agriculture, and the energy sector [4]. To facilitate sharing, utilization, integration, and repurposing of open IoT sensor data for building public or proprietary IoT applications, the data or sensor readings must be clearly described, classified, or annotated using well-structured metadata[3] following certain ontologies [5–7]. However, users of open IoT platforms often do not follow a common ontology or use vocabulary-based annotation methods. As a result, the metadata describing the context of sensor readings are frequently incomplete or ambiguous. For example, on ThingSpeak [4], the vast majority of sensor data remain undescribed or is described ambiguously. User-annotated names for a temperature sensor, for instance, can vary significantly across users and channels, such as “tmp”, “temp”, “T@C” or “Teplota”. Furthermore, in the context of smart cities and buildings [8, 9], IoT sensors are annotated using diverse naming standards that vary across vendors. These real-world IoT scenarios give rise to the challenge of sensor metadata normalization [8, 9], and consequently, to the problem of sensor data classification [10, 11] for the effective management of heterogeneous sensor data.

The lack of universally standardized metadata can render IoT sensor data unusable, unsharable, or even undiscoverable [2, 12]. Furthermore, the manual classification or annotation of IoT sensor data incurs prohibitive labor and time costs [10, 13]. Nevertheless, the automated classification of IoT sensor data using machine learning (ML) techniques applied to sensor time series readings and/or textual metadata offers a promising solution. For example, if the type (*e.g.*, temperature, humidity, or light sensor) of an IoT sensor can be automatically classified, it can support various real-world IoT applications and operations such as, automatic discovery and registration of IoT sensors in large-scale data-sharing platforms [2], dynamic configuration of IoT systems when new sensors are added to a network [10], and recovery of IoT sensor type information when metadata is missing or ambiguous due to sensor malfunction or network communication issues [10, 14].

Recently, researchers have developed several machine learning (ML) methods such as Metadata-Assisted Classification Ensemble (MACE) [13] and TopK Sequential Ensemble (TKSE) [15] to incorporate both numerical time series data and textual metadata for IoT sensor data classification.

Corresponding Author: K. Liao (Email: kewen.liao@gmail.com)
M.S.K. Inan and K. Liao are with Deakin University, Australia
H. Shen is with Southern Cross University, Australia
P.P. Jayaraman and D. Georgakopoulos are with Swinburne University of Technology, Australia
F. Montori is with University of Bologna, Italy

[1]We define ”sensor data” as the time series datastreams of physical measurements (*e.g.*, temperature) recorded by IoT sensors.

[2]The term ”readings” in this study refers to physical measurements (*e.g.*, temperature, humidity, wind speed) from IoT sensors.

[3]The term ”metadata” refers to textual descriptions or annotations of sensor measurements, which helps to define and understand the data sensor generates.

The developments have led to a semantic annotation tool INFORM [12] that supports automated metadata annotation for IoT sensor data. INFORM facilitates machine-to-machine discovery of IoT sensor data so that data exchange can be repurposed without manual intervention. MACE and TKSE are customized filtering-based sequential ensemble learning methods in which each classifier is independently trained on a single source of information—either text or time series—and then sequentially combined in different ways during testing to produce the final classification. However, these sequential ensemble learning methods are prone to overfitting and can not learn relationships between data modalities. Also, the state-of-the-art, best-performing MACE is computationally expensive.

With the advancements of deep learning in Natural Language Processing (NLP), pre-trained Large Language Models (LLMs) like BERT [16] and GPT [17] have demonstrated their ability to learn semantic contextual features from unstructured and ambiguous text data which are similar to textual metadata from heterogeneous IoT sensors. However, to the best of our knowledge, such advanced pre-trained LLMs have not been adopted to learn sensor metadata embeddings[4]. Moreover, the multimodal deep learning approach has not been fully explored to effectively leverage information from both data modalities, namely, time series sensor measurements and textual metadata. In this study, we develop a novel multimodal deep learning framework incorporating a cross-residual dense fusion architecture to fuse the learned latent embeddings of numerical time series data and textual metadata. The fused representation is then used for classification so that the relationship between data modalities is captured. Our proposal achieves superior classification performance in practice compared to state-of-the-art methods. The main contributions of the study can be summarized as follows:

- We propose a novel end-to-end multimodal deep learning (DL) framework, DeepMetaIoT, that leverages both sensor metadata (via pre-trained LLM-based text encoder) and time series readings (via DL-based time series encoder) in a cross-residual dense fusion architecture for effective IoT sensor data classification.
- We create two new multimodal IoT sensor datasets, SensEURCity (SensEUR) and ThingSpeak–NEW (TS–N), both containing sensor textual metadata and numerical time series data. These datasets offer diverse and representative multimodal sensor data that contribute meaningfully to the field of IoT sensor data analytics. The TS–N dataset extends the previously proposed ThingSpeak–MACE (TS–M) dataset [13] by retrieving more diverse and recent sensor data from ThingSpeak [4], with each sample collected over a longer duration of one week (instead of one day) and an hourly sampling frequency (instead of every 15 minutes). The SensEUR dataset is derived from a publicly available sensor system data archive [18] and has been pre-processed into a multimodal IoT sensor dataset suitable for classification tasks.
- We conduct a comprehensive and rigorous experimental analysis to compare DeepMetaIoT with state-of-the-art ML and DL models which demonstrates its advantage.

The rest of the paper is organized as follows: Section II sheds light on previous related works on IoT sensor data classification, leading to Section III, where the proposed multimodal deep learning framework, DeepMetaIoT, is discussed with model architecture and algorithmic details. Section IV presents all the experimental results and comparisons with previous studies to validate the effectiveness and efficiency of DeepMetaIoT, leading to the discussions and conclusion in Sections V & VI.

II. Related Works

In this section, we organize the related works on IoT sensor data classification from the perspectives of unimodal (data) classification and multimodal (data) classification.

A. Unimodal classification

In the unimodal space of IoT sensor data classification, most previous studies focused only on classifying the numerical time series sensor readings. Those methods include: transforming time series into intermediate representations using statistical methods like ordinal pattern transformations [11] for class separability analysis, and incorporating string representation (symbolic approximation) [19] based transformation of IoT time series into graph topology-based analysis [20]. More recently, the machine learning method Random Forest was explored by previous studies [10] in a multi-class setting to classify IoT time series data early with less number of timestamps. Moreover, a deep learning method [21] was developed to capture complex non-linear patterns of heterogeneous IoT time series data to improve the classification performance.

In the context of smart buildings, studies either focus on only time series sensor readings or unstructured textual metadata of sensors. For example, Chen *et al.* [22] and Gao *et al.* [23] incorporated pre-computed statistical features from historical time series sensor readings for machine learning. Other studies [8, 9, 24] utilized only textual sensor metadata for classification. As pointed out by [12, 13, 25], unimodal classification methods are less effective on heterogeneous IoT sensor data. Therefore, new approaches considering both time series sensor readings and textual metadata are desired.

There are also fundamental time series classification methods that can be adapted for IoT sensor data classification. Those methods, though, were designed for the more regular time series benchmark datasets [11, 26]. The methods have evolved from the traditional distance-based classifiers such as Dynamic Time Warping and Weighted Dynamic Time Warping [26] to the recent advanced deep learning methods [27] such as the self-attention mechanism based transformer model PatchTST [28], patching-guided multi-layer perception-based model PatchTSMixer [29], and the very recent LLM-based model OFA [30].

[4]Embeddings in the context of deep learning are latent numerical feature vector representations that capture unique pattern from data to support tasks like classification.

B. Multimodal classification

Classification of IoT sensors sets a multimodal challenge in learning model development with sensor metadata and time series readings. Montori *et al.* [15] introduced an ensemble machine learning approach, TKSE, which employs a two-layer sequential ensemble of cascading classifiers for IoT sensor data classification using both sensor readings and metadata. In the context of smart buildings, Mishra *et al.* [25] proposed a unified architecture that combines machine learning with a set of rule-based strategies to support metadata tagging. Recently, Montori *et al.* [13] introduced MACE, a state-of-the-art ensemble machine learning framework incorporating classifier selection and ordering heuristics, and several class filtering strategies. All these methods use distance-based string-matching classifiers to deal with sensor metadata. However, string-matching classifiers cannot learn semantic features from the metadata text, so they are not robust to the ambiguity and heterogeneity presented in the sensor metadata texts [2]. Moreover, these methods train time series and text classifiers independently and then form an ensemble for final classification. The relationship between time series and text modalities can not be learned through such ensemble methods.

In the space of deep learning, recent development has witnessed several multimodal architectures such as ALBEF (Align Before Fuse) [31] and CLIP (Contrastive Language Image Pre-training) [32], both following a transformer-based cross-attention mechanism (with late fusion) to fuse multimodal features. This attention-based fusion strategy works well in general vision-language tasks where multimodal features share common or complementing information at the feature level. However, this approach is not particularly useful in the IoT sensor data classification domain (as also verified in our experiments), because textual metadata and time series readings do not align well at the feature level, unlike typical multimodal language-vision tasks [33, 34], where feature alignments commonly exist.

Previous efforts in other domains have been made by Rodrigues *et al.* [35] to combine time series data and textual data for taxi demand prediction. Their method utilized a common concatenation fusion strategy. A clear research gap exists in developing a multimodal deep learning framework that effectively utilizes time series readings and textual metadata for IoT sensor data classification.

III. The DeepMetaIoT Framework

In this section, we delve into the details of our proposed multimodal deep learning framework DeepMetaIoT. DeepMetaIoT's architecture can be divided into a deep learning-based Time Series Encoder, LLM-based Textual Metadata Encoder, and the key module of Multimodal Fusion. An overview of the framework is depicted in Fig. 1. From the figure, it can be seen that DeepMetaIoT incorporates both sensor readings and metadata as its inputs for the classification process. DeepMetaIoT embeds time series readings and metadata text with specific deep learning encoders and later fuses the encoded embeddings with our novelly designed two-way *cross-residual dense fusion module* for a joint learning process, effectively leveraging information from multiple domains.

A. Deep Learning based time series encoder

Let's define $\boldsymbol{s}$ as the one-dimensional vector containing time series data from an IoT sensor, where $\boldsymbol{s} \in \mathbb{R}^{t \times 1}$ and t denotes the length of time series. Then the two-dimensional vector $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{n-1}, \boldsymbol{s}_n]$ contains a total of n time series reading samples. In DeepMetaIoT, the raw sensor readings $\boldsymbol{s}_i \in \boldsymbol{S}$ are fed into a time series encoder ($\mathbf{E}_r$) to capture time series features and generate a latent feature space vector $\boldsymbol{T}_s$ for each $\boldsymbol{s}_i$. For this time series encoder, we have adopted our previous DeepHeteroIoT [21] model with several modifications. DeepHeteroIoT was originally designed to capture complex non-linear local and global features from heterogeneous IoT time series data. It consists of a set of convolutional blocks composed of Convolutional Neural Network (CNN) layers and one block containing a stack of Bi-GRU (Bidirectional Gated Recurrent Unit) layers. To adopt it into DeepMetaIoT as the time series encoder, we first exclude its MLP Classifier module (a stack of feedforward layers) after the depth-wise concatenation layer of global and local features. Furthermore, we add a Masking Layer[5] as the very first layer of the time series encoder to enhance the encoder's capability to deal with missing values in the sensor readings. The additional masking layer would allow our updated encoder to reason the presence of missing values in the original time series reading and learn the model weights accordingly. The above operations can be mathematically represented as below:

$$\boldsymbol{T}_s = \mathbf{E}_r(\boldsymbol{s}_i)\,, \tag{1}$$

$$\mathbf{E}_r(\boldsymbol{s}_i) = concat(\mathbf{conv}(\boldsymbol{s}_i), \mathbf{gru}(\boldsymbol{s}_i))\,. \tag{2}$$

B. LLM-based textual metadata encoder

As discussed earlier in Related Works, string-matching classifiers cannot learn semantic features from the metadata text, limiting the value of metadata. On the other hand, pre-trained large language models (LLMs) can uncover semantic relationships between texts through generated high-dimensional latent-vector representations, known as text embeddings. Pre-trained LLMs which are trained using siamese BERT networks (also called as sentence transformers) [36] to compute similarity between texts or sentences can be incorporated to generate numerical text embedding for a variety of natural language processing tasks. This pre-training [36] has already demonstrated its effectiveness in tasks such as semantic search, semantic textual similarity, and paraphrase mining.

In our proposed framework, we integrate a state-of-the-art lightweight LLM model, MiniLM [37], one of the smallest and fastest sentence transformer models [36] to support scalable IoT deployments. MiniLM is pre-trained using the Sentence Transformer framework [36] to generate numerical embeddings from the textual metadata of IoT sensors within DeepMetaIoT. This process transforms raw textual metadata into high-dimensional latent vector representations, enabling effective sensor data classification within our proposed multimodal deep learning framework.

Let's define $\boldsymbol{M}$ to contain the raw textual metadata from sensors, where $\boldsymbol{m}_i \in \boldsymbol{M}$. The pre-trained LLM (*i.e.*, MiniLM)

[5] https://www.tensorflow.org/api_docs/python/tf/keras/layers/Masking

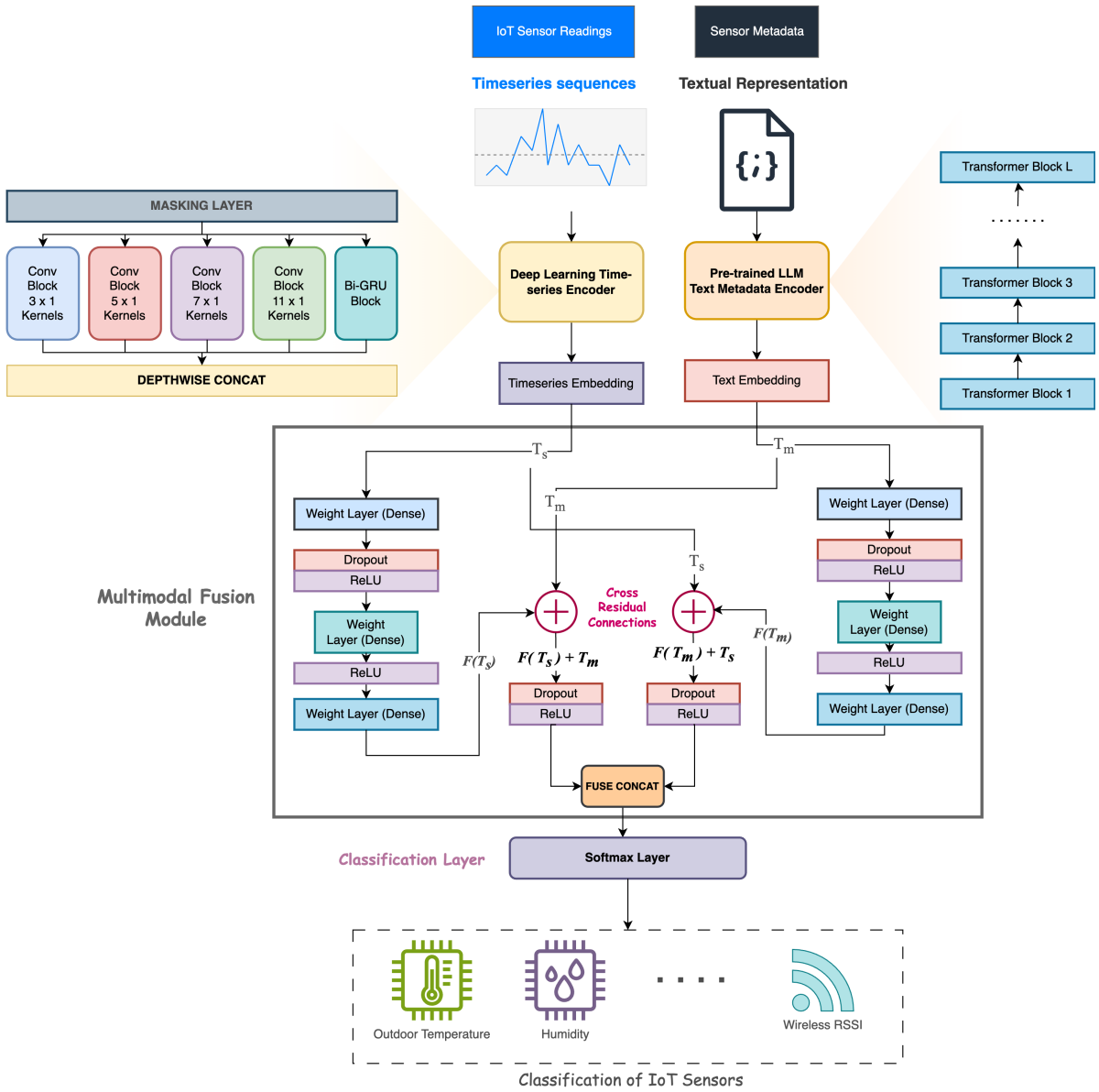


Fig. 1: Overview of the proposed multimodal deep learning framework DeepMetaIoT.

model incorporated into DeepMetaIoT serves as the metadata encoder $\mathbf{E}_m$ to generate the contextual embedding vector $\boldsymbol{T}_m$. MiniLM [37] utilizes a deep self-attention knowledge distillation mechanism through a teacher-student network to compress a pre-trained LLM into a smaller and faster architecture with fewer parameters. In our proposed framework, we incorporate this pre-trained MiniLM model that takes the textual sensor metadata $\boldsymbol{M}$ as input and embeds each $\boldsymbol{m}_i$ into a 384-dimensional dense latent vector, collectively representing the contextual text embedding $\boldsymbol{T}_m \in \mathbb{R}^{1\times 384}$. The operation is represented as:

$$\boldsymbol{T}_m = \mathbf{E}_m(\boldsymbol{M}) \tag{3}$$

C. Multimodal cross-residual dense fusion

The most crucial part of our proposed multimodal deep learning framework is the fusion module, which presents a novel cross-residual dense fusion strategy that effectively incorporates the latent feature space vectors $\boldsymbol{T}_s$ and $\boldsymbol{T}_m$ to make best use of both modalities (textual metadata and time series readings) for the final decision-making or classification. The original ResNet [38] architecture introduced skip-connections with convolutional layers for image data. A key building block in the original ResNet architecture is defined as:

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}, \{W_i\}) + \boldsymbol{x}\,, \tag{4}$$

where, $\boldsymbol{x}$ and $\boldsymbol{y}$ are accordingly input and output vectors for the respective neural network layer and the function $\mathcal{F}(\boldsymbol{x}, \{W_i\})$ represents the residual mapping to be learned where W_i is the weight vector of the respective layer i where, the layer number increases until the last output classification layer. Inspired by the residual architecture, we re-imagine the residual mapping and the use of shortcut connections to build our two-way multimodal fusion module based on dense feed-forward layers

instead of the original convolution layers. We incorporate dense layers in this stage, considering the better capability of fully connected layers in dealing with the integration of encoded latent-representation of multimodal features along with dimensional flexibility [39]. On the other hand, convolutions are generally designed for image data suitable for capturing spatial relationships [40] that are not present in encoded latent representation. The building blocks of the two-way fusion can be defined as:

$$\boldsymbol{T'}_s = \mathcal{F}(\boldsymbol{T}_s, \{W_i\}) + \boldsymbol{T}_m \,, \tag{5}$$

$$\boldsymbol{T'}_m = \mathcal{F}(\boldsymbol{T}_m, \{W_j\}) + \boldsymbol{T}_s \,. \tag{6}$$

Here, $\boldsymbol{T'}_s$ and $\boldsymbol{T'}_m$ are accordingly the resultant latent vectors after learning residual mappings with $\mathcal{F}(\boldsymbol{T}_s, \{W_i\})$ and $\mathcal{F}(\boldsymbol{T}_m, \{W_j\})$ and shortcut connection operations of $\mathcal{F} + \boldsymbol{T}_m$ and $\mathcal{F} + \boldsymbol{T}_s$ respectively. Considering the weight layers, the building block of $\mathcal{F}$ can be mathematically represented as:

$$\mathcal{F} = W_3\sigma(W_2\sigma(W_1\boldsymbol{x}))\,, \tag{7}$$

where σ denotes the non-linear activation operation ReLU [41] and $\boldsymbol{x}$ represents the input which is either $\boldsymbol{T}_s$ or $\boldsymbol{T}_m$ in two-way parallel residual blocks as illustrated in Fig. 1. Before applying the cross-residual operations, the dense weight layers (*e.g.*, W_1, W_2, W_3) are followed by dropout layers, standard layer normalization, and ReLU activation.

After the two-way element-wise addition operations ($\mathcal{F}+\boldsymbol{T}_s$ and $\mathcal{F}+\boldsymbol{T}_m$), we apply another dropout layers and ReLU non-linearity over the $\boldsymbol{T'}_s$ and $\boldsymbol{T'}_m$ vectors to get activated $\boldsymbol{T''}_s$ and $\boldsymbol{T''}_m$. Then followed by layer normalization, we concatenate $\boldsymbol{T''}_s$ and $\boldsymbol{T''}_m$ to get the fused feature vector $\boldsymbol{F}_c$ where:

$$\boldsymbol{T''}_s = \sigma(\boldsymbol{T'}_s)\,, \tag{8}$$

$$\boldsymbol{T''}_m = \sigma(\boldsymbol{T'}_m)\,, \tag{9}$$

$$\boldsymbol{F}_c = concat(\boldsymbol{T''}_s.\boldsymbol{T''}_m). \tag{10}$$

Finally, $\boldsymbol{F}_c$ goes through a Softmax activation layer for a probabilistic mapping $\boldsymbol{P} \in \mathbb{R}^{c\times 1}$ over all c class labels of IoT sensors.

$$\boldsymbol{P} = softmax(\boldsymbol{F}_c)\,. \tag{11}$$

An algorithmic representation in pseudocode format for our complete multimodal deep learning framework is delineated in Algorithm 1. It can be seen from the algorithm that for every epoch, raw data from both modalities are input into the network, and after passing through specific encoder modules, the proposed cross-residual dense module in-parallel takes one modality and its alternate modality to learn cross-modal latent features that impact the final IoT sensor data classification. The proposed deep learning architecture is trained for 200 epochs utilizing the cross-entropy loss function and Adam optimizer with a common learning rate of 0.0001.

D. Justification of the neural network design

To address the structural heterogeneity between IoT sensor time series data and textual metadata, we designed *DeepMetaIoT* with modality-specific encoders ($\mathbf{E}_m$ for metadata and $\mathbf{E}_r$ for time series readings) and a novel cross-residual dense fusion mechanism. The time series encoder ($\mathbf{E}_r$) captures both local and global temporal patterns, while the metadata encoder ($\mathbf{E}_m$) generates semantically rich text embeddings without requiring task-specific fine-tuning. Unlike traditional fusion methods that assume feature alignment across modalities, our proposed cross-residual fusion strategy enables modalities to influence each other through residual mappings – as illustrated in Eq. 5 and 6, where $\mathcal{F}(\boldsymbol{T}_s, \{W_i\})$ and $\mathcal{F}(\boldsymbol{T}_m, \{W_j\})$ denote multi-layer dense transformation functions. This mechanism facilitates joint latent feature learning despite modality asymmetry. The entire architecture forms an end-to-end trainable pipeline, avoiding the heuristic-driven ensemble complexity of prior state-of-the-art frameworks such as MACE [13].

Algorithm 1 DeepMetaIoT multimodal framework

Input: $\boldsymbol{S}$ and $\boldsymbol{M}$ are raw input data containing time series reading inputs and text inputs (metadata) from IoT sensors, respectively.

1: **for** Each *epoch* in *Epochs* **do**
2: For each $\boldsymbol{s}_i \in \boldsymbol{S}$ and $\boldsymbol{m}_i \in \boldsymbol{M}$: compute embedding vectors $\boldsymbol{T}_s$ using Eq. 1 and $\boldsymbol{T}_m$ using Eq. 3.
3: Compute $\boldsymbol{T'}_s$ using Eq. 5 and $\boldsymbol{T'}_m$ using Eq. 6 after residual mappings.
4: Perform multimodal learned feature concatenation for a final feature vector $\boldsymbol{F}_c$ using Eq. 10.
5: Compute the classification probability distribution $\boldsymbol{P}$ using Eq. 11.
6: Update weight vectors of respective layers based on the defined loss function and learning rate.
7: **end for**

Output: $\boldsymbol{P}$: predicted sensor class probability distribution.

IV. Experiments

This section details the experimental validation and analysis to demonstrate the effectiveness of our proposed deep learning framework DeepMetaIoT. We conduct rigorous experiments on several multimodal IoT sensor datasets for performance comparisons against baseline deep learning methods and state-of-the-art machine learning methods for IoT sensor data classification. The tested models and experiments were implemented using Python programming language on top of popular libraries such as Tensorflow, Pytorch, and Huggingface Transformers. The pre-trained LLM embeddings were extracted using a Sentence Transformer library[6], and the base deep learning architecture was primarily implemented with TensorFlow. The machine where the experiments were conducted is equipped with a Linux OS (Ubuntu 20.04 LTS), a 24-core Intel CPU (x86 64), 128 GiB of RAM, and an NVIDIA GeForce RTX 3080 GPU. Each experiment was designed with a stratified split over the datasets, assigning 70% to the training set and 30% to the testing set with a random seed value of 100 for each dataset using the Scikit-Learn library to maintain consistency with the previous benchmark study [13]. All the models included in this study for baseline comparison were

[6]https://sbert.net/

trained and tested across all three IoT sensor datasets with consistent experimental settings and training configurations to ensure a fair comparison of performance against our proposed model.

A. Datasets

In this subsection, we describe the multimodal IoT sensor datasets containing textual sensor metadata and numerical time series sensor readings. We utilize three such datasets in this study[7], summarized in Table I below. The first dataset is ThingSpeak MACE, or TS–M in short, which is the only existing multimodal open IoT sensor dataset adopted from the previous state-of-the-art study [13]. In addition, we create and contribute two new multimodal IoT sensor datasets, TS–N and SensEUR, for the purposes of 1) thoroughly evaluating our multimodal deep learning framework and 2) advancing the development of multimodal IoT sensor data classification. The TS–N dataset is also developed from the open IoT platform ThingSpeak [4] (similar to TS–M), but it covers recent 5-year data until 2024. The second new SensEUR dataset is built from the archive [18] that stores the sensed environmental data of several European cities. The development process of these datasets, along with their characteristics and analysis, is delineated in the following.

TABLE I: Summary of IoT sensor datasets. '*' denotes newly developed multimodal datasets from this work.

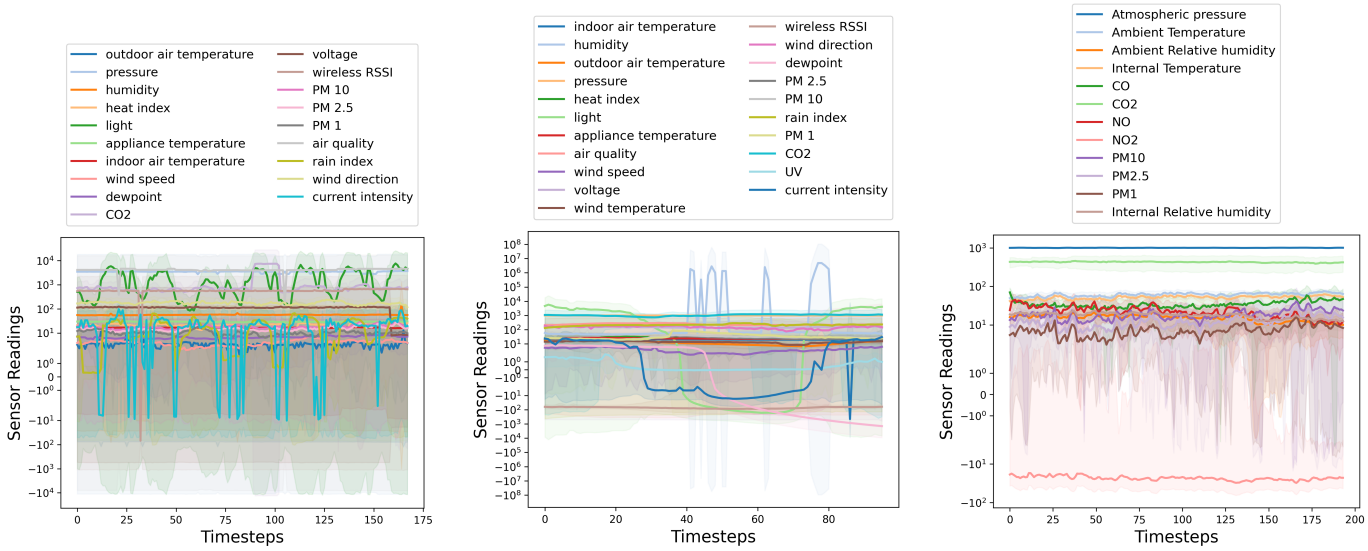
	TS–M	**TS–N***	**SensEUR***
Reading Length	96	168	194
Time Series Frequency	15 mins	Hourly	Daily
Total Samples	2120	1023	1090
Class Labels	21	19	12

1) ThingSpeak (TS) datasets: ThingSpeak[8] [4] is an online cloud platform that offers IoT analytics facilities along with allowing its users to send data to the cloud from IoT sensors in real-time. ThingSpeak offers public and private channels that users can create easily and send sensor data into channels. The sensor data hosted via public channels is accessible to anyone on the internet through API services. In general, the channels have a set of information along with sensor readings. This information includes user-annotated metadata: channel name, channel description, a name for each data sample or a series of sensor readings, and the latitude-longitude data. The metadata is user-annotated, often resulting in non-standardized, ambiguous, or incomplete information due to the lack of standardized vocabulary guidelines for annotating these sensor data. In this study, we utilized two versions of datasets created from the ThingSpeak platform, which are named ThingSpeak MACE (TS–M) and ThingSpeak NEW (TS–N). TS–M is a benchmark IoT sensor dataset collected by a previous study in the domain [13]. The TS–M dataset has 21 class labels including appliance temperature, humidity, pressure, indoor air temperature, outdoor air temperature, wind speed, light, air quality, wind direction, wind temperature, voltage, current intensity, wireless RSSI, heat index, dew point, rain index, UV, PM_1, $PM_{2.5}$, PM_{10}, and CO_2. The new ThingSpeak dataset (TS–N) that we contributed through this study contains all class labels of TS–M except UV and wind temperature. Furthermore, the new dataset contains time series sensor readings fetched from the same channels as TS–M, but the readings are more recent (from 2019 to 2024). TS–N also contains raw sensor name metadata (*e.g.*, "T@C", "temp", or "Teplota") from respective channels. However, unlike TS–M, the name metadata of TS–N does not require preprocessing, such as English translation or noise removal. This is more realistic but could also make the new dataset more challenging for existing methods. Table I highlights additional differences in terms of time series frequency and reading length between these two datasets. TS–M and TS–N datasets are country-wide datasets consisting of sensor data from European countries. TS–N dataset is perceived to contain more diverse temporal characteristics of sensor data due to greater time points, lower frequency, and longer duration of time series data collection. Fig. 2a and 2b below display the statistical means and standard deviations of the time series sensor reading data (for each timestamp by class label) in TS–N and TS–M, respectively. It can be seen that though drawn from similar sources, both datasets are evincing unique temporal patterns across labels, possibly due to the variability in sampling frequency and timestamp range. However, both datasets are common in high-level heterogeneous patterns and overlapping complex temporal patterns across class labels.

2) SensEURCity (abbrv. as SensEUR) dataset: The original SenEURCity data archive [18] contains raw numerical sensor data (non-calibrated) and textual sensor metadata from AirSensEUR sensor systems deployed in three cities of Europe (Antwerp, Oslo, and Zagreb). In each of these cities accordingly, 30, 28, and 27 sensor systems were deployed over a year to capture various meteorological and ambient conditions from the outside environment. In our study, we pre-processed these sensor readings along with the given sensor metadata to build a combined new IoT sensor data classification dataset. The developed IoT sensor dataset has multimodal information including both textual metadata from sensors and time series sensor readings against each class label. As a result, the dataset contains 12 unique class labels which are: atmospheric pressure, ambient temperature, ambient relative humidity, internal temperature, internal relative humidity, CO, CO_2, NO, NO_2, PM_{10}, $PM_{2.5}$, PM_1. Fig. 2c depicts the statistical means and standard deviations of the time series sensor readings in the SensEURCity dataset. It can be seen that though the temporal patterns are quite separable from each other across class labels, there exist high standard deviations of readings, indicating the existence of variability or heterogeneity in sensor readings. The mean values of temporal sequences or readings from the sensor data types of atmospheric pressure, CO_2, and internal relative humidity have substantial disparity in magnitudes compared to other types.

In addition, Fig. 3 illustrates an example of a particular sensor's textual metadata from the SensEURCity dataset. The data/archive originally contains sensor readings at every minute and commonly with missing timestamps. To develop a dataset with another representative sensor reading frequency,

[7] https://github.com/skinan/DeepMetaIoT
[8] https://ThingSpeak.com/



(a) TS–N (b) TS–M (c) SensEUR

Fig. 2: Visualization of sensor readings (mean and standard deviation) for the (a) TS–N, (b) TS–M, and (c) SensEUR datasets. The y-axis follows a symmetrical logarithmic scale.

i.e., daily for SensEURCity *vs.* hourly for TS–N *vs.* every 15 minutes for TS–M, we first pre-process the raw data into a mean sensor reading of each day so the dataset is reduced to one data point per day. In the final dataset, we do not include a data sample consisting of preprocessed daily sensor readings if more than 25% of daily timestamps are missing over the duration of data collection.

```
{
    'supplier': 'Bosh Sensortech',
    'sensors': 'BMP280',
    'type': 'Piezo-resistive',
    'units': 'hPa'
}
```

Fig. 3: Example of a sensor's textual metadata from the SensEURCity dataset.

Overall, as shown in the dataset details and the summary provided in Table I, it is evident that the newly added multimodal IoT sensor datasets, TS–N and SensEUR, contain more diverse and representative data samples compared to the existing TS–M dataset. For instance, while the TS–M dataset includes readings over a 24-hour period for each sample, TS–N contains 1 week of data per sample, and SensEUR provides 1 year of data per sample. As a result, the new datasets offer greater diversity in sensor readings and better reflect the dynamic nature of real-world environments.

Regarding the metadata, the TS–M dataset includes extensively pre-processed metadata (*e.g.*, translated to English). In contrast, for the TS–N dataset, we intentionally avoided over-processing the raw metadata to preserve the diversity and variety typically found on open IoT platforms such as ThingSpeak [4], where data providers from different countries or regions may use distinct dialects or language structures.

While preparing the two new IoT sensor datasets (TS–N and SensEUR), we paid close attention to IoT data quality evaluation frameworks [42, 43] to ensure the reliability of the datasets used in this study. The key quality dimensions we considered from IoT data quality include correctness, completeness, consistency, and timeliness [42, 43]. For example, in the SensEUR dataset, to ensure the correctness of sensor readings, we only included sensor data at a particular timestamp if it successfully passed the QA/QC (Quality Assurance / Quality Control) procedures defined by sensor providers [18]. For both datasets, to maintain completeness and consistency, we pre-processed the data and removed any sensor data samples with more than 25% of missing values. Regarding metadata quality, for the TS–N dataset, we conducted a manual review of each sample to ensure the metadata is meaningful. Finally, both datasets were compiled using the most recent available data to ensure timeliness for the newly contributed datasets.

B. *Performance evaluation metrics*

The performance evaluation metrics used in the experiments are as follows. They are incorporated throughout the experimental validation of our proposed framework against all other baseline methods.

- *Accuracy:* It indicates the percentage of IoT sensor data that have been correctly classified from all the tested sensor data samples.
- *F1-Score:* It is the harmonic mean of Precision and Recall giving us a balanced perception of model performance. The macro-average version of the F1-score is incorporated as a metric in our study, considering the existing class imbalance in the IoT sensor datasets.

C. Baselines

To demonstrate the effectiveness of our proposed deep learning framework (DeepMetaIoT), we set recent state-of-the-art (SOTA) models as competitive baseline methods for comparison. Although the classification of IoT sensor (measurement) types using multimodal data is a growing field, there is a notable lack of baseline multimodal architectures. The most relevant and recent multimodal model in this domain is the ensemble machine learning model MACE [13], which has four versions based on different model selection and filtering strategies. We consider all four versions of MACE as state-of-the-art baselines and provide rigorous experimental results across all IoT sensor datasets. Additionally, we include the attention-based Fusion Transformer [44] architecture to evaluate a Transformer-based multimodal fusion approach across all datasets. Finally, we also compare our proposed fusion strategy with two commonly used fusion methods: Concat Fusion (CF) [35] and Cross Attention Fusion (CAF) [45] to establish a strong set of baselines. More details are given below:

1) MACE [13]: This is our most relevant baseline model and the biggest competitor, which was specifically designed for the multimodal IoT sensor data classification. It is a sequential ensemble learning model that incorporates both textual metadata and time series sensor readings into one unified framework by independently training multiple machine learning models and, in the testing phase, heuristically selecting and ordering the trained classifier models and incorporating sequential filtering strategies on the model output classes. To handle sensor readings, MACE extracts statistical features from time series readings for classification. For textual metadata, MACE incorporates an NLP classifier built on a distance-based (Damerau-Levenshtein edit distance) dictionary classifier. In this study, we include the 4 versions of MACE: *TopK*, *PF* (Percentage Filtering), *SoF* and *Bruteforce* search and filtering strategy. Both *TopK* and *PF* are rank-based adaptive strategies where the latter is a more generalized version of the former. The *SoF* version can efficiently organize the classifier models using probability distributions output by the classifiers while the *Bruteforce* version tries all possible classifier combinations and orderings, which is more accurate than *SoF* but computationally much more expensive.

2) Fusion Transformer [44]: This multimodal fusion transformer architecture utilizes the AutoGluon-Multimodal [44] framework, which incorporates a multi-layer perceptron (MLP) for the time series reading component and a pretrained BERT backbone model for the textual metadata component, both fused using multi-head transformer modules. This architecture closely resembles recent popular transformer-based fusion strategies commonly found in vision-language multimodal domains.

3) DeepMetaIoT–CF [35]: This fusion strategy (Concat Fusion strategy abbrv. as CF) is designed to directly concatenate latent-space vectors from the time series encoder and pre-trained LLM encoder in an appending manner. Generally, it falls into the category of late fusion strategies in multimodal learning. This fusion strategy serves as a common baseline for the multimodal capabilities of a deep learning model. Previous studies [35] have utilized this type of fusion strategy for the fusion of text data and time series data.

4) DeepMetaIoT–CAF [45]: This fusion strategy (Cross Attention Fusion strategy abbrv. as CAF) is designed to follow the popular attention mechanism [45] adopted in recent popular multimodal deep learning models specifically designed for the vision-language domain [31, 32]. However, in this strategy, the latent space vector from the time series encoder and pre-trained LLM-based text encoder is incorporated into the attention module in alternating query-value orientation.

D. Performance comparison : *DeepMetaIoT* vs. *baselines*

Table II present the classification performances of our proposed multimodal deep learning framework, DeepMetaIoT, based on standard evaluation metrics (Accuracy and F1-Score) and against state-of-the-art baseline methods across different multimodal datasets. Overall, it can be seen that DeepMetaIoT has achieved the highest average accuracy and F1-score of 89.69% and 92.63% accordingly across IoT sensor datasets. These key performance indicators of DeepMetaIoT are clearly superior compared to the SOTA baselines presented in the tables, demonstrating the effectiveness of our proposed multimodal deep learning framework. However, MACE–Bruteforce is the second-best model in terms of average accuracy and F1 score, with values of 86.51% and 83.34%, respectively. MACE–SoF has shown slightly lower accuracy and F1-score compared to MACE–Bruteforce but still outperforms other deep learning-based models. For the SensEUR dataset, MACE–Bruteforce and MACE–SoF demonstrate a very competitive classification performance compared to DeepMetaIoT, where DeepMetaIoT outperforms slightly by 2.14% and 0.99% in terms of accuracy and F1 score, respectively. This indicates that ensemble machine learning approaches could still be highly effective in less heterogeneous datasets like SensEUR (see Fig. 2c) where sensors come from a single domain (*e.g.*, Smart City). For both the old and new ThingSpeak datasets (*i.e.*, TS–M and TS–N) with high data heterogeneity (see Fig. 2b and 2a), DeepMetaIoT demonstrated a clear advantage over the second-place MACE–Bruteforce, with an absolute improvement in accuracy of 3.81% and 3.6%, respectively. The Fusion Transformer model presented a competitive edge on the TS–M dataset, achieving the second-best F1-score with a value of 80.85%. It also performed comparably to MACE–Bruteforce in terms of accuracy on the TS–M dataset. However, for the other two datasets (TS–N and SensEUR), the Fusion Transformer lagged behind significantly in both accuracy and F1-score. As shown in Table II, the top two performing classification models are DeepMetaIoT and MACE–Bruteforce.

In terms of training efficiency, DeepMetaIoT leads with the lowest average training time (in minutes) and the lowest number of approximate trainable parameters (in millions) or model size, as shown in Table III. This is because the well-performing MACE models rely on sequential ensembles of multiple individual classifiers, which require considerable time to construct. In contrast, DeepMetaIoT operates as an end-to-end pipeline without the need for sequential parameter

TABLE II: Comparative performance on multimodal classification of DeepMetaIoT against state-of-the-art baselines across IoT Sensor Datasets. **Bold**: best, Underline: second best

	TS–M		TS–N		SensEUR		Average	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
MACE–TopK	84.30	68.19	71.57	76.54	91.43	93.28	82.43	79.34
MACE–PF	83.35	64.66	71.57	76.01	91.43	93.28	82.12	77.98
MACE–SoF	88.22	78.38	67.97	71.53	93.58	95.40	83.26	81.77
MACE–Bruteforce	88.50	76.70	77.45	77.91	93.58	95.40	86.51	83.34
Fusion Transformer	87.28	80.85	69.61	57.72	71.56	72.97	76.15	70.51
DeepMetaIoT–CF	87.28	69.50	67.97	56.59	94.80	95.82	83.35	73.97
DeepMetaIoT–CAF	69.86	27.41	60.46	25.84	93.27	93.63	74.53	48.96
DeepMetaIoT (Proposed)	**92.31**	**94.61**	**81.05**	**86.90**	**95.72**	**96.39**	**89.69**	**92.63**

TABLE III: Trade-off analysis between average accuracy, training time, model size (trainable parameters), and inference time for top baselines and the proposed *DeepMetaIoT* framework

	Average Accuracy	Training Time (minutes)	Model Size (millions)	Inference Time (seconds)
MACE–SoF	83.26	210	6.86	6.5
MACE–Bruteforce	86.51	1480	772	6.5
Fusion Transformer	76.15	43	118	1.1
DeepMetaIoT	89.69	16	2.74	2.3

selection. Although MACE–Bruteforce demonstrates competitive classification performance across IoT sensor datasets compared to DeepMetaIoT, Table III shows that the average training time for MACE–Bruteforce is significantly higher, leading to greater computational costs compared to DeepMetaIoT and MACE–SoF. This is expected, as MACE–Bruteforce explores all combinations of individually trained machine learning models. As a result, it also has the highest number of trainable parameters (approximate). MACE–SoF, on the other hand, achieves relatively lower computational costs by using crafted heuristics to build the ensemble model much faster (210 minutes *vs.* 1480 minutes) than MACE–Bruteforce. The Fusion Transformer architecture is faster to train compared to MACE–Bruteforce and MACE–SoF, but it does not achieve highly competitive accuracy or F1 scores across IoT datasets. Inference times remain relatively low for all models: approximately 6.5 seconds for MACE–Bruteforce, 1.1 seconds for Fusion Transformer, and 2.3 seconds for DeepMetaIoT on average across datasets.

Therefore, DeepMetaIoT proves to be a better choice both in terms of effectiveness and efficiency. In the context of IoT systems, such efficiency is crucial for maintaining the scalability of models in resource-constrained environments.

E. Ablation study

In this section, we present an ablation analysis of DeepMetaIoT to validate the impact of incorporating multimodality over single-modality, and to assess the effectiveness of different components, including the time series encoder and text metadata encoder. Table IV provides the mean and standard deviation of test accuracies and F1-scores for the best five epochs of different ablation combinations, including classification using only time series data (*i.e.*, Time series only) and only textual metadata (*i.e.*, Metadata only). According to the DeepMetaIoT architecture shown in Fig. 1, the "Time series only" ablation incorporates only the time series encoder from our framework (before the softmax classification layer), while the "Metadata only" ablation incorporates the pre-trained LLM text metadata encoder. We also evaluate the effectiveness of DeepMetaIoT's time series encoder by comparing the results with fundamental state-of-the-art time series models such as PatchTST [28], PatchTSMixer [29], and OFA [30]. For these fundamental time series models, we follow the previously described concat fusion strategy, alternating the time series encoder component.

Finally, we conduct a rigorous experimental analysis of DeepMetaIoT by alternating its pre-trained LLM encoder with state-of-the-art LLM models like BERT [16] and DistillBERT [46], as well as the non-LLM-based word embedding model GloVe [47]. It can be clearly seen from the Table IV that DeepMetaIoT outperforms all other ablation combinations across datasets, again demonstrating the framework's effectiveness and stability.

Furthermore, we also evaluate DeepMetaIoT framework under varying degrees of missing time series sensor readings by conducting experiments with different percentages of missing data. From the experimental analysis, we found that as the percentage of missing sensor readings increased, the model's test accuracy generally decreased, particularly when missing values in training data exceeded 50%. For example, accuracy on the TS–M dataset dropped from 90.89% with 10% missing sensor readings to 83.67% with 80% missing sensor readings. However, DeepMetaIoT still maintained commendable performance across all datasets, never dropping below 70% accuracy. This robustness is due to the model's multimodal design, which helps mitigate the impact of missing sensor readings by leveraging textual metadata to support classification decisions. For further improvement, missing data can also be handled during preprocessing using various time series imputation techniques [48].

V. Discussion & Limitation

Compared to the existing state-of-the-art approaches, DeepMetaIoT is an end-to-end framework that can robustly learn non-linear patterns from time series sensor readings and textual sensor metadata and then novelly fuse the learned patterns for classification. With an end-to-end neural network design, the proposed model is more scalable than the previous state-of-the-art ensemble machine learning approach, MACE [13],

TABLE IV: Ablation analysis of DeepMetaIoT framework in terms of Accuracy and F1-Score across IoT sensor datasets

	TS–M		**TS–N**		**SensEUR**	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
Time series only	78.02±0.52	39.03±1.33	59.41±0.71	52.82±0.43	94.01±0.17	95.40±0.16
Metadata only	76.36±0.14	66.01±0.45	78.50±0.43	78.78±0.14	58.10±0.00	58.56±0.00
PatchTST with meta	59.53±0.37	35.40±0.78	37.65±0.63	22.69±0.31	59.76±0.17	59.53±0.44
PatchTSMixer with meta	86.97±0.00	86.76±0.19	70.20±0.27	67.19±0.19	62.51±0.51	66.06±0.43
OFA with meta	80.13±0.24	57.84±0.50	60.00±0.37	36.56±0.51	82.32±0.50	85.00±0.46
DeepMetaIoT–GloVe	89.17±0.00	80.26±0.41	72.16±0.43	58.21±1.06	94.25±0.33	95.24±0.23
DeepMetaIoT–BERT	90.27±0.22	85.84±0.13	76.93±0.18	77.17±0.36	93.52±0.26	94.93±0.23
DeepMetaIoT–DistillBERT	90.52±0.14	87.78±0.26	80.52±0.18	84.14±0.56	95.05±0.26	95.97±0.16
DeepMetaIoT–Proposed	**92.12±0.13**	**93.91±0.54**	**80.72±0.33**	**86.20±0.42**	**95.41±0.22**	**96.19±0.14**

which is not fully end-to-end and relies on a brute-force strategy, resulting in a computationally expensive process.

Once trained, the proposed model can be deployed on various cloud-based systems for prediction tasks and can communicate with distributed IoT sensors at large scale through APIs, thus avoiding the need for deployment on edge devices. To assess the impact of potential communication overhead, we conducted a lightweight simulation using the SensEUR dataset. Each input sample consists of 192 numerical time series values (768 bytes), along with 108 bytes of textual metadata. The size of the trained model during inference remains constant regardless of batch size: 2.13 million parameters for time series only, and 24.74 million parameters for multimodal, including 22 million from the incorporated lightweight language model. When an IoT device sends a batch of 10 input samples (approximately 8.76 KB with 7.68 KB for time series data and 1.08 KB for metadata), the corresponding simulated inference times were 4.18 seconds for multimodal input and 2.64 seconds for time series input only, including I/O time. For a batch of 100 samples, inference time increased slightly to 5.28 seconds for multimodal input and 2.95 seconds for time series only. Considering typical network latency ranges of 30 to 50 milliseconds over WiFi or LTE/5G, and 1 to 10 seconds over low-power wide-area networks (*e.g.*, LoRaWAN) [49, 50], the observed 2.33 second increase for multimodal input remains within acceptable bounds.

This framework can be adopted to facilitate major challenges of automated metadata annotation problems in open IoT platforms [2, 4, 13] and sensor metadata normalization [8, 9] in building automation systems. In a broader sense, the application of the framework can be extended to the QA (Quality Assurance) process of sensor data [10] shared over distributed networks. In addition, potential benefits of deploying this framework in various IoT contexts include: enabling automatic sensor discovery and registration in large-scale sensor data-sharing platforms such as Senshamart [2]; automatic recognition of sensor types when new sensors are added to dynamic IoT network systems [10], as commonly found in smart applications (*e.g.*, Smart Cities); and interpreting time series signals when associated metadata is ambiguous due to sensor malfunctions or communication issues [14]. Furthermore, the proposed framework can be adapted for a wide range of other application domains, such as demand prediction [35] and human mobility prediction [51], where time series data can be enriched with additional textual information to enhance the effectiveness of complex predictive modeling tasks.

However, within the current scope of the study, our deep learning framework operates in a supervised setting that relies on labeled data. Metadata quality issues or missing metadata can negatively impact the performance of the proposed framework. For example, on the TS–N dataset, we observed that the classification accuracy drops by approximately 8.5% when the proportion of training samples affected by metadata quality issues increases from 10% to 80%. To mitigate this, it is essential to perform rigorous quality assurance on IoT datasets by adhering to established IoT data quality frameworks [42, 43], with a particular focus on metadata. Furthermore, since this study focuses on supervised classification of IoT sensor data, one limitation is that the original sensor metadata cannot be fully regenerated, as would be possible with text-generation AI models. However, the sensor measurement type classification provided by our proposed model can help recover lost metadata information to a significant extent.

Additionally, the scarcity of large benchmark datasets in the IoT sensor data classification domain limits the exploration of developing large foundational learning models. These areas warrant future attention to address the current limitations.

VI. Conclusion

In this study, we propose a novel multimodal deep learning framework, referred to as DeepMetaIoT, for improved classification of IoT sensor data by incorporating both time series sensor readings and textual sensor metadata generated from heterogeneous environments. DeepMetaIoT employs deep time series embedding, LLM-based text embedding, and an effective multimodal fusion strategy. Its superiority has been extensively validated through rigorous experimental analysis against state-of-the-art machine learning and deep learning baselines. Additionally, this study introduces two new multimodal IoT sensor data classification datasets to the domain. These datasets contain more recent sensor data with diverse configurations and more representative characteristics, reflecting the complexity of heterogeneous IoT sensor data environments found in real-world applications. To address current limitations of the proposed framework, several interesting and significant directions for future work can be explored. These include adopting recent advances in generative LLM models to generate metadata annotations, enabling direct semantic mapping of sensor ontologies and alleviating the issue of limited labeled data in IoT sensor data classification. Continual learning and active learning strategies could also be investigated to reduce the dependency on supervised

models for continuous sensor data classification. Furthermore, federated learning-based strategies could be integrated into the proposed framework.

References

[1] H. Y. Teh, A. W. Kempa-Liehr, and K. I.-K. Wang, "Sensor data quality: A systematic review," *Journal of Big Data*, vol. 7, no. 1, p. 11, 2020.

[2] D. Georgakopoulos, P. P. Jayaraman, and A. Dawod, "Senshamart: A trusted lot marketplace for sensor sharing," in *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, 2020, pp. 8–17.

[3] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs, "Unlocking the potential of the internet of things," 2015. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world

[4] T. Zachariah, N. Klugman, and P. Dutta, "Thingspeak in the wild: Exploring 38k visualizations of iot data," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 1035–1040.

[5] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois, "Sosa: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics*, vol. 56, pp. 1–10, 2019.

[6] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, 2012.

[7] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "Iot-lite: a lightweight semantic model for the internet of things and its use with dynamic semantics," *Personal and Ubiquitous Computing*, vol. 21, pp. 475–487, 2017.

[8] D. Hong, H. Wang, and K. Whitehouse, "Clustering-based active learning on sensor type classification in buildings," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 363–372.

[9] D. Madithiyagasthenna, P. P. Jayaraman, A. Morshed, A. R. M. Forkan, D. Georgakopoulos, Y.-B. Kang, and M. Piechowski, "A solution for annotating sensor data streams - an industrial use case in building management system," in *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, 2020, pp. 194–201.

[10] J. Čulić Gambiroža, T. Mastelić, I. Nižetić Kosović, and M. Čagalj, "Lost in data: recognizing type of time series sensor data using signal pattern classification," *International Journal of Data Science and Analytics*, pp. 1–12, 2023.

[11] J. B. Borges, H. S. Ramos, and A. A. Loureiro, "A classification strategy for internet of things data based on the class separability analysis of time series dynamics," *ACM Transactions on Internet of Things*, vol. 3, no. 3, pp. 1–30, 2022.

[12] A. Hassani, F. Montori, K. Liao, P. D. Haghighi, P. P. Jayaraman, D. Georgakopoulos, M. D. Giosa, and A. Zaslavsky, "Inform: A tool for classification and semantic annotation of iot datastreams," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 223–228.

[13] F. Montori, K. Liao, M. D. Giosa, P. P. Jayaraman, L. Bononi, T. Sellis, and D. Georgakopoulos, "A metadata-assisted cascading ensemble classification framework for automatic annotation of open iot data," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[14] J.-P. Calbimonte, O. Corcho, Z. Yan, H. Jeung, and K. Aberer, "Deriving semantic sensor metadata from raw measurements," in *Proceedings of the 5th International Workshop on Semantic Sensor Networks*, 2012.

[15] F. Montori, K. Liao, P. P. Jayaraman, L. Bononi, T. Sellis, and D. Georgakopoulos, "Classification and annotation of open internet of things datastreams," in *Web Information Systems Engineering–WISE 2018: 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part II 19*. Springer, 2018, pp. 209–224.

[16] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[17] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.

[18] M. Van Poppel, P. Schneider, J. Peters, S. Yatkin, M. Gerboles, C. Matheeussen, A. Bartonova, S. Davila, M. Signorini, M. Vogt *et al.*, "Senseurcity: A multi-city air quality dataset collected for 2020/2021 using open low-cost sensor systems," *Scientific data*, vol. 10, no. 1, p. 322, 2023.

[19] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, pp. 107–144, 2007.

[20] M. Postol, C. Diaz, R. Simon, and D. Wicke, "Time-series data analysis for classification of noisy and incomplete internet-of-things datasets," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1543–1550.

[21] M. S. K. Inan, K. Liao, H. Shen, P. P. Jayaraman, D. Georgakopoulos, and M. J. Tang, "Deepheteroiot: Deep local and global learning over heterogeneous iot sensor data," *arXiv preprint arXiv:2403.19996*, 2024. (Accepted for Publication in MobiQuitous 2023). [Online]. Available: https://doi.org/10.48550/arXiv.2403.19996

[22] L. Chen, H. B. Gunay, Z. Shi, W. Shen, and X. Li, "A metadata inference method for building automation systems with limited semantic information," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 2107–2119, 2020.

[23] J. Gao, J. Ploennigs, and M. Berges, "A data-driven meta-data inference framework for building automation systems," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015, pp. 23–32.

[24] A. A. Bhattacharya, D. Hong, D. Culler, J. Ortiz, K. Whitehouse, and E. Wu, "Automated metadata construction to support portable building applications," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015, pp. 3–12.

[25] S. Mishra, A. Glaws, D. Cutler, S. Frank, M. Azam, F. Mohammadi, and J.-S. Venne, "Unified architecture for data-driven metadata tagging of building automation systems," *Automation in Construction*, vol. 120, p. 103411, 2020.

[26] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data mining and knowledge discovery*, vol. 31, pp. 606–660, 2017.

[27] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[28] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2022.

[29] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 459–469.

[30] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, 2024.

[31] J. Li, R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," *Advances in neural information processing systems*, vol. 34, pp. 9694–9705, 2021.

[32] 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 *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[33] W. C. Sleeman IV, R. Kapoor, and P. Ghosh, "Multimodal classification: Current landscape, taxonomy and future directions," *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–31, 2022.

[34] S. A. Abdu, A. H. Yousef, and A. Salem, "Multimodal video sentiment analysis using deep learning approaches, a survey," *Information Fusion*, vol. 76, pp. 204–226, 2021.

[35] F. Rodrigues, I. Markou, and F. C. Pereira, "Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach," *Information Fusion*, vol. 49, pp. 120–129, 2019.

[36] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[37] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5776–5788, 2020.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[40] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[41] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[42] L. Zhang, D. Jeong, and S. Lee, "Data quality management in the internet of things," *Sensors*, vol. 21, no. 17, p. 5834, 2021.

[43] D. Kuemper, T. Iggena, R. Toenjes, and E. Pulvermueller, "Valid. iot: A framework for sensor data quality analysis and interpolation," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 294–303.

[44] Z. Tang, H. Fang, S. Zhou, T. Yang, Z. Zhong, T. Hu, K. Kirchhoff, and G. Karypis, "Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models," *arXiv preprint arXiv:2404.16233*, 2024.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[46] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[47] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[48] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan, "A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 466–10 485, 2024.

[49] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[50] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2019.

[51] J. Wang, X. Kong, F. Xia, and L. Sun, "Urban human mobility: Data-driven modeling and prediction," *ACM SIGKDD explorations newsletter*, vol. 21, no. 1, pp. 1–19, 2019.

Muhammad Sakib Khan Inan holds a Bachelor of Science (B.Sc.) in Computer Science and Engineering with distinction (honours). He is currently a Ph.D. student at Deakin University, funded by a prestigious Australian Research Council (ARC) Discovery Project. His research focuses on developing novel deep learning algorithms for time series IoT sensor data analytics. As a principal author, he has produced several original research works in the domains of artificial intelligence, addressing cross-disciplinary challenges.

Kewen Liao holds a PhD in Computer Science from The University of Adelaide, awarded in 2014. He is currently an Associate Professor in Data Analytics and Machine Learning at Deakin University, where he serves as a Chief Investigator on an Australian Research Council Discovery Project and is a key member of the Smart Sensing, Coding, and Analytics Lab within the Centre for Internet of Things ECOsystems Research and Experimentation (CITECORE). He previously held research fellow positions in Algorithms and Data Analytics at The University of Melbourne and Swinburne University of Technology. He has published in premier venues such as VLDB, ICDE, CVPR, AAAI, IJCAI, WWW, CHI, WSDM, CIKM, ICDCS, ICSOC, and IoT-J. He recently served as the conference Program Chair of AusDM 2024 and Area Chair of KDD 2025. His research interests include algorithm design and analysis, data science and machine learning, and IoT sensor data analytics.

Haifeng Shen is currently a full professor of information technology at the Faculty of Science and Engineering at Southern Cross University in Australia. Before that, he was an associate professor in Information Technology at Australian Catholic University, a senior lecturer in computer science at Flinders University in Australia, and an assistant professor in computer science at Nanyang Technological University in Singapore. His research covers software engineering, artificial intelligence, human-centred computing, and software and application security. He has published over 140 research papers at peer-reviewed international conferences and journals, including premier venues such as TSE, ICSE, ASE, ICDE, TOCHI, CHI, and CSCW. He founded the HilstLab (Human-centred Intelligent Learning and Software Technologies Research Lab), which develops innovative software systems to solve real-world problems and studies their potential impacts on humans. He is a senior member of both ACM and IEEE and has chaired several international and national conferences.

Prem Prakash Jayaraman is a Professor in Internet of Things and Distributed Systems and Director of Swinburne's Factory of the Future and Digital Innovation Lab. He leads industry-funded groundbreaking research in Internet of Things (IoT), Mobile and Cloud computing to co-create novel solutions underpinned by digital technologies solving industry problems and aiding their digital transformation journey. He has received over $18 Million in competitive external research funding from both industry and Australian Research Council to fund his research. He has (co) authored 150+ journal, conference and book chapter in highly ranked venues the above research areas. Contact: www.premjayaraman.com

Federico Montori received the B.S. and M.S. degrees (summa cum laude) in computer science and the Ph.D. degree in computer science and engineering from the University of Bologna, Italy, in 2012, 2015, and 2019, respectively. He was a Visiting Researcher at Swinburne University of Technology (Australia), Luleå Tekniska Universitet (Sweden), and Technische Universität Ilmenau (Germany). He is currently a Senior Assistant Professor at the University of Bologna. He participated in several EU projects and was WP Leader for the H2020 Project Arrowhead Tools. His primary research interests include mobile crowdsensing (MCS), pervasive and mobile computing, IoT automation, and data analysis for IoT scenarios.

Dimitrios Georgakopoulos is the Director of the "Australian Research Council's Industrial Transformation Research Hub for Future Digital Manufacturing" and Computer Science Professor at Swinburne University of Technology. Before that he was Research Director of CSIRO's ICT Centre, where he led Australia's largest Computer Science research program, and Professor at RMIT University. Before that Dimitrios held senior research and management positions in leading industrial laboratories in the USA, including Telcordia Technologies; Microelectronics and Computer Corporation (MCC); GTE Laboratories (currently Verizon); and Bell Communications Research (Bellcore). Dimitrios has authored 325 articles in IoT, AI, process automation, and data management that have received approximately 24,100 citations. According to Elsevier's science-wide author databases of standardized citation indicators, he is in the 2% of top-cited authors globally (top 1% if self-citations are excluded) in both career and 2024 rankings. His research has attracted $77M of external research grants/contracts. He also led major cross-disciplinary research initiatives that received over $100M funding from the Australian and USA governments. Dimitrios has received 19 awards that include, the "2023 National iAward for Government & Public Sector Solution of Year" from the Australian Information Industry Association (AIIA), several best paper awards in CORE A*/A conferences, and various research impact awards.