



DeepHeteroIoT: Deep Local and Global Learning over Heterogeneous IoT Sensor Data

Muhammad Sakib Khan Inan¹, Kewen Liao^{1(✉)}, Haifeng Shen¹,
Prem Prakash Jayaraman², Dimitrios Georgakopoulos², and Ming Jian Tang³

¹ Australian Catholic University, North Sydney, NSW, Australia
muhammadsakibkhan.inan@myacu.edu.au, {kewen.liao,haifeng.shen}@acu.edu.au

² Swinburne University of Technology, Hawthorn, VIC, Australia
{pjayaraman,dgeorgakopoulos}@swin.edu.au

³ Atlassian, Sydney, NSW, Australia

Abstract. Internet of Things (IoT) sensor data or readings evince variations in timestamp range, sampling frequency, geographical location, unit of measurement, etc. Such presented sequence data heterogeneity makes it difficult for traditional time series classification algorithms to perform well. Therefore, addressing the heterogeneity challenge demands learning not only the sub-patterns (local features) but also the overall pattern (global feature). To address the challenge of classifying heterogeneous IoT sensor data (e.g., categorizing sensor data types like temperature and humidity), we propose a novel deep learning model that incorporates both Convolutional Neural Network and Bi-directional Gated Recurrent Unit to learn local and global features respectively, in an end-to-end manner. Through rigorous experimentation on heterogeneous IoT sensor datasets, we validate the effectiveness of our proposed model, which outperforms recent state-of-the-art classification methods as well as several machine learning and deep learning baselines. In particular, the model achieves an average absolute improvement of 3.37% in Accuracy and 2.85% in F1-Score across datasets.

Keywords: IoT Sensor Data · Data Heterogeneity · Deep Learning · Time Series Classification

1 Introduction

IoT sensor data¹ or readings exhibit time series-like sequences. The classification of such sequence data has emerged as an in-demand research area, given the ubiquitous presence of IoT sensors in our daily lives in accordance with the growing dominance of mobile computing [10, 20, 35]. The next decade is predicted

¹ Aligning with the works of [5, 21, 22], we define IoT sensor data as sequences of raw numerical readings/observations obtained from IoT sensors over time.

to see IoT devices creating \$14.4 trillion worth of value across different industries [3]. IoT sensor devices are widely adopted in the sectors like healthcare, environmental monitoring, traffic management, and energy management, in a diverse range of settings [5, 21, 22]. This naturally causes the IoT data to vary in timestamp range, observation configurations, sampling frequency, unit of measurement, and geo-location [30], which leads to the production of a plethora of IoT sensor data with inherent heterogeneity [2, 9, 21]. The inherent heterogeneity, where sensors of the same type (e.g., temperature) record data in different units of measurement (Celsius or Fahrenheit) or with varying sampling frequencies, results in inconsistent and non-repetitive complex discriminatory patterns in time-series readings from IoT sensors. In such scenarios, time-series sensor readings/sequences display a mixture of local sub-patterns occurring at irregular intervals, forming a global pattern of heterogeneous nature at times.

The growing use of IoT sensors, integrated with a variety of mobile computing devices, highlights the importance of accurate classification, such as distinguishing between temperature and humidity data, to ensure the reliability of dynamic systems like smart homes [7]. The heterogeneous nature of IoT sensor data also affects the data quality [10], making the classification tasks challenging even for the existing machine-learning techniques. It also hinders the effective re-purposing of these data for interoperability [20]. For instance, the traditional time series classification methods often underperform [1, 22] when it comes to classifying IoT sensor data in a multi-class setting [1, 2].

Recent research has witnessed the development of machine learning (ML) algorithms that leverage textual metadata (e.g., sensor names and descriptions) to ameliorate the classification of IoT sensor data [19, 21, 22]. However, such textual metadata are often not available or presented inaccurately in IoT sensor datasets [2]. The expansion of IoT systems also increases the chance of significant metadata loss due to disrupted or corrupted connections to configuration metadata, making collected IoT sensor data unusable [5]. Moreover, depending on the availability of accurate metadata limits the capability and feasibility of learning where only numeric sensor readings are available [2, 4]. Though optimized ensemble machine-learning algorithms [21, 22] and feature transformation-based strategies [2] recently achieved notable successes in heterogeneous IoT sensor data classification, those approaches lack generalizability and scalability and they are also not deeply learned end-to-end processes [14]. Along with the rapid advancement of artificial intelligence, deep learning (DL) algorithms have demonstrated massive potential in time series classification [14] due to their advantage of representation learning. Some studies [6, 26] developed DL-based solutions for IoT sensor data classification but only considered a homogeneous data domain. There also have been studies that incorporated deep learning-based strategies for IoT sensor data forecasting [24, 28]. However, there were no DL models specifically designed for heterogeneous IoT sensor data classification. In search of such a DL solution, we first rigorously investigated the state-of-the-art deep time series classification algorithms including InceptionTime [15] and TapNet [34], along with other traditional machine learning and deep learning based algorithms. From our empirical study, we found that in terms of the heterogeneous

IoT sensor datasets, existing solutions do not perform well which was similarly highlighted in previous studies [1, 2]. Though there exist effective ensemble learning models like MACE [21] and TKSE [22], these models are highly complex, easily overfit, and do not learn the underlying artifacts of heterogeneous IoT sensors with a contextual-learned approach.

To this end, we propose a novel end-to-end deep learning model with the aim of better capturing the heterogeneous patterns of IoT sensor data that would lead to enhanced classification performance. The model learns ensembled *local* neighborhood-based features with Convolutional Neural Network (CNN) convolution layers of varying kernel sizes, and *global* whole series sequential features with a stack of Bi-directional Gated Recurrent Units (GRUs). We deliberately name the model as DeepHeteroIoT for ease of reference. The main contributions of the work are summarized as follows:

- We propose the first end-to-end DL model for heterogeneous IoT sensor data classification. The model facilitates a novel combination of CNN and Bi-directional GRU modules to respectively learn local and global patterns of heterogeneous IoT sensor data.
- We provide rigorous empirical comparisons across multiple IoT datasets along with the development of a new heterogeneous IoT sensor dataset. Experimental results demonstrate that our deep learning model achieves state-of-the-art classification performance in both accuracy and F1-score.

2 Related Works

IoT Sensor Data Classification. IoT sensors typically generate numerical continuous data, forming numerical sequences when collected at regular or irregular intervals over a specific time period. The task of classifying these continuous numerical series is akin to problems such as Time Series Classification (TSC) or Sequence Classification. To deal with challenging heterogeneous IoT sensor data, Montori et al. [22] proposed a two-layer sequential ensemble approach of classifiers where individually trained classifiers are stacked during the prediction pipeline in a sequential manner to filter out classes with fewer prediction probabilities. In their extended version [21], an improved multi-layer sequential ensemble model was proposed with novel heuristics for selecting and ordering classifiers and filtering classes between classifiers. Borges et al. [2] proposed a transformation-based classification strategy that converts raw sensor data to an ordinal pattern with improved feature representation and class separability. Postol et al. [25] proposed a topological data analysis-based strategy for noisy IoT sensor data classification. A random forest-based strategy was recently proposed to classify a signal type being read from an IoT sensor utilizing raw IoT sequence data [5]. Moreover, a probabilistic data mining approach [4] incorporating slope distribution computation via linear approximation of time series sequence was first developed for heterogeneous IoT sensor data classification. Although previous approaches achieved notable classification performance, they have only sought global patterns from different perspectives. Several studies have

explored the potential of deep learning for classifying IoT sensor data for tasks such as intelligent waste management [26] and human activity recognition [6]. However, the sensors used in these domains are predominantly homogeneous and share similar characteristics. To the best of our knowledge, the development of end-to-end deep learning algorithms to better capture rich low-dimensional temporal semantics of heterogeneous IoT sensor data (that comes from diverse environments) is still missing in the literature.

Time Series Classification. The time series classification domain has seen many developments over the years such as the classical methods of Dynamic Time Warping (DTW) and Weighted Dynamic Time Warping [1] that rely on whole-series distance-based similarity measures. Additionally, methods that work with the approach of subsequence extraction, coined as “shapelets” based strategies, have been popular in data mining [27,33]. However, to deal with the dimensionality issue present in the distance measure-based approaches, a lower-bound symbolic representation of time series sequence named Symbolic Aggregation approXimation (SAX) [17] was introduced to first transform data via Piecewise Aggregate Approximation (PAA) and then symbolize the PAA representation into a discrete string. Lin et al. [18] proposed a histogram-based high-level feature extraction strategy utilizing PAA through SAX representation for time series classification, taking inspiration from the popular “bag of words” strategy in the domain of text mining. Nevertheless, the majority of conventional techniques concentrate on individual time series attributes like shape or frequency. These methods lack scalability and efficacy in handling the diverse characteristics inherent to the IoT data [21]. In recent years, learning-based methods that incorporate state-of-the-art deep learning algorithms have been developed for time series classification due to their capability of learning low-dimensional feature representations automatically without domain knowledge [14]. Convolutional Neural Network (CNN) based model InceptionTime [15] and Attention-mechanism incorporated hybrid approach TapNet [34] are two prominent examples of such effective deep learning-based solutions for time series classification. Additionally, ensemble or hybridization of deep learning algorithms including CNN and recurrent neural networks (RNN) have shown great potential in extracting rich semantics of time series in terms of sensor data classification [13] and forecasting challenges in various domains such as water level prediction [24] and short-term residential load forecasting [28]. While current state-of-the-art time series classification algorithms exhibit strong performance on general time series data, they often lack the specialized design required to effectively capture the inherent heterogeneity within IoT sequence data. As a consequence, there is a critical need to develop novel methodologies that can accommodate the varying characteristics of IoT sequence data and extract meaningful insights from these distinct and heterogeneous sources.

3 DeepHeteroIoT: Our Proposed Deep Learning Model

In this section, we delineate a brief description of the key components of our proposed end-to-end deep learning model. As represented in Fig. 1, our proposed deep learning model, DeepHeteroIoT, extracts learned feature representation of global patterns using a stack of bi-directional GRU and local patterns using an ensemble of decoupled CNN stacks with varying kernel sizes. Learned local and global feature vectors are then combined through a concatenation layer which connects to a multilayer perceptron (MLP) head for final classification.

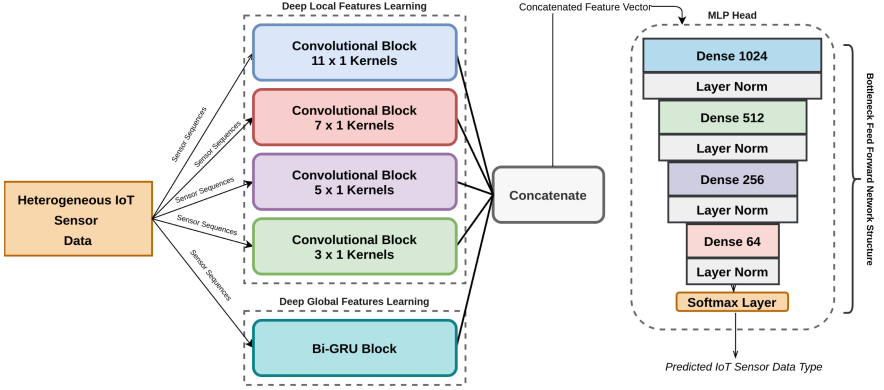


Fig. 1. An overview of our proposed deep learning model with learnable local and global learnable patterns for heterogeneous IoT sequence data classification

Deep Local Features. Initially developed to tackle complex issues within Computer Vision, Convolutional Neural Networks (CNNs) leverage local neighborhood-based features to enhance the distinction between different classes by shaping decision boundaries. As the realm of deep learning continues its rapid evolution, CNN architectures have showcased significant potential in extracting robust spatial features from input sequences. By employing convolutional operations with small kernel sizes, CNNs excel at capturing local intricacies, making them an excellent fit for intricate time series or sequence analyses. In our proposed deep learning model, we introduce a customized novel CNN architecture tailored to seamlessly integrate into our end-to-end framework for IoT sensor data classification. A single convolutional block incorporated in our proposed model can be expressed mathematically;

$$F_f = \text{ConvBlock}(X, f, p) \quad (1)$$

In Eq. 1, X represents a vector containing input sequences such as $[X_0, X_1, \dots, X_t]$, distributed over t timestamps, f indicates kernel size for convolutional layers, p indicates the type of padding to be incorporated and F_f is the feature

space computed base on kernel size f . Also, each *ConvBlock* consists of a stack of convolutional layers with designated kernel size and pooling layers. Each set of convolutional blocks comprises nine consecutive 1D-convolutional layers. Following the 3rd and 6th convolutional layers, a max-pooling layer with a size of 2 is incorporated. Each convolutional block ends with a Global Average Pooling layer [31]. The pooling operations allow the models to downsample feature maps in the way the most significant information has been retained and passed to the next layer while reducing the spatial dimensions. Within each convolutional block, the sequence of stacked convolutional layers is set up with 128 filters for the layers preceding the initial max-pooling stage, and subsequently, 64 filters are employed for all successive convolutional layers.

The size of the kernels used in convolutional layers is of utmost importance for capturing local features, as they function as windows to compute features on sub-patterns extracted from the entire time series like sequences at a lower level. In Algorithm 1, local deep learned features are computed at steps 2,3,4 and 5 incorporating X into Eq. 1 for varying kernel sizes with designated padding strategy. Drawing inspiration from the idea of incorporating kernels (convolutional windows) of different sizes to enhance the extraction of local features from different spatial ranges [29], we propose a decoupled ensemble structure for our convolutional module. In our design, we stack convolutional layers with kernel sizes of 3, 5, 7, and 11, allowing our model to learn sub-patterns using various receptive field sizes in separate blocks. These small kernel sizes were adopted on the basis of their proven computational efficiency in prior state-of-the-art CNN models [29], as well as their effectiveness in capturing local features in complex computer vision problems. The final choice of kernel sizes was also determined from randomized parameter tuning. Additionally, padding plays a critical role in the design of CNNs as it ensures the preservation of spatial dimensions in input sequences. By retaining spatial information, padding enables CNNs to process time series alike sequences with varying lengths consistently. For the convolutional layers, we use 'causal' padding. This type of padding strategy adds zeros at the start of the input sequence after each convolution operation. Importantly, this maintains the temporal sequence order, ensuring that $output_t$ does not depend on $input_{t+1}$ to $input_{t+n}$ [23].

Deep Global Features. Gated Recurrent Unit (GRU), one type of Recurrent Neural Network (RNN), has emerged as a powerful deep learning technique for capturing overall long temporal dependencies or global patterns in sequential data for challenges like time series classification [8, 32]. Though GRU possesses a simpler architecture with fewer gates and parameters to train, it still excels at capturing both short-term and long-term dependencies in sequential data compared to Long Short-Term Memory (LSTM) [32]. These unique traits make it computationally more efficient and potentially better suited for our IoT Sensor data classification challenge. A GRU unit maintains a gating mechanism incorporating an Update Gate (u_t) and Reset Gate (r_t) based strategy to selectively update and reset its internal state for capturing long-term dependencies

and global facets or features of sequence data along with solving the vanishing gradient problem. Mathematically,

$$u_t = \sigma(W_u \odot x_t + U_z \odot h_{t-1} + b_u) \quad (2)$$

$$r_t = \sigma(W_r \odot x_t + U_r \odot h_{t-1} + b_r) \quad (3)$$

Here, t denotes the timestamp, x_t , and h_{t-1} accordingly denotes the input in the current timestamp and hidden state in the previous timestamp. W , U and b are respective weight matrices and vectors for update and reset gates. To better capture the global sequential patterns of IoT sensor data, we develop a stacked Bi-directional GRU module for integration into our deep learning model. Bi-directional means capturing the patterns of the sequence data from both directions (forward and backward) to allow it to learn temporal dependencies and patterns that exist in both past and future contexts. A single block of bi-directional GRU unit, incorporated into our deep learning model can be represented mathematically as;

$$gru_i = Bidirectional(GRU(X, D)) \quad (4)$$

Here, X represents a vector containing input sequences such as $[X_0, X_1, \dots, X_t]$, distributed over t timestamps, and D represents dimensions of output representation space computed by the GRU function. gru_i is the output sequence returned by one *Bidirectional* function that concatenates the computed output features by the GRU function from both the forward and backward direction in an appending manner. As described in Fig. 1 and Algorithm 1, for deep global sequential feature extraction we stack 3 Bi-directional GRU units accompanied by Batch Normalization after each unit. The hyperparameter setting of parameter D for each unit is accordingly 128, 128, and 64 in steps 7, 8, and 9 for Algorithm 1 with the incorporation of Eq. 4 for computation of global patterns at each epoch for every sample in X .

Combined Deep Learning Model. To improve the classification performance on challenging IoT data, as delineated in Algorithm 1, we propose a novel deep learning model that incorporates deep learning-based local features by our customized CNN and global features by our Bi-directional GRU. As depicted in Algorithm 1, the learned feature space at every epoch from both of these networks is concatenated through a concatenation layer in step 10 (Algorithm 1) and fed into an MLP Head (Multi-layer perception-based fully-connected layer) in step 11 (Algorithm 1) which consists of a stack of dense feed-forward layers accompanied by layer-normalization after each dense linear projection layer. The stack of dense linear projection layers follows a bottleneck design (accordingly 1024, 512, 256 and 64) in terms of the number of neurons in each layer inspired by the previous state-of-the-art architectures [12]. The key motivation for this type of design is to reduce the number of parameters in subsequent layers to force the network to learn more compact representations that eventually improve generalization reducing the chance of overfitting [12]. The number of *Epochs* was set to 200 using ADAM (Adaptive Moment Estimation) optimizer with a learning rate

of 0.001. The weights of the best epoch on the validation stage were considered for the best model.

Algorithm 1. DeepHeteroIoT: Our Proposed Deep Learning Model

Input: \mathbb{X} : 2-D matrix that contains N number of individual sequences represented as x_i where IoT Sensor data sequences is distributed of t time-stamps. $\mathbb{X} = [x_i, x_{i+1}, \dots, x_{i+N}]$. Each x_i contains a sequence of data points representing IoT sensor readings.. Every x_i in \mathbb{X} feeds as input to the neural network over the epochs into individual steps.

- 1: **for** Each *epoch* in *Epochs* **do**
- 2: $F_3 = \text{ConvBlock}(X, 3, \text{"causal"})$
- 3: $F_5 = \text{ConvBlock}(X, 5, \text{"causal"})$
- 4: $F_7 = \text{ConvBlock}(X, 7, \text{"causal"})$
- 5: $F_{11} = \text{ConvBlock}(X, 11, \text{"causal"})$
- 6: Concatenate F_3, F_5, F_7, F_{11} in an appending manner which generates a vector concat_{local} , containing a matrix of local features.
- 7: $\text{gru}_1 = \text{Bidirectional}(\text{GRU}(X, 128))$
- 8: $\text{gru}_2 = \text{Bidirectional}(\text{GRU}(\text{gru}_1, 64))$
- 9: $\text{gru}_3 = \text{Bidirectional}(\text{GRU}(\text{gru}_2, 64))$
- 10: Concatenate extracted local feature (concat_{local}) and global features (gru_3) in an appending manner.
- 11: Feed the concatenated feature space into the MLP Head.
- 12: Compute softmax predictions.
- 13: Compute loss of training epoch.
- 14: Update weights of each layer based on the learning rate.
- 15: **end for**

Output: \mathbb{O} : predicted class labels defining IoT sensor data

4 Experiments

In this section, we outline the outcomes of our model's experiments using benchmark IoT datasets [21, 22]. To evaluate the efficacy of our novel deep learning model, we contrasted its Accuracy and F1 scores with conventional machine learning and deep learning models. We also compared it to time series classification-centric deep learning models like InceptionTime [15] and TapNet [34]. Furthermore, we conducted a comparison between our model and a contemporary state-of-the-art ensemble machine learning model known as MACE (as presented in [21]), which was explicitly tailored for classifying IoT data in a prior research endeavor. We omitted comparisons with feature-transformation or data-mining oriented methodologies [2] for classifying IoT sensor data. Because, these techniques rely on statistical transformations of IoT sequence data and our study emphasizes a comprehensive exploration of the capabilities of ML and DL methods in capturing heterogeneous patterns within IoT sequence data at a more granular level [14]. To ensure a consistent comparison within our domain

and alignment with the methodology in the prior study [21], we adopted the identical experimental design, including the division of datasets into training and testing subsets. We performed a stratified split over the datasets assigning 70 % to the training set and 30 % to the testing using the Scikit-Learn library by setting parameter value “random_state=100”. The tested classifiers in this paper are implemented in Python 3.9.16 on top of Scikit-Learn for machine learning models (except XGBoost) and Tensorflow for deep learning models. For XGBoost, we utilize the Python implementation of the XGBoost algorithm. All the experiments were conducted on a Linux server running Ubuntu 20.04 LTS as the OS, equipped with a 24-core Intel CPU (x86_64) and 128 GiB of RAM. In our work, we include the Swiss Experiment dataset and Urban Observatory dataset from MACE [21] paper. We exclude the ThingSpeak dataset which was included in the MACE [21] paper as this dataset contains textual meta-data describing IoT sensors which do not go with the motivation of this study of dealing with IoT sensor classification challenge utilizing only raw numeric sensor observations. Additionally, to validate the generalizability of our proposed solution, we constructed another dataset named IOWA ASOS, inspired by a recent study [2] which will be discussed in detail in the remaining sections.

4.1 IoT Sensor Datasets

In this section, we delineate the three benchmark IoT Sensor datasets that have been utilized in this study including the IOWA ASOS dataset that we have developed specifically for this paper to prove the generalizability of our model performance along with Urban Observatory (UrbObs) and Swiss Experiment (Swiss) datasets from previous studies [4, 21].

- **IOWA ASOS (IOWA).** Inspired by the study of [2], we developed this IoT sensor dataset by collecting sensor data of various meteorological sensors from the data archive of Iowa Environmental Mesonet (IEM) prepared by the IOWA State University and made available via API and web interface² This dataset contains sensor data from meteorological sensors at airports using Automated Surface Observing Systems (ASOS)³ These sensors can produce observations every minute or every hour based on the requirements of the airport authority to corroborate aviation operations by using weather forecasting reports. To make our own version of the new IoT dataset for this study, we collect data with 1-h intervals over a 6-month time-span from 5 different stations in the IOWA state of the United States. To make this dataset suitable for an IoT Sensor data classification challenge, we include 1-week data at 1-h intervals for each individual classification sample. The dataset contains 8 class labels and these are: Air Temperature, Dew Point Temperature, Relative Humidity, Wind Direction, Pressure Altimeter, Visibility, Wind Gust,

² Iowa Environmental Mesonet ASOS-METAR Data Download: <https://mesonet.agron.iastate.edu/ASOS/>.

³ Automated Surface Observing Systems from the U.S. National Weather Service: <https://www.weather.gov/asos/asotech..>

and Apparent Temperature (Heat Index). The missing values are imputed by the standard imputation strategy [2] of replacing by global average (mean) on that time-stamp across the individual class labels.

- **Urban Observatory (UrbObs)**. The version of this dataset we utilized for our experimentation was developed in a previous study [21] of IoT sensor data classification. We have incorporated the exact same version of this dataset to make our experimental results comparable to the literature. This dataset contains sensor data generated through an innovative initiative⁴ spearheaded by the University of Newcastle. The program aims to establish an urban sensor network in Newcastle, United Kingdom (UK) and provides publicly accessible real-time environmental data [16]. The dataset has 16 different class labels including NO_2 (Nitrogen Dioxide), Wind Direction, Humidity, Wind Speed, Temperature, Pressure, Wind Gust, Rainfall, Soil Moisture, Average Speed, Congestion, Traffic Flow, Journey time, Sound, CO (Carbon Monoxide) and NO (Nitrogen Monoxide). This dataset contains highly correlated IoT data, primarily due to its city-wide scope and originating from a single source [21].
- **Swiss Experiment (Swiss)**. This dataset is one of the few extremely noisy and heterogeneous IoT datasets that consist of data from various microscopic locations in various time-stamps within the Swiss Alps Mountain range [4]. This dataset contains data from sensors having a diverse range of sampling rates making the phase-shift of data series very significant [4, 21]. Following the exact same version of this dataset from the previous study [21], we cut each time series data to the length of the shortest stream as the original data contains data of slightly different lengths. It includes class labels such as CO_2 (Carbon Dioxide), Humidity, Lysimeter, Moisture, Pressure, Radiation, Snow Height, Temperature, Voltage, Wind Speed, and Wind Direction.

Table 1. Summary of IoT Datasets

Dataset	Length	Duration	Samples	Labels
Urban Observatory	864	1 day	1065	16
Swiss Experiment	445	Variable	346	11
IOWA ASOS	168	1 week	1000	8

Table 1 provides a summary of 3 IoT datasets highlighting time series sequence length, duration of record per sample, the total number of samples, and the total number of class labels for each dataset. Out of these three IoT datasets, the Swiss Experiment dataset exhibits a notable class imbalance and has a very limited number of training samples, making it a particularly challenging dataset for deep learning methods. On the other hand, the IOWA ASOS dataset is almost completely balanced where each class has almost the same amount of data samples. The Urban Observatory exhibits some degree of class

⁴ Urban Observatory Environment Program: <http://newcastle.urbanobservatory.ac.uk/>.

imbalance, but due to the presence of less noisy IoT sensor data, the imbalance issue does not pose a major problem for this dataset [21]. In the Swiss Experiment, the majority class has 78 samples, while the minority class has only 14 samples in total.

4.2 Experimental Results

In this section, a detailed representation of experimental results along with a rigorous comparison against baseline machine learning and deep learning models have been delineated. Also, performance comparison with previous state-of-the-art models across 3 datasets is presented and validated.

Evaluation Metrics. Before starting, we describe the evaluation metrics that have been computed and compared throughout the experimental validation stage for our proposed model and all other experiments. These are:

- *Accuracy*: It indicates the percentage of IoT Sensor data that have been classified correctly by our model out of total data samples.
- *F1-Score*: It is a harmonic mean of Precision and Recall which gives us a balanced perception of our model’s performance and helps to validate DL model’s effectiveness properly.

For the evaluation of each experiment, we focus on the Accuracy and weighted average F1-Score throughout the study. Later, we computed the macro-average F1 Score for comparison of the model’s effectiveness with a previous state-of-the-art study [21]. The weighted average F1 Score is a more suitable way to compute F1-Scores for scenarios where the class imbalance is present to find the dataset. On the other, the macro average F1-Score does not take into account that class imbalance is present in the dataset.

Ablation Studies. To validate the efficacy of our proposed combination of global and local deep learning-based learned patterns for IoT sequence data, we illustrate a detailed ablation analysis of our proposed model in Table 2 with Accuracy scores and weighted average F-1 scores across three different IoT datasets. From the results of Table 2, it is clearly visible that our proposed novel deep-learning model outperforms individual components including only global features, only local features, and only MLP Head (without any deep feature extraction) based classifiers with a high margin. However, out of individual components, classification with only global features dominates compared to combinations with only local features and only MLP Head across 3 datasets.

It is also noticeable that only MLP Head performs badly, indicating the importance of deep-learning-based feature extraction before training a fully-connected network-based classifier for the classification of heterogeneous IoT sensor data. If we draw attention to the accuracy scores presented in Table 2, we can notice that our proposed deep learning model outperforms the second-best combination in Table 2 by 3.75%, 6.73% and 1.68% accordingly on Urban Observatory, Swiss Experiment and IOWA ASOS datasets. Similarly, in terms

Table 2. Ablation study of our proposed deep learning model across IoT datasets

	UrbObs (Accu- racy)	UrbObs (F1-Score)	Swiss (Accu- racy)	Swiss (F1-Score)	IOWA (Accuracy)	IOWA (F1-Score)
Global Features	90.63%	89.92%	79.81%	78.5%	94.33%	94.32%
Local Features	88.13%	87.72%	77.88%	76.41%	87.33%	87.50%
MLP Head	77.51%	76.02%	45.19%	42.04%	86.01%	85.80%
DeepHeteroIoT	94.38%	94.27%	86.54%	85.12%	96.01%	95.93%

of weighted average F1-score our proposed model outperforms by 4.35%, 6.62%, and 1.61%. The outperforming scores achieved by our proposed model justify the importance of learning both global and local deep learning-based patterns of IoT sequence data within one combined and end-to-end learning model. As IoT sensor data is heterogeneous, the proposed combination of both local and global learnable deep neural network architecture is a more suitable approach to capture different dynamic facets of IoT sensor data.

Out of the 3 IoT datasets, as described earlier, the Swiss Experiment dataset is highly heterogeneous with a very limited number of available samples including class imbalance issues. In such scenarios, intricate deep-learning architecture like ours tends to suffer from non-convergence issues and poor training generalizability due to data scarcity issues. For this, only the Swiss Experiment dataset was pre-processed in the training stage by incorporating the time series data-augmentation method followed by oversampling by B-SMOTE [11]. Throughout our evaluation process, we also conducted experiments involving the data augmentation module on the other two datasets. However, since these datasets had a sufficient number of samples and exhibited either fewer or no significant class imbalance issues in their original training data, it was obvious that the improvements in accuracy and f1-scores were not as substantial in comparison to the Swiss Experiment dataset after applying data augmentations. And so, for later experimental results only for the Swiss Experiment dataset, we present the results that have been achieved by our model after incorporating the augmented training data. An illustrated representation of the learning curves delineating Accuracy per epoch for the training phase of our proposed deep learning model is outlined in Fig. 2. The learning curves (Fig. 2) certainly validate the training efficacy and robust learning capacity of our proposed model across 3 IoT datasets.

Comparison of Performance Against Other Models. In this stage, we compared our novel DL approach with traditional baseline ML models, including K Nearest Neighbor (KNN), Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), Extreme Gradient Boosting (XgBoost), and Support Vector Machine (SVM). To ensure baseline settings for the machine learning algorithms that have been used from Scikit-learn’s package, default parameter settings for every algorithm are utilized. For our baseline setting of DL models such as baseline RNN and baseline CNN, we accordingly incorporate the Simple

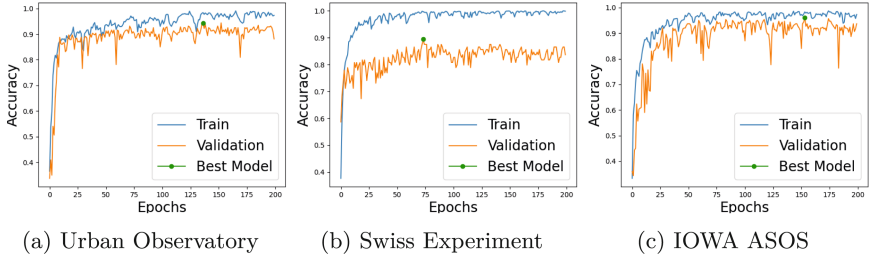


Fig. 2. Learning-curve representation for training and validation (per-epoch) for our proposed model across 3 datasets

RNN API⁵ from Tensorflow with a filter size of 32 and Conv1D API⁶ by Tensorflow and follow the traditional example⁷ of stacking Convolutional units (3 by 1 kernel sizes and “relu” with no padding) followed by a MaxPooling layer. For time series DL models such as InceptionTime [15] and TapNet [34], the sktime implementation was utilized⁸. We trained all the DL models for 200 epochs.

Table 3, depicts a rigorous comparison of our DeepHeteroIoT model against baseline ML and DL methods in terms of accuracy and weighted average F1-Score across 3 IoT datasets. Among all machine learning classifiers, only ensemble tree-based models, such as RF, GB, and XgBoost, seem to achieve competitive performance against our proposed deep learning model. Non-tree-based classifiers, such as LR, SVC, and KNN, demonstrate comparatively poor performance across all three datasets. Traditional CNN outperforms traditional RNN in UrbObs and IOWA datasets but lags behind in terms of the Swiss dataset. Notably, the popular InceptionTime model [15] exhibited very poor performance with a lower accuracy of 29.81% and an F1-score of 20.15% on the Swiss dataset. Another such model, TapNet [34], also failed to outperform the tree-based ensemble ML classifiers. The results shed light on the ineffectiveness of traditional time-series classifiers and traditional DL and ML models in heterogeneous IoT sensor scenarios, as discussed earlier in the literature review. Whereas, the potential of ensemble tree-based ML models is worth highlighting for their effective classification in heterogeneous IoT sensor scenarios. From the results, it is apparent that DeepHeteroIoT outperforms every learning-based baseline classifier by a significant margin, that is, achieving a greater average accuracy by absolute 9.29% and F1-score by 10.07 % across 3 datasets compared to the second-best performing models. Furthermore, RF emerged as the second-best model for both the Swiss and IOWA datasets, while GB outperformed RF by a slight margin on the UrbObs dataset.

⁵ https://www.tensorflow.org/api_docs/python/tf/keras/layers/SimpleRNN.

⁶ https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv1D.

⁷ <https://www.tensorflow.org/tutorials/images/cnn>.

⁸ https://www.sktime.net/en/latest/api_reference/classification.html.

Table 3. Comparison of performance against state-of-the-art baseline machine learning and deep learning models across 3 IoT datasets. The best results are in bold and the second best are underlined.

	UrbObs (Accuracy)	UrbObs (F1-Score)	Swiss (Accuracy)	Swiss (F1-Score)	IOWA (Accuracy)	IOWA (F1-Score)
CNN	78.13%	77.71%	47.11%	41.11%	83.01%	82.43%
RNN	50.31%	42.87%	58.65%	51.11%	65.01%	62.27%
InceptionTime [15]	64.98%	53.80%	29.81%	20.15%	87.98%	87.99%
TapNet [34]	53.44%	48.36%	42.31%	33.45%	55.68%	54.52%
LR	55.63%	51.75%	23.08%	15.52%	53.01%	50.41%
KNN	77.19%	73.78%	55.77%	52.36%	75.34%	74.18%
SVC	55.63%	47.98%	45.19%	33.85%	55.67%	46.95%
RF	86.56%	86.63%	73.07%	70.49%	92.00%	91.84%
GB	<u>86.87%</u>	<u>86.74%</u>	65.39%	64.53%	91.33%	91.35%
XgBoost	85.63%	85.27%	64.42%	65.36%	90.68%	90.63%
DeepHeteroIoT	94.38%	94.27%	89.42%	89.07%	96.01%	95.93%

Table 4. Comparison of performance across 3 IoT datasets with previous state-of-the-art study in IoT sensor data classification domain

	UrbObs (Accuracy)	UrbObs (macro F1-Score)	Swiss (Accuracy)	Swiss (macro F1-Score)	IOWA (Accuracy)	IOWA (macro F1-Score)
MACE (Top-k)	90.6%	87.9%	84.6%	83.2%	87.33%	87.40%
MACE (PF)	90.6%	87.9%	83.2%	83.2%	87.33%	87.40%
MACE (SoF)	90.6%	87.9%	82.7%	80.0%	90.33%	90.48%
MACE (brute-force)	92.2 %	88.7%	86.5%	84.0%	91.01%	91.04%
DeepHeteroIoT	94.38 %	92.01%	89.42%	87.34%	96.01%	95.93%

Additionally, the accuracy scores and macro-average F1-scores provided in Table 4 unequivocally demonstrate that DeepHeteroIoT consistently outperforms all alternative combinations involving the previous state-of-the-art machine learning model, MACE [21], across all datasets. Our model eventually outperforms MACE [21] by absolute 2.18 %, 2.92 %, 5 %, in accuracy and 3.31 %, 3.34 %, 4.89 % in macro-average F1-score on the respective Urban Observatory, Swiss Experiment, and IOWA ASOS datasets. It is worth noting that MACE employs an ensemble method alongside a brute-force strategy, which leads to considerable computational costs and undermines its scalability. On the other hand, our proposed deep learning model is a complete end-to-end solution which makes it scalable to the increase of dataset or problem size. For instance, the brute-force approach proposed in the MACE paper [21] took about 2 h and 17 min to train and build on the smallest dataset (Swiss), while our proposed deep learning model only required approximately 23 min and 55 s. Moreover, previous studies employing ensemble machine learning techniques [21, 22] did not incorporate the learning of heterogeneous patterns from IoT sensor data during

the training phase. Our proposed deep learning architecture fills this gap by enabling end-to-end learning of complex contextual patterns in challenging IoT sensor data sequences.

5 Conclusion

In this study, we proposed a novel deep learning model that integrates learnable local features using CNN and global features using Bi-GRU in one end-to-end network, leading to significantly improved classification of heterogeneous IoT sensor data. Through rigorous experiments across three IoT datasets, we validated that our proposed deep learning model not only outperforms baseline machine learning and deep learning methods but also achieves state-of-the-art performance in classifying heterogeneous IoT sensor data. In the future, we would like to extend our model to multi-modal IoT datasets that occur from a diverse range of mobile computing environments.

Acknowledgements. This research was funded by the Australian Research Council (ARC) under ARC Discovery Project DP220101420 and partially supported by the Australian Government Research Training Program (RTP) Allowance Scholarship (Relocation).

References

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Disc.* **31**, 606–660 (2017)
2. Borges, J.B., Ramos, H.S., Loureiro, A.A.: A classification strategy for Internet of Things data based on the class separability analysis of time series dynamics. *ACM Trans. Internet Things* **3**(3), 1–30 (2022)
3. Bradley, J., Barbier, J., Handler, D., Bradley, D.H.: The Internet of Everything is Happening Now (2013). https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoE_Economy.pdf
4. Calbimonte, J.P., Corcho, O., Yan, Z., Jeung, H., Aberer, K.: Deriving semantic sensor metadata from raw measurements (2012)
5. Čulić Gambiroža, J., Mastelić, T., Nžetić Kosović, I., Čagalj, M.: Lost in data: recognizing type of time series sensor data using signal pattern classification. *Int. J. Data Sci. Anal.* 1–12 (2023)
6. Dua, N., Singh, S.N., Semwal, V.B., Challa, S.K.: Inception inspired CNN-GRU hybrid network for human activity recognition. *Multimed. Tools Appl.* **82**(4), 5369–5403 (2023)
7. Elazhary, H.: Internet of things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: disambiguation and research directions. *J. Netw. Comput. Appl.* **128**, 105–140 (2019)
8. Elsayed, N., Maida, A.S., Bayoumi, M.: Gated recurrent neural networks empirical utilization for time series classification. In: 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1207–1210 (2019). <https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00202>

9. Georgakopoulos, D., Jayaraman, P.P.: Internet of Things: from internet scale sensing to smart services. *Computing* **98**, 1041–1058 (2016)
10. Georgakopoulos, D., Jayaraman, P.P., Dawod, A.: SenShaMart: a trusted lot marketplace for sensor sharing. In: 2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC), pp. 8–17 (2020). <https://doi.org/10.1109/CIC50333.2020.00012>
11. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
12. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
13. Interdonato, R., Ienco, D., Gaetano, R., Ose, K.: DuPLO: a dual view point deep learning architecture for time series classification. *ISPRS J. Photogramm. Remote Sens.* **149**, 91–104 (2019)
14. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. *Data Min. Knowl. Disc.* **33**(4), 917–963 (2019)
15. Ismail Fawaz, H.: InceptionTime: finding alexnet for time series classification. *Data Min. Knowl. Disc.* **34**(6), 1936–1962 (2020)
16. James, P.M., Dawson, R.J., Harris, N., Jonczyk, J.: Urban observatory environment. Newcastle University (2014). <https://doi.org/10.17634/154300-19>
17. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. *Data Min. Knowl. Disc.* **15**, 107–144 (2007)
18. Lin, J., Khade, R., Li, Y.: Rotation-invariant similarity in time series using bag-of-patterns representation. *J. Intell. Inf. Syst.* **39**, 287–315 (2012)
19. Madithiyagasthenna, D., et al.: 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), pp. 194–201 (2020). <https://doi.org/10.1109/MDM48529.2020.00042>
20. Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M.: Deep learning for IoT big data and streaming analytics: a survey. *IEEE Commun. Surv. Tutor.* **20**(4), 2923–2960 (2018)
21. Montori, F., et al.: A metadata-assisted cascading ensemble classification framework for automatic annotation of open IoT data. *IEEE Internet Things J.* (2023). <https://doi.org/10.1109/JIOT.2023.3263213>
22. Montori, F., Liao, K., Jayaraman, P.P., Bononi, L., Sellis, T., Georgakopoulos, D.: Classification and annotation of open internet of things datastreams. In: Hacid, H., Cellary, W., Wang, H., Paik, H.-Y., Zhou, R. (eds.) WISE 2018. LNCS, vol. 11234, pp. 209–224. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02925-8_15
23. van den Oord, A., et al.: WaveNet: a generative model for raw audio. In: 9th ISCA Speech Synthesis Workshop, p. 125 (2016)
24. Pan, M., et al.: Water level prediction model based on GRU and CNN. *IEEE Access* **8**, 60090–60100 (2020)
25. Postol, M., Diaz, C., Simon, R., Wicke, D.: 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), pp. 1543–1550. IEEE (2019)

26. Rahman, M.W., Islam, R., Hasan, A., Bithi, N.I., Hasan, M.M., Rahman, M.M.: Intelligent waste management system using deep learning with IoT. *J. King Saud Univ.-Comput. Inf. Sci.* **34**(5), 2072–2087 (2022)
27. Rakthanmanon, T., Keogh, E.: Fast shapelets: a scalable algorithm for discovering time series shapelets. In: *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 668–676. SIAM (2013)
28. Sajjad, M., et al.: A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. *IEEE Access* **8**, 143759–143768 (2020)
29. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826 (2016)
30. Teh, H.Y., Kempa-Liehr, A.W., Wang, K.I.K.: Sensor data quality: a systematic review. *J. Big Data* **7**(1), 1–49 (2020)
31. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: a strong baseline. In: *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585 (2017). <https://doi.org/10.1109/IJCNN.2017.7966039>
32. Yamak, P.T., Yujian, L., Gadosey, P.K.: A comparison between Arima, LSTM, and GRU for time series forecasting. In: *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 49–55 (2019)
33. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 947–956 (2009)
34. Zhang, X., Gao, Y., Lin, J., Lu, C.T.: TapNet: multivariate time series classification with attentional prototypical network. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6845–6852 (2020)
35. Zhang, Y., Suleiman, B., Alibasa, M.J.: FedGroup: a federated learning approach for anomaly detection in IoT environments. In: Longfei, S., Bodhi, P. (eds.) *MobiQ-uitous 2022*. LNICST, vol. 492, pp. 121–132. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34776-4_7