

# Internet of Things Fault Detection and Classification via Multitask Learning

Mohammad Arif Ul Alam  
University of Massachusetts Lowell  
Lowell, MA, USA  
mohammadariful\_alam@uml.edu  
University of Massachusetts Chan Medical School  
Worcester, MA, USA  
mohammadariful.alam1@umassmed.edu

## Abstract

This paper presents a comprehensive investigation into developing a fault detection and classification system for real-world Industrial Internet of Things (IIoT) applications. The study addresses challenges in data collection, annotation, algorithm development, and deployment. Using a real-world IIoT system, three phases of data collection simulate 11 predefined fault categories. We propose SMTCNN for fault detection and category classification in IIoT, evaluating its performance on real-world data. SMTCNN achieves superior specificity (3.5%) and shows significant improvements in precision, recall, and F1 measures compared to existing techniques.

## 1 Introduction

The Industrial Internet of Things (IIoT) refers to the network of interconnected devices, sensors, and systems in industrial settings that enable the collection, exchange, and analysis of data to enhance operational efficiency and productivity [1, 2]. It involves integrating physical machinery and equipment with digital technologies, enabling real-time monitoring, control, and automation of industrial processes [3]. However, within the complex IIoT ecosystem, various faults and anomalies can occur, including equipment malfunctions, communication failures, cybersecurity breaches, and data inaccuracies. Detecting and addressing these faults is crucial for maintaining uninterrupted operations, ensuring worker safety, minimizing downtime, and optimizing resource utilization [4].

Detecting and predicting anomalies in industrial environments is crucial for economic and security purposes. However, the rarity of these events poses challenges when applying existing algorithms, leading to false alarms or misdetections [6]. Previous studies conducted by Nardelli

et al. [7], Leitão et al. [8], and Oks et al. [10] have explored the modeling of IIoT networks and cyber-physical systems in industrial settings. Notably, promising solutions have been presented in reference [12]. Fault detection in industrial environments has always been challenging, with difficulties arising from device interoperability and limitations in data collection. The most promising state-of-the-art method utilizes Generative Adversarial Networks (GANs) for the detection and classification of system failures using a dataset containing missing values [9]. However, the focus of our paper is to develop a simple yet powerful unified method for the real-time detection and classification of IIoT faults using data collected from a real-world system.

To date, the existing literature on IIoT fault detection and classification has predominantly focused on either anomaly detection or fault classification for industrial processes. In this paper, we propose a novel framework called Sequential Multitask Cascaded Neural Network (SMTCNN) that integrates various approaches for anomaly detection and classification. The concept of Multitask Cascaded Neural Network (MTCNN) has gained significant popularity in face recognition research [5], where it involves three cascaded tasks: detecting the face area, detecting facial features such as eyes, nose, and mouth, and finally speeding up the face detection process. We adapt the MTCNN framework for sequential data, resulting in SMTCNN, which consists of three tasks: Change-Point Detection (CPD), Anomaly Detection (AD), and Fault Classification (FC). The framework is designed to effectively handle small fault event datasets and large normal datasets using a cascaded learning scheme. Our work makes the following key contributions:

- We developed a novel Sequential Multitask Cascaded Neural Network (SMTCNN) architecture, which incorporates three sequential tasks: change-point detection (CPD), refinement of CPD for anomaly/fault event identification, and classification of the detected faulty events using simple neural networks.
- We devised a comprehensive data collection methodology encompassing normal, abnormal, and real-world scenarios to gather IIoT data from a deployed system. Subsequently, we collected the data and

conducted an evaluation of our proposed SMTCNN model. Our experimental results demonstrate the superior performance of our approach compared to existing state-of-the-art methods, particularly in terms of specificity (3.5%).

## 2 Testbed Setting

The four IIoT devices consisted of an industrial application module and an IIoT device monitoring agent module. Two devices monitored the oven temperature, while the other two monitored smoke and humidity levels, all at a frequency of once per minute. Data from the devices were transmitted to the network gateway via WirelessHART for alarm functionality. The IIoT device monitoring agent captured hardware and firmware metrics, which were relayed to the gateway. Each device had sensors, a radio, microcontroller unit (MCU), and power supply, with the radio using WirelessHART for communication with the MCU. The gateway managed network operations, security tasks, and connected the IWSN to an IP network.

## 3 Datasets Generation

The dataset utilized in this research was created by executing the firmware of the IIoT devices for an extended duration and capturing the corresponding metrics for each function in a dedicated database. To evaluate the system's behavior in the presence of faults, deliberate firmware and hardware anomalies were introduced into the WirelessHART testbed at predetermined intervals. These anomalies were carefully designed to emulate real-world situations and assess the resilience and efficacy of the fault detection and classification algorithms. The injected faults encompassed 11 distinct fault classes that can be classified into four distinct types (Table 1).

**Table 1. Different types of IIoT faults with description**

Types	Class	Description
Under voltage faults	1	with 3.3-3.0v
	2	with 3.3-2.8v
	3	with 3.3-2.6v
	4	with 3.3-2.4v
	5	with 3.3-2.3v
	6	with 3.3-2.2v
Sensor stuck-faults by stimulating SPI	7	with VCC temperature with Clock
	8	
High temperature faults in microcontroller unit	9	under high voltages
	10	under low voltages
Buffer overflow faults	11	A buffer overflow vulnerability used for running arbitrary code

By employing the aforementioned fault generation technique, three distinct datasets were generated, each comprising time logs, energy consumption, CPU usage, and time fields (representing the duration of each instance in seconds) for every data point.

- **Anomaly Only Dataset:** To collect this small dataset, a series of continuous anomalous event were deliberately induced in the IIoT system. The dataset specifically targets 11 distinct categories of anomalies, aiming to provide comprehensive coverage of potential abnormal behaviors within the system.
- **Normal Only Dataset:** This dataset was generated in a normal operational environment of IIoT, where no known faults were present.
- **Real-time Mixed Dataset:** This dataset comprises a fully operational real-time deployed IIoT system, where randomly chosen faults from the aforementioned 11 categories were sporadically introduced (1% of the time during the day) to simulate a realistic real-world scenario.

## 4 IIoT Faults Detection and Classification

### 4.1 Unsupervised Change-Point Detection (CPD) using Simple LSTM Autoencoder

We have developed an unsupervised CPD algorithm utilizing an autoencoder that is based on the Long Short-Term Memory (LSTM) neural network. Our approach was inspired by the work of Elhalwagy et al. [11], who proposed a hybridization of unsupervised LSTM and Capsule network for online CPD in time-series data. The architecture of our autoencoder model consists of two LSTM-based encoders, one for each channel of the input data. The encoded representations from both channels are concatenated and passed to the decoder. The decoder's role is to reconstruct the input data, and during training, the model is optimized using the mean squared error (MSE) loss function and the Adam optimizer. To mitigate overfitting, we employed early stopping during training and restored the weights associated with the best performance. Subsequently, the trained autoencoder is utilized to reconstruct the time series data. Change-points are identified by establishing a threshold, denoted as  $\tau$ , which is determined based on the mean and standard deviation of the reconstruction errors. Any reconstruction error exceeding the threshold is considered a change-point. The algorithm effectively detects change-points, signifying significant variations or anomalies in the sequential data that may indicate the occurrence of faulty events in IIoT.

### 4.2 Supervised IIoT Faults Classification

Due to the limited size of the collected IIoT faults dataset, we employed state-of-the-art algorithms, excluding neural networks, for fault classification. Specifically, we utilized Random Forest (RF), Support Vector Machines (SVM), Naive Bayes (NB), Logistic Regression (LG), Decision Tree (DT), and Stochastic Gradient Descent (SGD) algorithms. To effectively apply these algorithms, we employed suitable segmentation and windowing techniques.

### 4.3 Sequential Multitask Cascaded Neural Networks (SMTCNN)

Fig. 2 shows our overall SMTCNN model architecture for IIoT Fault detection and classification.

**Problem formulation:** Let  $X$  be the input feature matrix with three columns. The binary label set  $Y_{anomaly}$  in-

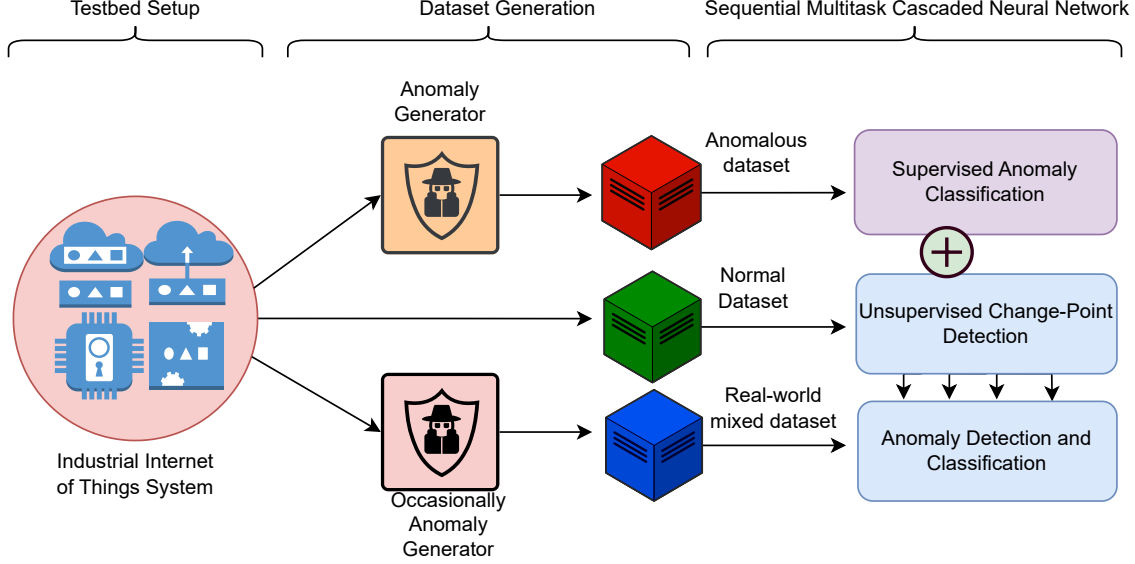


Figure 1. The schematic diagram of proposed framework

indicates the presence or absence of an anomaly, while the 12-class set  $Y_{class}$  represents different fault types, with label 12 indicating no fault and labels 1 to 11 corresponding to 11 specific fault types. We define three tasks:  $T_1$ ,  $T_2$ , and  $T_3$ , each with their respective output variables  $O_{t1}$ ,  $O_{t2}$ , and  $O_{t3}$ .

**Task 1: Anomalous segment identification:** This approach takes inspiration from the original Multitask Cascaded Neural Network (MTCNN) algorithm [5], originally designed for face recognition. Specifically, we focus on Task-1 of the MTCNN algorithm, which is responsible for detecting bounding boxes around faces. In the context of our CPD algorithm for real-time sequential data, we adapt this concept to identify the start and end points of change-points, similar to bounding boxes in the temporal domain. These identified change-points act as markers to segment the data into regions of interest that may contain anomalous events requiring further analysis. The underlying hypothesis behind this segmentation approach is that these specific regions contain valuable information related to anomalous events and merit closer examination.

**Task 2: Fine-tuning anomaly detection:** This step considers the potential segments predicted by Task 1 and aims to predict anomalies. For this purpose, we employ a simple two-layer LSTM network followed by a fully connected layer with two outputs. This network takes the original features as input.

**Task 3: Anomaly detection and classification:** This task involves the final refinement of anomaly detection and classification using the original feature set  $X$ , the output of Task 1 ( $O_{t1}$ ) which represents the segmented mask on the time-series data indicating detected change-points, and the output of Task 2 ( $O_{t2}$ ) which represents the fine-tuned detected anomalies. The core network for Task 3 is also a simple two-layer LSTM network with a fully connected

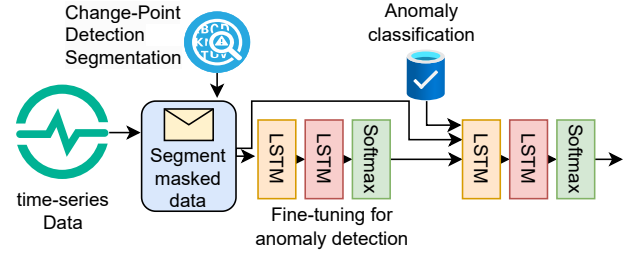


Figure 2. Proposed Sequential Multitask Cascaded Neural Network Architecture for IIoT faults detection and classification

neural network that uses softmax activation and has an output dimension of 12 (representing 11 different fault categories and one label for the existence of a fault). Thus, the final loss function is defined as follows:

$$Loss_{SMTCNN} = -\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \sum_{c=1}^C y_{i,t,c} \log(p_{i,t,c}) \quad (1)$$

In the above equation,  $N$  is the number of samples,  $T$  is the number of time steps in the sequential data,  $C$  is the number of classes,  $y_{i,t,c}$  is the one-hot encoded ground truth label for sample  $i$ , time step  $t$ , and class  $c$ , and  $p_{i,t,c}$  is the predicted probability of sample  $i$  at time step  $t$  belonging to class  $c$ .

## 5 Experimental Evaluation

The proposed SMTCNN framework was implemented using Python, TensorFlow, and scikit-learn. For our study, we collected three datasets: 8,432 data points of anomalous data, 740,448 data points of normal data, and 718,444 data points of a mixed dataset representing real-time scenarios. The unsupervised CPD algorithm was trained us-

ing the normal data only, while the supervised IIoT faults classification algorithm was trained using the anomalous data only. Finally, the SMTCNN algorithm was trained and evaluated using the mixed dataset, encompassing both normal and anomalous data.

### 5.1 Baseline

We have implemented few state-of-art algorithms as well as few versions of our proposed framework by removing few modules for proving their importance in the pipeline.

**B1 (GAN based Method):** This method [9] proposed a GAN-based anomaly detection and classification algorithm to detect and classify faults in IIoT.

**B2 (SMTCNN without CPD segmentation):** In this framework, we removed the CPD based segmentation masking, but kept all other modules.

**B2 (SMTCNN without supervised faults classification):** In this framework, we removed supervised faults classification module in the SMTCNN entwork, but kept every other modules.

### 5.2 Results

We employed the traditional 10-fold cross-validation technique to evaluate the supervised segmented IIoT fault classification. However, as traditional 10-fold cross-validation is not suitable for sequential data, we adapted our approach. To train and assess the performance of our proposed sequential algorithm, SMTCNN, we divided the entire sequential data into two halves. Subsequently, we randomly selected a sequence of data from the first half for training purposes and another sequence of data from the second half for testing. The lengths of the training and testing sequences were also randomly chosen within a range of 50% to 80% of the available data samples for each fold. This process was repeated 10 times to generate 10 different sets of training and testing sequences. We evaluated our proposed algorithm's performance using metrics such as balanced accuracy, precision, recall, specificity, and F1 measure. Additionally, we calculated the standard deviation of these metrics to assess overfitting. The performance details of the baselines and our overall framework are provided in Table 2, clearly demonstrating the superior performance of our framework compared to the baseline algorithms.

We observed that our framework exhibits marginal improvements in terms of accuracy, precision, recall, and F-1 measure compared to the baseline algorithms. However, the improvement in specificity is particularly noteworthy, with a significant increase of 3.5% over the baseline (B1). This enhancement represents a substantial advancement in the field of IIoT fault detection research. A closer examination of Table 2 reveals that the performance of SMTCNN without change-point segmentation (B1) and without supervised fault classification pretraining (B2) experiences a significant drop in performance.

### 5.3 Conclusions and Future Works

This paper presented an initial step towards the collection, simulation, and prediction of faults in real-world

**Table 2. Performance comparisons of our proposed method with baseline algorithms**

	Accuracy	Precision	Recall	Specificity	F1-measure
B1	96.9±.02	96.6±.02	96.9±.02	96.1±.01	97.4±.02
B2	94.5±.05	96.3±.03	94.5±.05	96.8±.02	96.5±.01
B3	95.1±.04	94.6±.05	95.1±.04	97.1±.03	97.1±.01
Our	97.9±.01	96.8±.02	97.9±.01	98.6±.01	98.1±.01

deployed IIoT systems. It was a part of our broader vision to develop a secure, fault-tolerant, and reliable smart and connected industrial infrastructure. The primary objective of this study was to release our preliminary collected datasets for the advancement of IIoT fault generation and the integration of multitask learning in fault detection. However, it was essential to acknowledge that automatic fault detection encompasses a wide range of use cases that were not specifically addressed in this research. However, in real-world scenarios, IIoT systems often encounter a significant number of missing values resulting from internet connectivity issues or power interruptions, which were not explicitly tackled in this study. However, in this paper, we did not consider the edge computing which also poses future challenge that will be addressed.

### 6 References

- [1] Boyes et. al. (October 2018). "The industrial internet of things (IIoT): An analysis framework". *Computers in Industry*. 101: 1–12.
- [2] Brauner et. al. "A Computer Science Perspective on Digital Transformation in Production". *ACM Transactions on Internet of Things*. 3 (2): 15:1–15:32 2022.
- [3] Gilchrist, Alasdair (2016). "Industry 4.0 - the industrial internet of things". *Apress Media*. doi:10.1007/978-1-4842-2047-4
- [4] Y. Chi et. al., "Knowledge-Based Fault Diagnosis in Industrial Internet of Things: A Survey," in *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12886–12900, 1 Aug.1, 2022
- [5] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503.
- [6] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
- [7] B. Cheng et. al., "Industrial cyberphysical systems: Realizing cloud-based big data infrastructures," *IEEE Ind. Electron. Mag.*, vol. 12, no. 1, pp. 25–35, Mar. 2018
- [8] P. Leitaio et. al., "Smart agents in industrial cyber-physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1086–1101, May 2016.
- [9] M. Dzaferagic et. al., Fault Detection and Classification in Industrial IoT in Case of Missing Sensor Data, in *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8892–8900, 1 June1, 2022
- [10] S. J. Oks, A. Fritzsche et. al., "An application map for industrial cyber-physical systems," in *Industrial Internet of Things*. Cham, Switzerland: Springer, 2017, pp. 21–46.
- [11] Ayman Elhalwagy, Tatiana Kalganova: Hybridization of Capsule and LSTM Networks for unsupervised anomaly detection on multivariate data. *CoRR* abs/2202.05538 (2022)
- [12] S. Yin et. al., "Real-time monitoring and control of industrial cyber-physical systems: With integrated plant-wide monitoring and control framework," *IEEE Ind. Electron. Mag.*, vol. 13, no. 4, pp. 38–47, Dec. 2019.