2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom) | 979-8-3503-2922-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom59178.2023.00109

Anomaly Detection in Battery Charging Systems: A Deep Sequence Model Approach

Li Zheng[1,2,*], Donghui Ding[3,4,*], Zhao Li[3,5(✉)], Jun Gao[1,2(✉)], Jie Xiao[3], Hongyang Chen[5(✉)], Schahram Dustdar[6], Ji Zhang[7]

[1]The Key Laboratory of High Confidence Software Technologies, Ministry of Education, China
[2]School of Computer Science, Peking University, Beijing, China
[3]Hangzhou Yugu Technology, Hangzhou, China, [4]East China Normal University, Shanghai, China
[5]Zhejiang Lab, Hangzhou, China, [6]Technical University of Vienna, Austria
[7]University of Southern Queensland, Toowoomba, Australia
{greezheng,gaojun}@pku.edu.cn, 72275900022@stu.ecnu.edu.cn, lzjoey@gmail.com,
113006811@qq.com, dr.h.chen@ieee.org, dustdar@dsg.tuwien.ac.at, Ji.Zhang@usq.edu.au

***Abstract*—While the popularity of electric vehicles brings great convenience to our lives, battery charging also leads to an increase in accidents, resulting in personal injuries and economic losses. The methods currently embedded in charging hardware mainly focus on the short-term state of the battery and fail to leverage historical information effectively. The development of the Industrial Internet of Things (IIoT) enables data collection from sensors on industrial devices, which can be analyzed using deep learning methods to support sophisticated analysis. This paper proposes an intelligent and secure battery charging system in the IIoT that establishes an interaction between battery charging devices and cloud-based algorithms. A novel anomaly detection method is introduced to deal with anomalous charging sequences by making good use of historical data. We evaluate our system using real-life data from 4,940 batteries in electric vehicles, and our experiments achieve satisfactory results in detecting anomalies in battery charging.**

***Index Terms*—Battery charging, industrial internet of things, anomaly detection, electric vehicle.**

I. Introduction

With the growing concern regarding the greenhouse effect and global warming, electric vehicles (EVs) have garnered significant attention and have witnessed an increasing market share in recent years. Rechargeable lithium-ion batteries, known for their advantages such as high energy density, high cell voltage, high efficiency, and long lifespan, are gradually replacing gasoline as the primary energy source in many electric vehicles, including cars, trucks, and bicycle [1]. However, despite the convenience offered by EVs, the potential risks associated with charging lithium-ion batteries cannot be ignored.

The security of EVs during the battery charging process remains an important concern that needs to be addressed. Despite batteries being designed with security measures, issues can still arise during their use [2], [3]. On one hand, unlike controlled laboratory or factory environments, the actual conditions for battery charging in real life are often unknown and varied. Factors such as humidity, high temperature, and physical impacts can lead to fluctuations in the battery's state. On the other hand, over an extended period of use, electronic components within the battery can age, resulting in differences in resistance and voltage among cells [4]. These variations can potentially lead to short circuits. Considering the high energy density of lithium-ion batteries, accidents involving these batteries can result in violent combustion or even explosions, posing a significant threat to people's lives and property [5].

To ensure a secure and stable experience for users, establishing a dependable charging infrastructure is crucial for controlling the charging process of EVs equipped with lithium-ion batteries. While rule-based methods embedded in integrated chips can detect certain anomalies, they often rely on predetermined thresholds. As a result, they may fail to capture subtle fluctuations in battery charging and overlook valuable historical information. Advancements in technology have made it increasingly convenient to collect data transmitted from sensors on industrial devices through the Industrial Internet of Things (IIoT) [6]. This wealth of data can be utilized in various applications, including tasks such as anomaly detection. The development of IIoT provides us with opportunities to monitor the real-time status of the battery charging process. By analyzing the data transmitted through IIoT, we can promptly identify potential risks associated with the battery charging process and take appropriate measures to mitigate them.

In this paper, an intelligent and secure battery charging system is proposed that leverages the Industrial Internet of Things (IIoT) to address the aforementioned issues. During the battery charging process, the system utilizes sensors to collect a wealth of data, including current, voltage, and power readings. These data points are then transmitted to the cloud, where a vast amount of charging sequence data is generated and stored for further analysis and processing.

Within the system, we propose a novel **An**omaly **d**etection model on **Ba**ttery **ch**arging (**AndBach**) for our secure battery

[*]Equal contribution
[✉]Contact Authors

979-8-3503-2922-3/23/$31.00 ©2023 IEEE
DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom59178.2023.00109

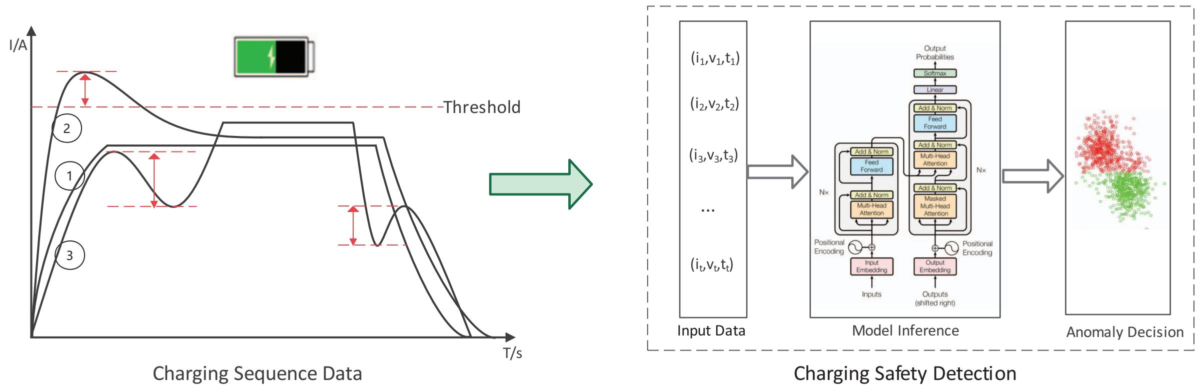

Fig. 1. The secure battery charging system utilizes a novel anomaly detection model. To overcome the limitations of rule-based methods, charging sequence data is transmitted to the cloud center via IoT technology. The cloud center processes the input data using the model and produces anomaly decisions for the battery charge sequences.

charging system. To address the fact that battery errors are not solely dependent on the current charging process but also influenced by historical charging patterns, we propose a multi-to-one encoder-decoder architecture. This design enables the comprehensive integration of the battery's charging evolution history. In order to handle irregular charging sequences, we tackle the challenge of varying data formats in the uploaded data. To accurately represent the battery's charging process, we employ different positional encodings for time and charge.

Specifically, the main contributions of our work are summarized as follows:

- Based on our in-depth study of the secure battery charging problem, we have developed a novel intelligent and secure battery charging system within the context of the Industrial Internet of Things (IIoT). This system is specifically designed to address the challenges associated with secure battery charging.
- We propose a novel anomaly detection model, called AndBach, in which we introduce a multi-to-one encoder-decoder architecture specifically designed to address the challenges of analyzing the sequence of time series data in battery charging. To accurately represent the status of battery charging, we adopt different positional encodings for time and charge. This enables us to precisely capture the temporal information and the varying charge levels during the charging process.
- We evaluate our system using large real-life datasets and experiment results show that our system achieve satisfactory results for detecting anomalies of battery charging.

The remaining parts of the paper are structured as follows. Section 2 reviews the application of IoT technology to lithium-ion batteries, followed by a discussion of the main approach to time series data. In Section 3, we introduce our proposed method and explain how the architecture model works. Section 4 presents the experimental results and analyzes the findings. In Section 5, we discuss the implementation and the deployment of a large-scale charging system. Finally, in Section 6, we summarize our findings and conclude the paper.

II. Related Work

In this section, we will review the applications of IoT technology in the context of lithium-ion batteries. Then, we will discuss some major approaches used for analyzing and processing time series data.

A. Applications of IoT for battery management

A real-time system for battery monitoring is build in [7] by using a coulomb counting method, where sensing technology, central processor and interfacing devices in IoT provide the environment for the implement of the proposed battery management system. [8] explores blockchain technology for ensuring the communication and data security of IoT devices from malicious cyber-attacks.

To allow individual battery cells communicate with the cloud and the battery model predicts the battery states executed in the cloud, [9] builds a cloud-based battery condition monitoring platform, which is able to early warn with various formats and material systems of lithium-ion batteries. [10] proposed a online estimation of battery health to facilitates the life-cycle management by capturing and characterizing instantaneous voltage drops, which are called V-edge dynamics.

In order to improve the effectiveness of signal processing, [11] builds an intelligent analysis system for signal processing tasks based on the LSTM recurrent neural network algorithm. To classify and detect single as well as multiple faults, measurements were made of supply air temperature, OA-damper position, supply fan pressure, indoor temperature and airflow rate in a variable air volume heating ventilating and air conditioning test facility. [12] embed a combination CNN and RNN to estimate Lithium-ion battery on IoT micro-controllers, where CNN preform the feature extraction and RNN perform the time-series prediction.

To improve the maintenance efficiency and lessen batteries operating risks in data centers, [13] use the time series clustering to present a battery anomaly detection method which uses only battery operating data and does not depend on offline testing data. [14] present an anomaly detection strategy for thermal parameters for lithium-ion batteries. A multiple-model residual generation method is proposed for anomaly detection and an RLS-based observer is presented to decouple the electrical and thermal dynamics.

B. Methods for time series data analysis

To process and analyse the time-series data, there have been numerous traditional signal processing techniques applied to extract series-data features, such as temporal features, spectral features and time-frequencys features. Transformation methods can recognize and extract time-frequency features in the time-series data. Fourier analysis can be used to decompose this signal in its periodic components [15]. Similar to Fourier Transform, Wavelet Transform transform a signal into its frequency domain and the output of a Wavelet transform hash a high resolution in the frequency domain and also in the time domain [16]. For spectral features, Higher-order spectral analysis(HOSA) allows one to reveal phase coupling between different frequency components [17]. For temporal features, Zero-crossing and period-amplitude analysis (PAA) can be adopted within frequency bands to mitigate the effects of noise and to reduce the issues associated with signals comprised of multiple components [18]. Detrended fluctuation analysis(DFA) is a method to characterise long range temporal correlations in a time series and can be used as a measure of self-similarity [19]. After extracting these features, we can input them in standard classifiers like Random Forest [20], Logistic Regression [21], Gradient Boosting [22] or Support Vector Machines [23]. However, unlike the deep learning methods, these methods can not extract high-order and nonlinear information in time-series data and also can not be trained in end-to-end.

Despite the above signal processing methods, quantity of time-sequence predicting approaches based machine learning have been proposed in recent years, such as RNNs and LSTM [24]. To alleviate the difficulties like gradient vanishing and exploding in training RNNs, the LSTM employs several gates to control which information the model goes to select, forget and update. GRU [25] simplifies the gates in the LSTM and achieves similar results with a smaller number of parameters, which avoids over-fitting. DARNN [26] is a dual-stage attention-based recurrent neural network where an input attention mechanism is introduced in the first stage to adaptively extract relevant driving series at each time step and a temporal attention mechanism is used to select relevant encoder hidden states across all time steps. ARIMA [27] is a generalized model of Autoregressive Moving Average (ARMA) that combines Autoregressive (AR) process and Moving Average (MA) processes and builds a composite model of the time series. DeepAR [28] is an autoregressive RNN-based prediction method that trains a model to obtain a joint conditional probability distribution, thereby generating a probability value for each time stamp

Unlike the RNN-based methods, Transformer [29] enables the model grasp the recurring patterns with long-term dependencies by a brand new architecture which leverages attention mechanism to process a sequence of data and allows the model to access any part of the history regardless of distance. Transformer-XL [30] is a variant of Transformer which use a segment-level recurrence mechanism and a novel positional encoding scheme.

TABLE I
NOMENCLATURE

Notation	Description
i	The serial number of battery b_i
y_i	The ground-truth label for battery b_i
N	The total number of batteries
$\mathcal{B}$	The set of batteries
d_0	The dimension of data points
$\mathcal{X}$	A charging sequence for a battery
l_x	The length of a charging sequence
$t^{(l)}$	The timestamp for the l-th data point
$Q^{(l)}$	The state of charge for the l-th data point
R_i	The evolutionary history for battery b_i
M_i	The number of charging sequences for battery b_i
$\mathcal{L}_i$	The set of charging sequence lengths for battery b_i
Q_S	The standard capacity of charge for a battery
d	The dimension of model
SOC	The state of charge
$\mathcal{P}et$	The positional encoding of time
$\mathcal{P}eq$	The positional encoding of quantity
$f(\mathcal{R}_i)$	The anomalous probability of battery b_i
$\mathbf{Q}, \mathbf{K}, \mathbf{V}$	Query, Key and Value vectors

ReFormer [31] replaces dot-product attention by one that uses locality-sensitive hashing and uses reversible residual layers instead of the standard residuals to improve the efficiency of Transformer. To solve locality-agnostics and memory bottleneck of Transformer, [32] propose LogSparse with $\mathcal{O}(L(logL)^2)$ memory cost. It produces queries and keys with causal convolution and improves forecasting accuracy for time series with fine granularity and strong long-term dependencies under constrained memory budget.

Informer [33] is an efficient transformer-based model for long sequence time-series forecasting. It propose a ProbSparse Self-attention mechanism, a self-attention distilling and a generative style decoder to achieve $\mathcal{O}(LlogL)$ in time complexity and memory usage.

III. METHODOLOGY

In this section, we will begin by defining the key concepts that are essential for understanding the context of this paper. We will then formalize the problem under investigation. Finally, we present the model architecture, illustrated in Figure 2, which outlines the flow of our proposed approach.

1) Problem Formulation: The goal of this paper is to detect anomalous batteries in $\mathcal{B}$. Specifically, given a battery b_i, and its evolutionary history $\mathcal{R}_i = \{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}, ..., \mathcal{X}_i^{(M_i)}\}$ with lengths $\mathcal{L}_i = \{l_x^{(1)}, l_x^{(2)}, ..., l_x^{(M_i)}\}$, this paper produces $\hat{y}_i = f(\mathcal{R}_i)$, the potential anomalous probability of battery b_i.

A. Notations

Let $\mathcal{B} = \{b_i\}_{i=1}^N$ be the set of batteries, where N is the number of batteries. There are d_0 sensors on each battery.

Definition 1: Data point. After the system sends a signal for data collection, each sensor takes one measurement of its monitored state, and the battery then uploads a data point $x \in \mathbb{R}^{d_0}$ to the server.

Definition 2: Charging Sequence. For a single charging process of a battery, the period from the time it is put into the charging cabinet to the time it is taken out, is called a charging sequence. We denote a charging sequence as $\mathcal{X} = \{x^{(1)}, x^{(2)}, ..., x^{(l_x)}\}$, where l_x is the length of this charging sequence. Generally, l_x is variable.

Definition 3: Evolutionary History. For a battery $b_i \in \mathcal{B}$, after a long period of use, it will have a series of charging sequence records. We will denote all the charging sequences of a battery as $\mathcal{R}_i = \{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}, ..., \mathcal{X}_i^{(M_i)}\}$, where M_i is number of charging sequences for battery b_i. We call it as the evolutionary history of a battery. Noting that the lengths of charging sequences are different, we denote the set of their lengths as $\mathcal{L}_i = \{l_x^{(1)}, l_x^{(2)}, ..., l_x^{(M_i)}\}$.

B. Multi-to-one Architecture

For the classification task on a single sequence, many methods use an encoder-decoder architecture [29], [34], [35] to model time series data. Specifically, the encoder maps the input time series $\mathcal{X} = \{x^{(1)}, x^{(2)}, ..., x^{(l_x)}\}$ to the hidden state space to obtain the corresponding representation $\mathcal{H} = \{h^{(1)}, h^{(2)}, ..., h^{(l_x)}\}$, and the decoder maps $\mathcal{H}$ to the output representation $\mathcal{Z}$. The representation $\mathcal{Z}$ will be the input of downstream tasks. However, in our problem, the evolution history of a battery includes multiple charging sequences, which makes it difficult for us to adopt a conventional encoder-decoder architecture directly.

To solve the sequence of sequences problem, we introduce a multi-to-one encoder-decoder architecture. As shown in Figure 2, the core idea behind multi-to-one architecture is to build multiple encoders for charging sequences in the evolutionary history of a battery. After each charging sequence is mapped to the corresponding representation, the decoder will produce a summary embedding for the representations except the one of the last charging sequence. The summary embedding and the representation of the last charging sequence will be concatenated and output to the fully-connected layer.

Specifically, for a battery $b_i \in \mathcal{B}$, we have its evolutionary history $\mathcal{R}_i = \{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}, ..., \mathcal{X}_i^{(M_i)}\}$ with lengths $\mathcal{L}_i = \{l_x^{(1)}, l_x^{(2)}, ..., l_x^{(M_i)}\}$. For each charging sequence in $\mathcal{R}_i$, we perform the encoder on it and get its representation:

$$\mathcal{H}_i^{(m)} = \mathbf{Enc}\big(\mathcal{X}_i^{(m)}; \Theta_E\big) \tag{1}$$

where $m \in [1, ..., M_i]$ is the serial number of the charging sequence, and Θ_E is the parameter of the encoder block Enc, which will be defined in Section III-C. Therefore, we get the representation sequence $\{\mathcal{H}_i^{(1)}, ..., \mathcal{H}_i^{(M_i)}\}$ of charging sequences for battery b_i. The decoder summarizes $(M_i - 1)$ representations and concatenated it with the last one:

$$\mathcal{Z}_i = \Big|\Big|\Big(\mathbf{Dec}\big(\mathcal{H}_i^{(1)}, ..., \mathcal{H}_i^{(M_i-1)}; \Theta_D\big), \mathcal{H}_i^{(M_i)}\Big) \tag{2}$$

where $||$ represents the concatenation operation, and Θ_D is the parameter of the decoder block Dec, which will be defined in Section III-D. We feed $\mathcal{Z}_i$ into a fully-connected network to get the results for the downstream task:

$$\hat{y}_i = \text{Softmax}(\mathbf{Ful}(\mathcal{Z}_i; \Theta_F)) \tag{3}$$

where Θ_F is the parameter of the fully-connected network **Ful**, and Softmax is operated along the dimension for classes. The model outputs the probability that the current sample belongs to each classes, *i.e.,* the probability that the sample is positive represents its anomaly score. Empirically, cross-entropy is used to evaluate the difference between the prediction $\hat{y}_i$ and the ground-truth label y_i of battery b_i:

$$L(\mathcal{R}_i|\Theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i) \tag{4}$$

where $\Theta = [\Theta_E, \Theta_D, \Theta_F]$. In our task, the goal is to learn the model parameters Θ by optimizing the objective:

$$\min_{\Theta_E, \Theta_D, \Theta_F} \mathcal{L} = \frac{1}{N} \sum_{b_i \in \mathcal{B}} L(\mathcal{R}_i|\Theta_E, \Theta_D, \Theta_F) \tag{5}$$

where y_i is the ground-truth label of battery b_i.

C. Encoder

1) Transformer: Transformer [29] and its variants [36]–[39] have been widely used for processing time series data in many fields. The multi-head self-attention mechanism enables Transformer to capture short and long-term dependency in different temporal aspects. In a multi-head self-attention layer, the model projects the input feature matrix $\mathcal{X}$ into multiple query matrices, key matrices, and value matrices:

$$\mathbf{Q}^j = \mathrm{Q}^j_{Linear}(\mathcal{X}), \mathbf{K}^j = \mathrm{K}^j_{Linear}(\mathcal{X}), \mathbf{V}^j = \mathrm{V}^j_{Linear}(\mathcal{X}) \tag{6}$$

where $j \in [1, .., h]$ is the serial number of multi-heads , h is the total number of heads, and $\mathrm{Q}^j_{Linear}, \mathrm{K}^j_{Linear}, \mathrm{V}^j_{Linear} \in \mathbb{R}^d \to \mathbb{R}^{\frac{d}{h}}$ are linear projections for key, query and value, respectively.

The output of multi-head scaled dot-product attention block are then fed into the decoder.

2) Informer: The $\mathcal{O}(L^2)$ complexity makes it difficult for Transformer to run on long charging sequences and huge amounts of data in our IIoT scenario. We adopt several optimization measures in Informer [33] to make the model achieve the time complexity and memory usage of $\mathcal{O}(L \log L)$.

Specifically, a kernel smoother is used to define the i-th query attention in a probabolity form, and a sparsity measurement for i-th query is calculated with the help of a simplified form of Kullback-Leibler divergence. Based on

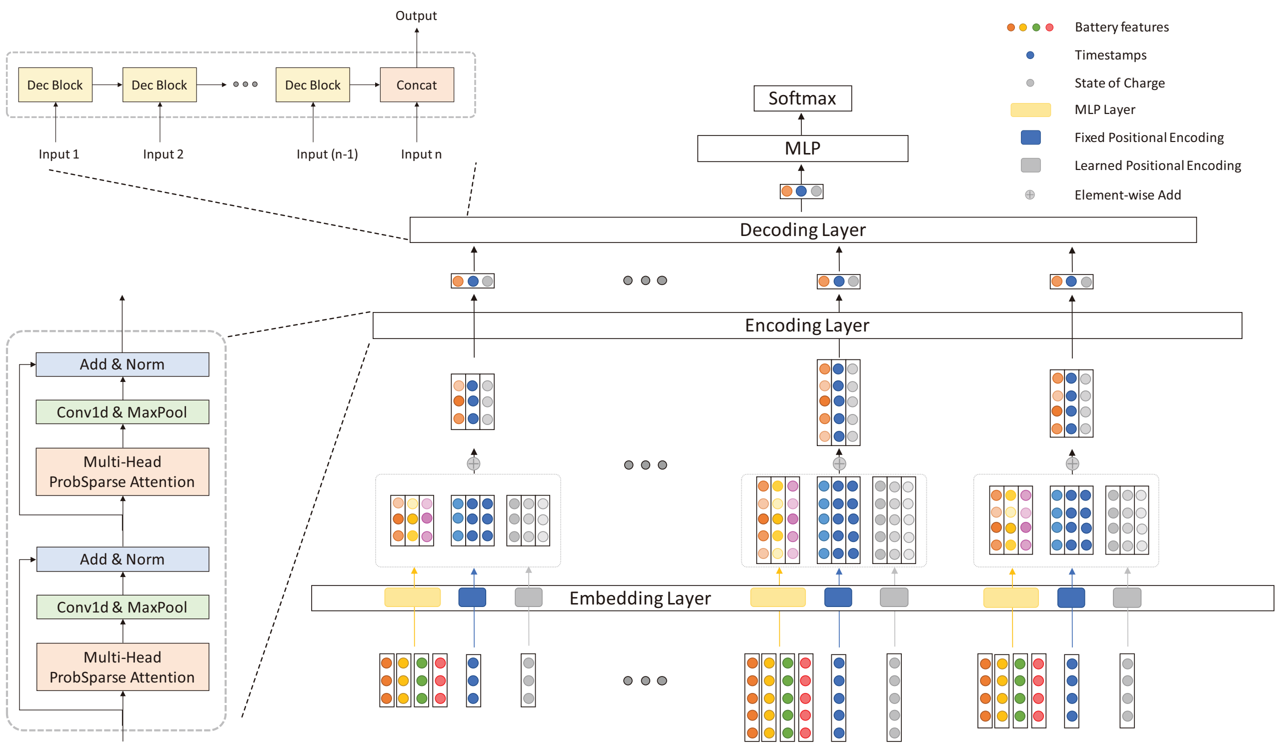

Fig. 2. The multi-to-one encoder-decoder architecture of the proposed method. The battery features, timestamps, and states of charge from the raw data are processed by MLP, fixed positional encoding, and learned positional encoding in the embedding layer to obtain the corresponding embeddings, respectively, after which they are element-wise added. Then, the sequence embeddings is input into the encoder to get the summarized embedding for the current charging sequence. Finally, the embedding of each sequence in a sample will be fed into the decoder and the final anomaly decision will be based on the output of the decoder.

the two above definitions, the ProbSparse self-attetion can be expressed as:

$$\mathcal{A}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}(\frac{\overline{\mathbf{Q}}\mathbf{K}^T}{\sqrt{d}})\mathbf{V} \tag{7}$$

where $\overline{\mathbf{Q}}$ is constructed by the top queries according to the defined sparsity measurement.

After the self-attention block, a distilling module is applied from l-th layer to $(l+1)$-th layer:

$$\mathcal{X}^{l+1} = \text{MaxPool}\Big(\text{ELU}\big(\text{Conv1d}(\tilde{\mathcal{X}}^l)\big)\Big) \tag{8}$$

where Conv1d is the 1-D convolutional filter along the dimension of time and ELU is the activation function.

3) Positional Encoding: The order of the sequence is very important for modeling the charging process. The self-attention mechanism does not include recurrence and convolution, so we use positional encoding to capture the features of sequence order. We use the time and quantity of charge as the position encoding rather than the position of the data point in the sequence.

On the one hand, unlike regular time series data, the time interval of the battery charging data obtained by the system over IoT is variable, which leads to the use of time rather than the order to more accurately describe the data points in the overall sequence. On the other hand, the quantity of charge reflects the progress of the battery charging process, which hints at the possible range of the individual battery parameters. We use fixed and trainable positional encoding to model time and quantity, respectively.

For positional encoding of time, we use the sine and cosine functions of different frequencies [29]:

$$\mathcal{P}et_{t,2i} = sin(t/10000^{2i/d}) \tag{9}$$

$$\mathcal{P}et_{t,2i+1} = cos(t/10000^{2i/d}) \tag{10}$$

where t is the timestamp of a data point and i is the dimension.

For positional encoding of quantity, we actually encode the state of charge, which is defined as the ratio of quantity of charge left in the battery to the quantity when the battery is brand new:

$$SOC = \frac{\int I\text{d}t + Q_0}{Q_S} \tag{11}$$

where Q_S is the nominal quantity of charge and Q_0 represents the quantity of charge at $t = 0$, considering that the charging sequences of batteries usually start at non-zero charge. The interval of SOC from 0 to 1 is cut into m segments, each cor-

TABLE II
EXPERIMENTAL RESULTS FOR COMPARISON ON CLASSIFICATION

Method	Precision	Recall	F1-score
RNN	0.5350	0.4269	0.4748
GRU	0.6118	0.3833	0.4714
LSTM	0.5572	0.4365	0.4895
DARNN	0.6250	0.4476	0.5216
Transformer	0.8802	0.8161	0.8469
Informer	0.8694	0.8477	0.8584
AndBach-T	0.9294	0.9317	0.9305
AndBach-I	**0.9552**	**0.9413**	**0.9482**

responding to a trainable embedding. The positional encoding of quantity is written as:

$$\mathcal{P}eq(SOC) = \mathbf{S}\big[\lfloor SOC * m \rfloor, : \big] \tag{12}$$

where $\mathbf{S} \in \mathbb{R}^{m \times d}$ is the matrix of trainable embeddings for state of charge.

D. Decoder

We adopt LSTM as the decoder of our method. The LSTM network [24] consists of LSTM blocks, each of which controls the selection, forgetting, and updating of information through input gates, forgetting gates, and output gates. We abbreviate above six steps as Dec-Block, as shown in Figure 2.

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets: The dataset we use is from Company X, which provides services such as battery sharing and public charging. The dataset comprises battery charging data samples from 4940 electric vehicle (EV) batteries. Among these samples, 2724 are classified as normal, while 2216 exhibit anomalous charging sequences. Each sample consists of numerous charging sequences, collected between August 2020 and July 2021. The data collected by sensors capture battery status information, resulting in 62-dimensional data points for each sequence. The data features include time, total current, total voltage, cell voltage, balanced current, quantity of charge, temperature, switch status, etc.

To preprocess the original data, we apply one-hot encoding and standardization techniques. As a result, each data point is represented by a 62-dimensional feature vector. Regarding the labeling of the data, for a normal battery, we consider all of its charging sequences as normal. However, for an anomalous battery, only its last charging sequence is labeled as faulty, while the preceding sequences remain unlabeled.

Given the significant relationship between battery anomaly detection and safety, it is crucial to prioritize both high precision and high recall in our evaluation. To capture the trade-off between precision and recall, we utilize the F1 score as the primary metric for assessing the performance of our experiments.

TABLE III
ABLATION STUDY

Encoder	PE	Decoder	Precision	Recall	F1-score
Transformer	w/o PE	RNN	0.8913	0.8893	0.8903
		GRU	0.8821	0.9015	0.8917
		LSTM	0.9001	0.8824	0.8912
		Transformer	0.8946	0.8768	0.8856
	w/ PE	RNN	0.9012	0.8943	0.8977
		GRU	0.8927	0.9088	0.9007
		LSTM	0.9290	0.9321	0.9305
		Transformer	0.9142	0.8924	0.9032
Informer	w/o PE	RNN	0.9048	0.8961	0.9004
		GRU	0.8932	0.9026	0.8979
		LSTM	0.9219	0.9076	0.9147
		Transformer	0.9178	0.9011	0.9094
	w/ PE	RNN	0.9333	0.9081	0.9205
		GRU	0.9149	0.9297	0.9223
		LSTM	**0.9552**	**0.9413**	**0.9482**
		Transformer	0.9348	0.9011	0.9176

2) Baselines and Implementation Details: We compare our method with the following competitive baselines:

- RNN-based: RNN, DARNN [26], DeepAR [28], ARIMA [27].
- LSTM-based: LSTM [24], GRU [25].
- Transformer-based: Transformer [29], Informer [33]

To streamline the experiment, we have applied a data filtering process to ensure that each charging sequence in our dataset consists of at least 30 data points. For the data splitting, we randomly divided the dataset into three sets: the training set, the validation set, and the test set. The ratio between these sets is 7:1:2, respectively. To maintain fairness and ensure a consistent comparison, our proposed method and all baseline approaches utilize the same network architecture settings. The implementation of our method is done using PyTorch 1.8.1, a popular deep learning framework.

B. Experimental Results

We make the comparison of classification on the real-life dataset. The experimental results are summarized in Table II. The suffix of AndBach represents the encoder block, where -T is Transformer and -I is Informer.

As shown in Table II, we can see that the proposed method AndBach outperforms all the baselines significantly and consistently. In particular, our method achieves relative performance gains over state-of-the-art baseline Informer by 10.5% in terms of F1 score. The results show that our method is capable of detecting anomalous charging sequences.

C. Ablation Study

We perform an ablation study on our method AndBach. The candidate set of encoders is Transformer, Informer and the

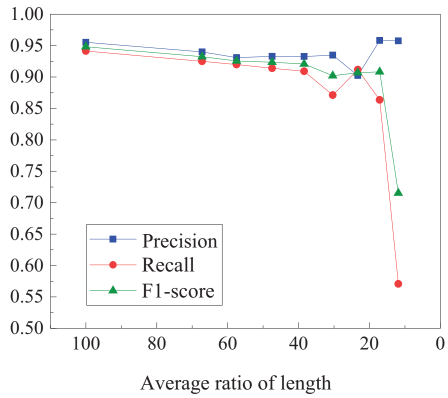

Fig. 3. Parameter Sensitivity. The x-axis represents the average ratio of lengths actually used for the model.

set of decoders is RNN, GRU, LSTM, Trasformer. Positional encoding will also be optional.

Experimental results are shown in Table III. After the introduction of positional encoding, the performance of each encoder-decoder combination is significantly improved, which shows that positional encoding can accurately capture the order information in the sequence. Under the same conditions, Informer performs better continuously as an encoder than Transformer, which reflects its ability to capture long-term dependence in the sequence. In terms of decoders, the performance of LSTM is better than that of Transformer, which may be since the two-tier Transformer architecture is difficult to be fully trained.

D. Parameter Sensitivity

We perform a parameter sensitivity study for the proposed method AndBach. We try to use charging sequences in limited lengths for the training and inference of our model. The range for the length l_0 of the sequence actually used is $\{25, 30, 35, 40, 45, 50, 55, 60\}$.

Experimental results are shown in Figure 3. For ease to compare, instead of length as the x-axis, we use the average ratio of actual used length to the sequence length:

$$p = \frac{1}{\sum_{i=1}^{N} M_i} \sum_{i=1}^{N} \sum_{j=1}^{M_i} \left(\frac{l_0}{l_{x,i}^{j}}\right) \tag{13}$$

With the decrease of p, the performance of the model decreases slightly. Although the performance of the model has a huge fall when p reaches 11.9% ($l_0 = 25$), the overall performance is pretty stable. In particular, F1-score of the model remains above 0.9, when only 17.1% ($l_0 = 30$) of the length in average is used. The experimental results show that our secure battery charging system can detect the anomalies of the battery even with limited data, which saves time for repairing the battery before the accident occurs.

V. System Development

In this section, we introduce the implementation and the deployment of large-scale charging system with real-time secure protection. The workflow of the charging systems is shown in Figure 4.

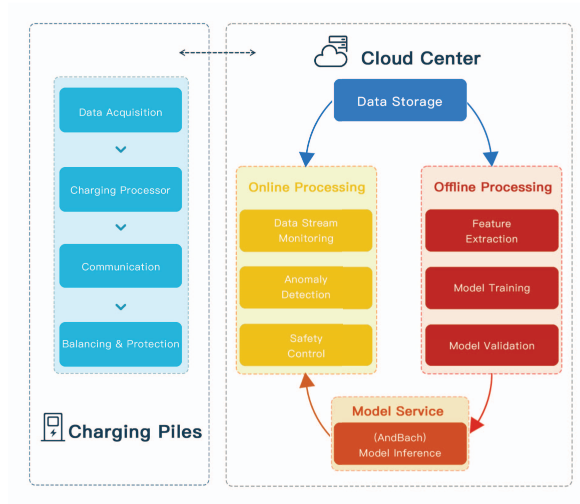

Fig. 4. Details of Online System Development.

The secure charging system mainly involves two parts: Charging Pile and Cloud Center. In the Charging Pile part, we develop the charging pile which includes data acquisition module, charging processor module, communication module, balancing and protection module. Based on those modules, we can achieve efficient data collection, real-time communication, and balanced charging control with safety protection. In the Cloud Center part, we aim to achieve exception detection on the collected charging data using the proposed AndBach algorithm. Based on the data type in the cloudy storage, Cloud Center can also divided into two parts: online and offline processing. In the offline processing, we firstly extract features from the collected raw data, and then divide the datasets to achieve model train and model validation in the large-scale cloudy environment. In the online processing, we regard the real-time data stream as input, then apply the AndBach approach by model inference to detect the anomaly, and finally handle the exception by controlling the charge process on/off to protect the electric vehicles safe.

Note that the training time of the AndBach is typically less than 1.5 hours on the cloud environment. The average response time of the secure charging system is typically less than 30ms.

VI. Conclusion

In this paper, we study the security of the battery charging process and design an intelligent, secure battery charging system with the help of IoT. We propose a novel anomaly detection model, AndBach, in this battery charging system. This model uses a multi-to-one encoder-decoder architecture to handle the sequence of time series data, and different positional encoding for time and charge to accurately reflect the status of battery charging. Our secure battery charging system was evaluated on data from 4,940 electric-vehicle batteries over the course of a year. Real-life experiments showed that our system is capable of detecting potential anomalies during battery charging. As a result, it may be applied to more charging scenarios in the future, thus protecting the safety of various electric devices.

Acknowledgment

This work was partially supported by NSFC (No.61832001, No.62272008 and No.62271452), and Key Research Project of Zhejiang Lab (No.2022PI0AC01).

References

[1] M. A. Hannan, M. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 834–854, 2017.

[2] A. B. Lopez, K. Vatanparvar, A. P. D. Nath, S. Yang, S. Bhunia, and M. A. Al Faruque, "A security perspective on battery systems of the internet of things," *Journal of Hardware and Systems Security*, vol. 1, no. 2, pp. 188–199, 2017.

[3] A. Amiruddin, A. A. P. Ratna, and R. F. Sari, "Systematic review of internet of things security," *International Journal of Communication Networks and Information Security*, vol. 11, no. 2, pp. 248–255, 2019.

[4] Z. Chen, R. Xiong, J. Tian, X. Shang, and J. Lu, "Model-based fault diagnosis approach on external short circuit of lithium-ion battery used in electric vehicles," *Applied energy*, vol. 184, pp. 365–374, 2016.

[5] S. Wang, L. Shang, Z. Li, H. Deng, and Y. Ma, "Lithium-ion battery security guaranteeing method study based on the state of charge estimation," *Int. J. Electrochem. Sci*, vol. 10, no. 6, pp. 5130–5151, 2015.

[6] C. Garrido-Hidalgo, F. J. Ramirez, T. Olivares, and L. Roda-Sanchez, "The adoption of internet of things in a circular supply chain framework for the recovery of weee: The case of lithium-ion electric vehicle battery packs," *Waste Management*, vol. 103, pp. 32–44, 2020.

[7] M. Asaad, F. Ahmad, M. Saad Alam, and Y. Rafat, "Iot enabled electric vehicle's battery monitoring system," in *Proceedings of the 1st EAI international conference on smart grid assisted internet of things*, 2017, pp. 1–10.

[8] T. Faika, T. Kim, J. Ochoa, M. Khan, S.-W. Park, and C. S. Leung, "A blockchain-based internet of things (iot) network for security-enhanced wireless battery management systems," in *2019 IEEE Industry Applications Society Annual Meeting*. IEEE, 2019, pp. 1–6.

[9] A. Adhikaree, T. Kim, J. Vagdoda, A. Ochoa, P. J. Hernandez, and Y. Lee, "Cloud-based battery condition monitoring platform for large-scale lithium-ion battery energy storage systems using internet-of-things (iot)," in *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2017, pp. 1004–1009.

[10] A. Kumar, M. A. Hoque, P. Nurmi, M. G. Pecht, S. Tarkoma, and J. Song, "Battery health estimation for iot devices using v-edge dynamics," in *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*, 2020, pp. 56–61.

[11] Y. Zhou and X. Jiao, "Intelligent analysis system for signal processing tasks based on lstm recurrent neural network algorithm," *Neural Computing and Applications*, pp. 1–13, 2021.

[12] G. Crocioni, D. Pau, J.-M. Delorme, and G. Gruosso, "Li-ion batteries parameter estimation with tiny neural networks embedded on intelligent iot microcontrollers," *IEEE Access*, vol. 8, pp. 122 135–122 146, 2020.

[13] S. N. Haider, Q. Zhao, and X. Li, "Data driven battery anomaly detection based on shape based clustering for the data centers class," *Journal of Energy Storage*, vol. 29, p. 101479, 2020.

[14] G. Dong and M. Lin, "Model-based thermal anomaly detection for lithium-ion batteries using multiple-model residual generation," *Journal of Energy Storage*, vol. 40, p. 102740, 2021.

[15] L. Cohen, *Time-frequency analysis*. Prentice hall New Jersey, 1995, vol. 778.

[16] A. Graps, "An introduction to wavelets," *IEEE computational science and engineering*, vol. 2, no. 2, pp. 50–61, 1995.

[17] C. L. Nikias and J. M. Mendel, "Signal processing with higher-order spectra," *IEEE Signal processing magazine*, vol. 10, no. 3, pp. 10–37, 1993.

[18] B. A. GEERING, P. ACHERMANN, F. EGGIMANN, and A. A. BORBÉLY, "Period-amplitude analysis and power spectral analysis: a comparison based on all-night sleep eeg recordings," *Journal of sleep research*, vol. 2, no. 3, pp. 121–129, 1993.

[19] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of dna nucleotides," *Physical review e*, vol. 49, no. 2, p. 1685, 1994.

[20] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and qsar modeling," *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[21] G. King and L. Zeng, "Logistic regression in rare events data," *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[22] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system acm sigkdd international conference on knowledge discovery and data mining," *ACM*, pp. 785–794, 2016.

[23] S. Suthaharan, "Support vector machine," in *Machine learning models and algorithms for big data classification*. Springer, 2016, pp. 207–235.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[26] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 2627–2633.

[27] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, 2014, pp. 106–112.

[28] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[30] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 2978–2988.

[31] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *International Conference on Learning Representations*, 2019.

[32] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 5243–5253.

[33] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of AAAI*, 2021.

[34] R. Yu, S. Zheng, A. Anandkumar, and Y. Yue, "Long-term forecasting using tensor-train rnns," *Arxiv*, 2017.

[35] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[36] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2249–2255.

[37] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, "An improved relative self-attention mechanism for transformer with application to music generation," *CoRR*, vol. abs/1809.04281, 2018. [Online]. Available: http://arxiv.org/abs/1809.04281

[38] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for asr," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5874–5878.

[39] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4055–4064.