RLIFE: Remaining Lifespan Prediction for E-scooters

Shuxin Zhong
Rutgers University
Piscataway, NJ, USA
sz517@cs.rutgers.edu

William Yubeaton
New York University
New York, NY, USA
wpy2004@nyu.edu

Wenjun Lyu
Rutgers University
Piscataway, NJ, USA
wenjun.lyu@rutgers.edu

Guang Wang
Florida State University
Tallahassee, Florida, USA
guang@cs.fsu.edu

Desheng Zhang
Rutgers University
Piscataway, NJ, USA
desheng@cs.rutgers.edu

Yu Yang
Lehigh University
Bethlehem, PA, USA
yuyang@lehigh.edu

ABSTRACT

Shared electric scooters (e-scooters) have been increasingly popular because of their characteristics of convenience and eco-friendliness. Due to their shared nature and widespread usage, e-scooters usually have a short lifespan (e.g., two to five months [2]), which makes it important to predict the remaining lifespan accurately, ensuring timely replacements. While several studies have focused on the lifespan prediction of various systems, such as batteries and bridges, they present a two-fold drawback. Firstly, they require significant manual labor or additional sensor resources to ascertain the explicit status of the object, rendering them cost-ineffective. Secondly, these studies assume that future usage is similar to historical usage. To solve these limitations, we aim at accurately predicting the remaining lifespan of e-scooters without extra cost, and its essence is to accurately represent its *current status* and anticipate its *future usage*. However, it is challenging because: i) lack of explicit rules for the e-scooters' status representation; and ii) e-scooters' future usage may significantly differ from their historical usage. In this paper, we design a framework called RLIFE, whose key insight is modeling user behaviors from trip transactions is of great importance in predicting the Remaining LIFespan of shared E-scooters. Specifically, we introduce an unsupervised contrastive learning component to learn the e-scooters' status representation over time considering degradation, where user preferences are served as a status reflector; We further design an LSTM-based recursive component to dynamically predict uncertain future usage, upon which we fuse the current status and predicted usage of the e-scooter for its remaining lifespan prediction. Extensive experiments are conducted on large-scale, real-world datasets collected from an e-scooter company. It shows that RLIFE improves the baselines by 35.67% and benefits from the learned user preferences and predicted future usage.

CCS CONCEPTS

• **Information systems** → **Data mining**.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615037

KEYWORDS

Micro-mobility Transportation, Remaining Lifespan Prediction, Contrastive Learning

ACM Reference Format:
Shuxin Zhong, William Yubeaton, Wenjun Lyu, Guang Wang, Desheng Zhang, and Yu Yang. 2023. RLIFE: Remaining Lifespan Prediction for E-scooters. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3615037

1 INTRODUCTION

Shared electrical micromobility have become increasingly popular in recent years. Let take e-scooters as a concrete example. Lime [4] served more than 55 million customers in 2021 and is projected to serve 124.8 million users in 2026 [1]. Compared with traditional human-powered bikes, e-scooters provide a faster and easier way to solve the first and last-mile problem during commuting, using battery-powered motors with speeds of up to 50 km per hour [3]. Due to their shared nature and widespread usage, e-scooters typically suffer from a short lifespan, ranging from two to five months [2]. Such a short lifespan makes it necessary to maintain or replace e-scooters timely in order to ensure a positive customer experience and prevent potential safety hazards before they become unserviceable. To this end, it is important to predict the remaining lifespan of e-scooters accurately.

To date, the remaining lifespan prediction problem has been studied in many systems, e.g., rail infrastructures [22], batteries [9], and bridges [23]. Existing works heavily rely on deploying dedicated sensors to collect explicit status indicators, e.g., state of health (SOH) in batteries [9]. Based on the sequentially collected data, [9, 38, 39] leverage neural networks(e.g., RNN, LSTM), to learn the non-linear degradation curve for the measured target, e.g., battery life curves [9]. However, those frameworks cannot be applied in our scenario directly, because: 1) the learned life curve typically works in ideal environments without considering uncertain noise; 2) e-scooters are sophisticated machines with multiple components (i.e., wheels, batteries, etc.) and different kinds of sensors are needed for status monitoring. Sensor deployment requires significant labor efforts and expensive fees, rendering it cost-effective. The limitations of the existing works motivate us to answer a research question: *can we predict the remaining lifespan of shared e-scooters without additional dedicated sensor deployment?*

In this work, we collaborate with an e-scooter company to learn the degradation process of e-scooters in a data-driven manner. This collaboration offers us the opportunity to predict the remaining lifespan based on large-scale operational data without extra labor or sensor deployment. Through detailed data analysis, we found that it is important to consider both e-scooters' *current status* and predicted *future usage* in lifespan prediction (supported by Figures 2, 3, 4). Though it sounds straightforward, there are two challenges:

- Lack of explicit rules for e-scooters' status representation and lack of explicit correlations between status and remaining lifespan. One straightforward approach is to leverage the served distance to estimate the status of e-scooters. Intuitively, a longer served distance leads to more significant wear and tear, consequently leading to a shortened lifespan. However, we found the correlation coefficient between the served distance and corresponding remaining lifespan is only 0.6302 (as depicted in Section 2). This relatively modest correlation is because of the fact that the longevity of e-scooters is not only affected by the used distance, but affected by other non-observable factors, e.g., weather conditions, riding habits, and accidents [2].
- The future usage of e-scooters deviates considerably from their historical patterns. Specifically, the daily trip distance decreases as the "age" of e-scooters increases (as shown in Section 2). For example, the average daily trip distance during the first 10% of the lifespan is 14.3% more than that in the last 10%.

To solve these challenges, we design a framework called RLIFE to predict the Remaining LIFespan of shared E-scooters. The key insight is that *modeling user behaviors from trip transactions is of great importance in remaining lifespan prediction* (detailed in Sec. 2). The rationale behind this insight is two-fold: i) the user behavior patterns indirectly reflect e-scooter status; and ii) user behavior trends can also provide valuable insights into future e-scooter usage. For instance, when faced with multiple nearby e-scooters, users typically opt for those in better condition, such as those without broken parts or with a pristine appearance. Furthermore, frequent usage is associated with accelerated wear and tear, ultimately resulting in a shortened lifespan. Drawing from this insight, we have devised two main components for our approach, including (i) self-supervised e-scooters status representation learning, and (ii) user preference evolution prediction. In component (i), we design a un-supervised contrastive learning, which learns the e-scooter's degradation status representation over time, where the trip records and user preferences are served as the direct and indirect reflectors, respectively. For component (ii), we train a recursive layer to project the user preferences after Δ days. Finally, we fuse the learned current status and Δ-day user preference to estimate the future status. By varying the value of Δ, we can estimate the future status of the e-scooter. Once the estimated status indicates that the end of life is approaching, we consider Δ as the remaining lifespan starting from the present moment. The key contributions of this paper are summarized as follows:

- We for the first time study the remaining lifespan prediction problem for e-scooters without extra dedicated sensors. It incorporates the *current status* representation learning and *future usage* estimation from operational data.
- We highlight the importance of modeling user preferences from transactions in the status learning and usage estimation. Specifically, we design an unsupervised contrastive learning framework to discriminate the lifespan status representation without annotations and a recursive layer to predict the dynamic future usage in a given Δ-day.
- We evaluate RLIFE with 9-month data collected from an e-scooter company in two cities. The results show RLIFE improves prediction accuracy by 35.67% and 29.81% compared with the SoA methods in the two cities. The code and the data are available [1].

The rest of the paper is organized as follows. In Section 2, we introduce the data sets, analyze the challenges and the key insight, and provide the formal definition of this problem. We show the technical design in Section 3, including the overview of RLIFE, and the detailed design. In Section 4, we evaluate the performance of RLIFE to show the effectiveness compared with baselines. We provide related works in Section 5. Finally, we discuss the lesson learned, the limitations, future works and privacy issues in Section 6 and conclude the paper in Section 7.

2 BACKGROUND AND MOTIVATION

2.1 Data

In this work, we mainly use two datasets, including an e-scooter trip record dataset and a weather dataset.

2.1.1 E-scooter Dataset. By collaborating with an e-scooter company , one of the major shared e-scooter service providers, we have access to real-world datasets in two cities of New Jersey, USA:

- In New Brunswick, the data is collected with 1, 179 e-scooters and 118, 609 trips in 9 months from April to December in 2021;
- In Newark, the data is collected with 639 e-scooters and 50, 631 trips in 4 months from August to December in 2021.

Each trip record captures data from the point a user picks up an e-scooter until the point the user drops it off, including vehicle ID, trip start and end time, and trip routes (i.e., GPS traces). All the data are obtained legally under the users' consents [6]. The detailed data format is listed in Table 1.

Table 1: Trip Record Format and Example

Field	Value
Vehicle ID	50109575
Trip ID	d0980b1f-59af-5944-980c-4ebb5336fdbe
Trip duration	211
Trip distance	232
Start time	August 7, 2021 8:57:29 PM
End time	August 7, 2021 8:59:49 PM
Routes	August 7, 2021 8:57:30 PM, [-74.448150, 40.499419], August 7, 2021 8:57:32 PM, [-74.448144, 40.499345], ...

[1] https://www.dropbox.com/s/2muo5q6gge0wd51/rlife-src.tar.gz

2.1.2 Weather Datatset. The weather condition data were collected from 2,400 stations in National Oceanic and Atmospheric Administration (NOAA) [5]. We utilize weather data from April 2021 to December 2021, including temperature, relative humidity, precipitation, wind speed and direction, visibility, atmospheric pressure, and duration of different weather types (e.g., rain, snow, etc.).

2.2 Problem Formulation

Suppose an e-scooter has been in service for t days, generating time-ordered trip records denoted as $R_t = [r_1, r_2, \ldots, r_t]$, where $r_i \in \mathbb{R}^{N_r}$ and N_r is the dimension of record.

Given those records, we aim to predict the remaining lifespan of this e-scooter. Formally, it is defined as:

$$\text{remaining lifespan} = \max \Delta | F(R_t, \Delta) \geq F_{th} \quad (1)$$

where Δ is the number of days, and F is the function that returns the probability of the e-scooter still in service in Δ days, F_{th} is a given probability threshold.

Because the remaining lifespan is affected by its current degradation status and future usage, Equation 1 is further extended to:

$$\text{remaining lifespan} = \max \Delta | F(f_d(R_t), f_s(R_t, \Delta)) \geq F_{th} \quad (2)$$

where $f_d(R_t)$ returns the status representation at t, $f_s(R_t, \Delta)$ returns future usage during t to $t + \Delta$.

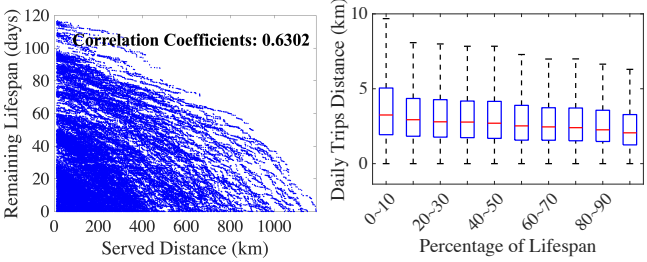


Figure 1: Correlations between Served Distance and Remaining Lifespan.

Figure 2: Daily Trips Distance vs. Percentage of Lifespan.

2.3 Key Insight

Our system RLIFE is based on one key insight: *modeling user behaviors from trip transactions is of great importance in reflecting current status and future usage for remaining lifespan prediction.* The rationale behinds it: i) the current user behavior patterns indirectly reflect e-scooter status; and ii) user behavior trends can also offer valuable insights into future e-scooter's usage. To visualize it, we quantify the user preference as selection probability, which is calculated as the total selected times over the total available times (as in Sec. 3.2). For example, if an e-scooter is available for 10 trips in a time period (e.g., within 10 meters to the start locations of these 10 trips) and it is selected twice, then the selection probability of this e-scooter is 0.2. Figure 3 shows that the e-scooters with long remaining days have a higher selection probability, which proofs that user behavior reflects the e-scooter's status. Figure 2 and Figure 4 shows that the selection probability and daily usage decreases with the increase of the e-scooters' "age", which validates the future usage is different from historical usage.

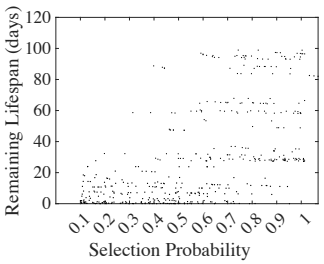


Figure 3: Selection Probability vs. Remaining Lifespan (days).

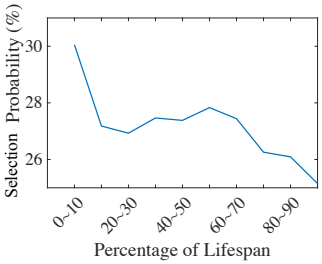


Figure 4: Selection Probability in Different Percentages of Lifespan.

2.4 Two Challenges

Even though the idea sounds straightforward, there are still two challenges, including a lack of explicit status representation and uncertain future usage. We perform data analysis to show the above two challenges as follows.

- **Lack of explicit rules for e-scooters' status representation.** Different from previous works [9, 29, 30, 39] that deployed sensors to monitor the operation status of machines, we lack the explicit factors to directly evaluate the e-scooters' status and the explicit relationships between status and remaining lifespan. The simplest way is that we can leverage the total served distance to reflect the remaining lifespan. Intuitively, a longer served distance may indicate a shorter remaining lifespan. However, when we investigate the correlation coefficient between the served distance and remaining lifespan (as shown in Figure 1), where each point is an e-scooter. We found that the coefficient between the total served distance and remaining lifespan is only 0.6302, which means the e-scooters with the same served distance may have significantly different remaining lifespans. In reality, the longevity of e-scooters is simultaneously affected by multiple factors, e.g., weather, riding habits, and accidents, which are non-observable sometimes [2]. Thus, it is inaccurate to directly use one single explicit data, e.g., the total served distance or duration, to represent the status of e-scooters.
- **E-scooters' future usages are significantly different from the historical usages.** As shown in Figure 2, the average trip distance continuously decreases as their "age" increases, which indicates a shift in usage patterns over time. Typically, we analyze the usage of e-scooters(i.e., average daily trip distance) during their different lifespan stages(i.e., from the first 10% to the last 10%). Future usage is one of the factors that affect the remaining lifespan. In this case, the remaining lifespan prediction works that do not explicitly consider the future usage patterns [22, 35] cannot achieve satisfying performance.

2.5 Motivation

Why do we choose contrastive learning? To better illustrate our motivation, we first introduce contrastive learning. It is an unsupervised framework that learns the general feature representations

from input data without explicit labels or categories. By comparing similar and dissimilar data pairs, the method can differentiate the two data types from the representations it learns. Typically, data augmentation techniques are designed and applied to generate similar data pairs, while other data points are treated as dissimilar pairs. In our research, as there are no clear indicators to represent the status of e-scooters, we utilize contrastive learning to investigate possible representations of the e-scooters' status. Using it, we aim to identify and differentiate the status of e-scooters by comparing similar and dissimilar pairs of data and learning representations that distinguish between them. Our assumption is that e-scooters that have similar trip records, such as similar locations and weather conditions, should have similar statuses. To take advantage of this, we use contrastive learning to optimize the alignment of the representations of e-scooters' status with similar trip records without the need for human annotations.

The key technical improvement. As described in literature [7, 11–13, 17, 18, 36], data augmentation is a critical element in contrastive learning. It plays an important role in creating semantically similar pairs of e-scooters' records, which in turn affects the quality of the learned representations of their status. However, traditional augmentation methods such as rotation or cropping, which are suitable for time-invariant data such as images or graphs, do not take into account temporal correlations and are not appropriate for sequential trip records. This highlights the need for specialized and tailored data augmentation techniques for our sequential data. To address these limitations, we have developed three specialized data augmentation techniques that take into account the time, geographical, and usage aspects of e-scooters' trip record simultaneously. These methods are called record masking, record shifting, and trip drifting, and they will be described in more detail in Section 3.3.

3 DESIGN OF RLIFE

3.1 Overall Architecture

Fig. 5 shows the overall architecture of RLIFE including Pre-processing, Status Representation Learning, Future Usage Prediction, and Remaining Lifespan Prediction.

3.1.1 Pre-processing. We extract features based on the aggregation of trip records, weather, and road networks. Specifically, we extract trip features (e.g., distance and duration) and user preference features (e.g., selection probability), and trip intervals for the status representation learning and future usage prediction, respectively.

3.1.2 Status Representation Learning. We process sequential trip features as explicit observation and user preference as an implicit reflection to learn e-scooters' current status representation. Specifically, we leverage a self-supervised contrastive learning component to discriminate the lifespan status representation. Different from traditional contrastive learning, we mainly have two improvements: i) the augmentation method perturbs the input in three aspects, i.e., time, geographical, and usage domain, simultaneously; ii) the similarity is guided by both degradation status and user preferences.

3.1.3 Future Usage Prediction. We model the dynamic evolution of user preference, which predicts the future embedding trend of user preference. This is done by leveraging an attention-based layer to project the embedding of user preference after a time lapse Δ. The projected embedding is used for downstream tasks, i.e., predicting the future usage at a given query time Δ.

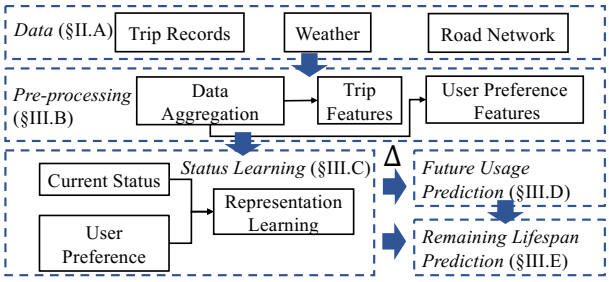


Figure 5: The Architecture of RLIFE. It consists of four components: Data Pre-processing, Status Learning, Future Usage Prediction, and Remaining Lifespan Prediction. In data preprocessing, we first clean the data, e.g., remove the outliers, and extract the features, including trip and user preference features(detailed in Sec.III.B). In status learning, we design an unsupervised contrastive learning framework to leverage the extracted features to discriminate the lifespan status representation without human annotations(detailed in Sec.III.C). In future usage prediction, we predict the future usage considering the given query time Δ(detailed in Sec.III.D). In the remaining lifespan prediction, we output the probability that an e-scooter is still in service after time Δ with the learned status representation from Sec.III.B. By computing the queries for multiple Δ, the final output is the Δ with the maximum probability.

3.1.4 Remaining Lifespan Prediction. We formulate the remaining lifespan prediction as a query task whose inputs are the e-scooters' current status and predicted future usage during time $[t, t+\Delta]$. The output is the probability that an e-scooter is still in service after time Δ. By computing the queries for multiple Δ, we can derive the distribution of the probability.

3.2 Pre-processing

We mainly clean the raw data, e.g., outliers removal, and extract features from them.

3.2.1 Data Cleaning. Since data collected from real-world sources may contain noise, we eliminate trip records with improbable speed and distance values. For example, e-scooters have a maximum speed of 30 mph [3]. So we remove the trips with an average speed of over 30 mph. Further, we identify and remove a set of trips with a distance over 25 miles that are abnormal in our dataset considering the service areas in the city.

3.2.2 Features Extraction. We first map the GPS points on the road network and obtain the sequence of passed regions. We then aggregate the records with weather information and derive the features from two aspects, i.e., trip features and user preference features.

- **Trip Features.** represent the trip features within one trip, including its start time and end time, origin and destination, trip

duration and distance, passed regions, and current weather situation (e.g., temperature, relative humidity, wind, etc.).

- **User Preference Features.** represent the trip features between consecutive trips, i.e., selection probability and idle intervals. Specifically, the selection probability p for an e-scooter is calculated by $p = \frac{n}{N}$ where N is the total potential trips for this e-scooter (i.e., this e-scooter is within a certain distance of the trip origin and can be potentially selected by users), n is the number of actually selected times. The idle intervals of an e-scooter are the interval between two consecutive trips, which also reflects its popularity.

3.3 E-scooter Status Representation Learning

3.3.1 Data Augmentation. Data augmentation is a critical step in contrastive learning. It helps to construct semantically similar e-scooters' record pairs and affects the quality of learned status representations. Current augmentation methods, e.g., rotation or cropping, are designed for time-invariant data, e.g., images or graphs, which do not consider temporal correlations and are not suitable for sequential trip records. Thus, we design three types of data augmentation methods for the e-scooters' record data.

Record Masking. Intuitively, the status of an e-scooter should be more similar to itself than to others, even though its historical usage is slightly adjusted. To reflect this, we disturb input data by selectively masking (deleting) the trip records for certain days.

Record Shifting. Similar trip records, e.g., used distance, frequencies, and regions, probably have a similar impact on status representation. Thus, we provide a record-shifting method that augments the data by shifting the trip records in the time domain. Specifically, this method involves selecting the trip records at random and shifting them to the neighboring days.

Trip Drifting. As we derive the geographical information from GPS sensors which may naturally have noise and drifting, such drifting should not impact our results too much. Therefore, we introduce a trip drifting augmentation, i.e., randomly disturbing the trip's passed region to some neighboring regions.

In our work, for each e-scooter, we treat the augmented trip records from the same e-scooter as the positive samples and the trip records from other e-scooters in the batch as negative samples.

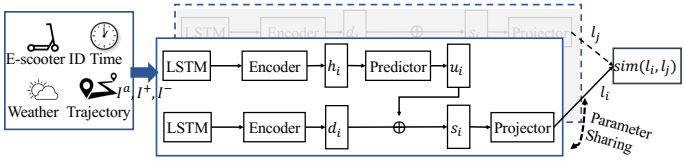


Figure 6: Contrastive Status Representation Learning. Based on trip records, each e-scooter is fed into an LSTM to generate the trip embedding. Then we encode the trip embeddings to get the corresponding degradation status and historical usage status. We then use a predictor to estimate the user preference based on historical usage status. Combining the user preference and degradation status, we project the e-scooter status which is then used to compare similar/dissimilar pairs.

3.3.2 Contrastive Learning. Fig. 6 shows the architecture of status representation learning. For each e-scooter x_i and its augmented sample x_j, we first embed them using an LSTM to generate trip embeddings. Then the embeddings are fed into an encoder $f_e(\cdot)$ (or $f'_e(\cdot)$) to get the corresponding degradation status d_i (or d_j). We adopt an MLP as the encoder. Following the design in [18], we call $f_e(\cdot)$ the query encoder and $f'_e(\cdot)$ the momentum encoder respectively. The degradation status representation d_i and d_j are extracted as:

$$\begin{aligned} d_i &= f_e(LSTM(x_i)) \\ d_j &= f'_e(LSTM(x_j)) \end{aligned} \tag{3}$$

where x_i and x_j are the input trip features.

Instead of directly calculating the similarity of the learned status representations, we introduce two projectors to project the status representations to different spaces. One is for the *status comparison* of similar/dissimilar e-scooters (i.e., self with its augmented positive samples and negative samples) and the other is for the *user preference estimation*. The motivation is that different views of the learned status representations in different spaces make it robust for different tasks.

Status comparison. We leverage an MLP as projector for status comparison and the process is formulated as:

$$l_i = g(d_i) = W^{(2)}\sigma(W^{(1)}d_i) \tag{4}$$

where σ is a ReLU non-linearity. After obtaining the output, we apply a loss function, following the form of InfoNCE [27], where one e-scooter is encouraged to be close to those with similar experiences.

$$\mathcal{L}^c_{l_i} = -log\frac{exp(l_i \cdot l_j^+/\tau)}{exp(l_i \cdot l_j^+/\tau) + \sum_{l_j^-} exp(l_i \cdot l_j^-/\tau)} \tag{5}$$

where l_j^+ is known as l_i's positive sample and the l_j^- is regarded as l_i's negative sample. τ is a temperature hyper-parameter for l_i and l_j with l_2 normalization [34].

User preference estimation. Meanwhile, we use user preference estimation to guide the status representation learning. We apply another projector to map the status representation d_i to the user preference space as p_i. We utilize an MSE loss function to calculate the loss between the projected user preference p_i and the ground-truth user preference p_i^y. Formally, the user preference estimation loss is defined as

$$\mathcal{L}^p(p_i, p_i^y) = \frac{1}{n} \cdot \sum_{t=1}^{n}(p_{i,t} - p_{i,t}^y) \tag{6}$$

Finally, we combine the contrastive loss function and the user preference estimation loss function as the total loss of our contrastive status representation learning. Formally, the total loss is defined as

$$\mathcal{L} = \mathcal{L}^p + w_l\mathcal{L}^c_{l_i} \tag{7}$$

where w_l is a learnable weight.

Momentum Update. After computing the total loss, we conduct back-propagation and update the parameters of the momentum encoder following the momentum update [18]. Specifically, given the momentum m, we update the f'_e by the following equation:

$$f'_e = m \cdot f'_e + (1 - m) \cdot f_e \tag{8}$$

3.4 Future Usage Prediction

After learning the status representation, the next step is to predict the future usage of the e-scooters. Traditional ways to predict future usage generally purely rely on the historical usage [22], while ignoring the degradation of the e-scooters. It makes the prediction sub-optimal because the usage would decrease with the gradual degradation of the e-scooters. In our work, we introduce the dynamically changed user preference (i.e., predicted future user preferences) in the prediction to represent the impacts of the degradation of the e-scooters on future usage.

3.4.1 Single-step Prediction. We consider the current e-scooter status and the user preference evolving process in the future usage prediction. As the user preference is influenced by the degradation status, we incorporate learned status representation to predict the future user preference and its influence on future usage.

We first apply an LSTM-based feature extractor to extract the trip features x_i^{trip}, which will be put into the encoder for the current status d_t generation. We then put the d_t into the user-preference projector to estimate the user preference p_i^t. The estimated user preference will be combined with the trip feature hidden states to predict future usage at time $t+1$.

$$\begin{aligned} h_t &= LSTM(x_t^{trip}) \\ p_t &= g'(f_e(h_t)) \\ \hat{x}_{t+1}^{trip} &= LSTM(h_t \oplus p_t) \end{aligned} \tag{9}$$

3.4.2 Multi-step Prediction. Given the output from the single-step prediction, we further design a recursive way for multi-step prediction. Similar to the single-step prediction, we first generate $\hat{x}_{t+1}^{trip}$ by Equation (9). Then, we treat $\hat{x}_{t+1}^{trip}$ as the input to generate $\hat{x}_{t+2}^{trip}$ in the same way. Given a future time slot parameter Δ, we can predict the future usage in the following Δ time slots $\hat{x}_{t+1}^{trip}, \hat{x}_{t+2}^{trip}, \ldots, \hat{x}_{t+\Delta}^{trip}$.

3.4.3 Multi-step Fusion. After predicting the future usage of the e-scooter in the following Δ time slots, we fuse the multi-step future usage to represent the future usage for this e-scooter. We apply an MLP network where the input is the predicted future usages $\hat{x}_{t+1}^{trip}, \hat{x}_{t+2}^{trip}, \ldots, \hat{x}_{t+\Delta}^{trip}$ and the output is the fused future usage feature at the following Δ days as follows:

$$u^{\Delta} = \phi(\hat{\mathbf{x}}_{\Delta}^{trip}, w_u) \tag{10}$$

where $\hat{\mathbf{x}}_{\Delta}^{trip} = \{\hat{x}_{t+1}^{trip}, \hat{x}_{t+2}^{trip}, \ldots, \hat{x}_{t+\Delta}^{trip}\}$, w_u are learnable parameters, and u^{Δ} is the predicted future usage.

3.5 Remaining Life Prediction

After learning the status representation and predicting the future usage, we predict the remaining life. Different from general machine learning tasks that directly output lifespan, we design a query scheme to output the probability of the predicted lifespan given Δ. In this way, we can introduce negative samples such as a very large lifespan but with a probability of 0. Formally, the remaining life prediction is defined as:

$$\text{remaining lifespan} = \max \Delta | F(d_t, u^{\Delta}, \Delta) \geq F_{th} \tag{11}$$

Considering a relatively small search space of Δ, we simply iterate all the possible Δ in a certain range (e.g., historically maximum lifespan of all the e-scooters) to obtain the optimal remaining lifespan.

4 EXPERIMENTS

4.1 Evaluation Settings

4.1.1 Baselines. We start this subsection by describing the baselines for comparison, followed by evaluation metrics. Then we summarize the implementation details. We include the following eight benchmark methods for evaluation, each of which serves as a representative framework for predicting the remaining lifespan of the e-scooter.

- **Historical Average (HA)**: We calculate the average length of lifespan for all the e-scooters and obtain the remaining lifespan of each e-scooter by subtracting the duration of service.
- **XGBoost [10]**: It is a boosting tree-based method that achieved outstanding performance in many prediction tasks. In our implementation, the input is the trip features, and the output is the remaining days.
- **LSTM [38]**: The Long Short-Term Memory Network is a suitable model for sequential data learning, i.e., sensors in manufacturing machines. The input of our baseline is the trip features in sequences, and the output is the same as that in XGBoost.
- **TCN [19]**: It is for rolling bearing remaining lifespan prediction. The input and output of the temporal convolutional network (TCN) are the same as that of LSTM. The difference between LSTM and TCN is that LSTM emphasizes long-term and short-term influences while TCN focuses on the neighboring influences determined by kernel size k. We set k to 5.
- **Linear Regression [26]**: It is a straightforward approach that uses a linear function to model the correlation between the input and the output. The input is the aggregated trip records, which is the same input used in the XGBoost method.
- **Auto-encoder [25]**: Auto-encoder uses the encoder-decoder framework with multiple-layer neural networks for the bearing lifespan prediction. It takes in the same input data as XGBoost and captures the complex, non-linear relationships for more accurate predictions.
- **Belief Network [21]**: It is a model for the machine's remaining lifespan prediction. It consists of multiple stacked restricted Boltzmann machines for greedy layer-by-layer training. Its input is the same as that of the XGBoost model.
- **AdaCare [20]**: The model is a general health-status representation learning model. It first adopts dilated convolutional layers as short, medium, and long-term convolutional layers for various time scales, where the kernel size k is set to 1, 2, and 3, respectively. Then, it adopts two fully-connected layers to learn the nonlinear dependencies between features explicitly.

4.1.2 Metrics. We introduce three metrics to evaluate the prediction performance, i.e., Mean Absolute Errors (MAE), Root Mean Squared Errors (RMSE), and Mean Absolute Percentage Error (MAPE). In particular, we use a day as the unit of the lifespan, which is consistent with the minimum operational intervals, such as daily rebalancing or charging.

Table 2: Overall Prediction Performance of Different Methods on the Newark and New Brunswick Datasets.

	Newark			New Brunswick		
	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)
HA	33.04±1.21	29.63±1.15	48.52±2.86	35.54±1.27	32.64±1.18	49.61±3.14
XGBoost [10]	26.47±1.18	19.29±1.16	35.31±2.93	29.70±1.22	21.18±1.15	44.83±3.07
LSTM [38]	25.73±0.94	23.13±1.01	28.14±1.87	28.90±0.98	24.27±1.02	33.81±2.37
TCN [19]	26.17±0.97	22.52±0.99	28.65±1.77	29.02±1.09	22.53±1.11	34.69±1.98
Regression [26]	17.23±1.09	12.13±0.98	20.87±0.95	21.06±1.12	13.57±1.07	21.43±0.96
Auto-encoder [25]	15.33±0.56	10.25±0.43	19.83±0.97	18.43±0.45	12.13±0.51	20.58±1.03
Belief Network [21]	19.23±0.43	14.09±0.37	21.32±1.06	19.98±0.52	14.47±0.48	22.90±1.26
AdaCare [20]	12.56±0.35	10.86±0.28	18.64±1.03	15.35±0.27	12.82±0.35	19.68±1.01
RLIFE	**6.51±0.21**	**4.97±0.11**	**13.96±0.94**	**7.23±0.17**	**5.46±0.13**	**15.74±0.82**

4.1.3 Implementation Details. The implementation details of each component are described as follows.

Contrastive learning. For the experiments, we split all datasets into training, validation, and testing sets with a 6:3:1 ratio. For e-scooter's trip records, we apply the methods introduced in Sec. 3.3.1 to construct positive samples. Similarly, we process other e-scooters' trip records as negative samples.

Future usage prediction. In order to dynamically explore future usage, we formulate it as a query task with a time variable Δ. Δ changes from 1 to the maximum threshold, which we set 50 in the experiments.

Remaining lifespan prediction. For each Δ, the output is the probability that this e-scooter is still alive in service in Δ days. By comparing the query results on multiple Δ to the given probability threshold, we find the largest one as our predicted remaining lifespan. Intuitively, the probability threshold is set to be 0.5.

We implement RLIFE with Keras 2.4 and test it on a server with NVIDIA A4000 GPU with Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, 256GB memory. For the hyper-parameters, the batch size is 256, and the decay weight is 10^{-6} for all datasets. The momentum coefficient is set as 0.5 and 0.9 for Newark data and New Brunswick data, respectively (detailed in Sec. 4.4). For contrastive learning, the learning rate is set as 1.5×10^{-5}, as the momentum mechanism requires a relatively smooth parameter update [14]. For the remaining life prediction, we set the learning rate as 0.01. The dimension of status representation is optimized as $1,024$. We optimize it with the Adam optimizer for 100 epochs and do not apply any non-mentioned optimization techniques. All the experiments are repeated 5 times, and the performances are presented using the "mean±standard deviation" format.

4.2 Overall Performance

From Table 2, we observe that:

- In general, the models [21, 25] that focus on capturing the integrated features of e-scooters' status (i.e., total served distance, and total served duration) achieve better performance than that [19, 38] explore the accumulated influences of time-series trip records. It is because the integrated results have a stronger representative power of e-scooters' status, and the models learned by individual records may drop partial information.
- AdaCare [20] outperforms others [19, 38] because it integrates the status representation considering the temporal correlation.
- RLIFE gains 35.67% and 29.81% improvement compared with AdaCare [20] by leveraging user behavior as an implicit input for the degradation status representation learning.

Moreover, different from previous work [19, 25], we explore the influences of dynamic future usage on the remaining days of service. The results show that the remaining lifespan of the e-scooter is determined by both its current degradation status and dynamic future usage.

4.3 Ablation Studies

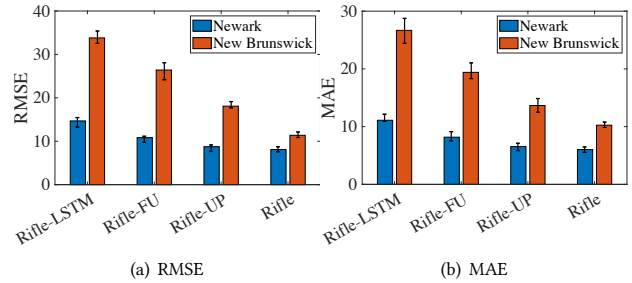
We conduct a comprehensive ablation study to further evaluate the status representation learning component, the future usage prediction component, and the impact of the user preference. We build the following variants of RLIFE.

- RLIFE-lstm removes the LSTM module and replaces it with the integrated trip records, i.e., total served distance and duration, to evaluate the strength of the degradation learning process.
- RLIFE-FU removes the future usage prediction module (i.e., Sec. 3.4) and predicts the remaining lifespan according to historical usage.
- RLIFE-UP removes the contributions of user preferences by (i) removing the user preference estimation loss in the status learning part and (ii) using LSTM only in the future usage prediction part without the user preferences.

We present the results in Fig. 7 and find that:

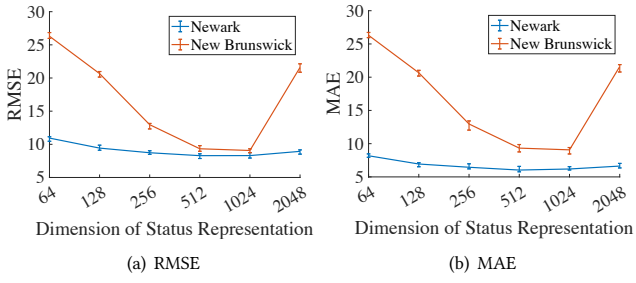
- RLIFE outperforms RLIFE-lstm, which demonstrates the importance of the degradation process (i.e., daily trip records).
- RLIFE outperforms RLIFE-FU, which shows the future usage is inconsistent with the historical usage and predicting the future usage strengthens the prediction performance.
- RLIFE outperforms RLIFE-UP, which verifies our intuition that the user preferences can serve as an implicit input to imply the overall status, and then improve the performance.

Overall, the results show that the learning of the degradation process (i.e., LSTM module), the future usage prediction, and the users' preference should be considered to improve the prediction performance.



(a) RMSE (b) MAE

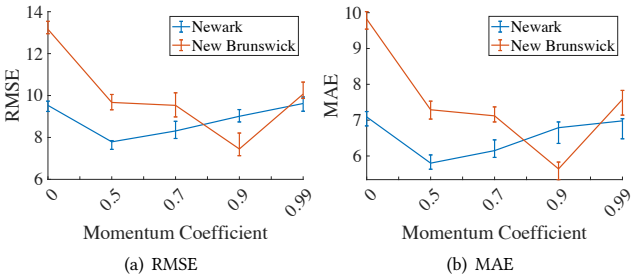
Figure 7: The Performance of Different Variants.



(a) RMSE (b) MAE

Figure 9: The Impact of Representation Dimension.

4.4 Sensitivity Analysis



(a) RMSE (b) MAE

Figure 8: The Impact of Momentum Coefficients.

The Momentum Coefficient. One key parameter in RLIFE is the momentum coefficient m, which influences the degradation condition representation learning. In general, the momentum coefficient adjusts the update rate or the encoders' consistency. If it is set to 0, it means the parameters of the momentum encoder are always updated with the query encoder. Such drastic updates influences the consistency of the encoded positive and negative samples, which eventually affects the representation learning. A relatively larger value indicates the samples are encoded by a slowly progressing encoder, which ensures consistency for better learning. However, if it is set close to 1 (e.g., 0.99), the encoders tend to keep the original parameters, which may also affect the representation learning. Thus, the optimal momentum coefficient needs to be neither too small nor too large. Fig. 8(a) and 8(b) show the effects of different momentum coefficients in the New Brunswick and Newark datasets, respectively. We observe that RLIFE achieves the best performance when the coefficient is set to be 0.9 and 0.5 in the New Brunswick and Newark dataset, respectively. This is mainly because the New Brunswick dataset has a much larger data capacity than the Newark dataset, which needs a larger momentum coefficient. Compared to not using the momentum encoder (i.e., set the coefficient to 0), the momentum encoder improves the performance by 13.6%.

Dimension of Learned Representation Vector. Another critical parameter in RLIFE is the dimension of the learned status representation vector, which indicates the information diversity. Fig. 9(a) and Fig. 9(b) show the effects of dimensions of representation vector on the Newark and New Brunswick datasets. We observe that on one hand, a larger dimension benefits to contain more information and learn more accurate status representation; on the other hand, a too large dimension significantly increases the number of parameters, leading to overfitting and low performance.

5 RELATED WORK

5.1 Remaining Lifespan Prediction.

There are lots of works exploring the information in operational records for remaining lifespan prediction such as trips, billing, and medical records. It can be further categorized into model-based and data-driven methods. For model-based methods, they use mathematical models to fit a degradation curve of the target, e.g., battery life curves [9]. However, they typically work in an ideal environment without noise and uncertainty. For data-driven methods, neural networks [38, 39] are applied to historical data to learn the non-linear degradation trend of sequential data. For example, MLP is useful for learning non-linear degradation patterns [39], but it lacks the ability to incorporate temporal information. Then, the RNN-based frameworks, e.g., RNN [39], LSTM [38], have been applied to learn the degradation trend of sequential data. Zhang et al. [39] utilized the long short-term memory (LSTM) recurrent neural network (RNN) to learn the long-term dependencies among the degraded capacities of lithium-ion batteries.

However, those methods heavily rely on sensors to collect explicit status indicators, e.g., state of health(SOH) in batteries [9], which incurs two limitations in our problem. First, existing sensors are designed to monitor only certain components of e-scooters, such as batteries [39], while other components, such as wheels and brakes, cannot be well monitored (or need more sophisticated and expensive sensors). Second, the cost of sensors is proportional to the number of e-scooters, so it is expensive to deploy sensors at a large scale.

5.2 Representation Learning

Representation learning aims to learn a low-dimensional vector for data representation, such as graphs[33], and hidden status [15, 20, 37]. For instance, AdaCare [20] depicted the health status by capturing the long and short-term variations of biomarkers and modeled the correlation between clinical features to enhance the ones which indicate the health status. GRASP [37] proposed a generic framework for healthcare models which aims to solve data sparsity or low-quality data. Med2Vec [15] learned the representations for both

medical codes and visits from large EHR datasets with over a million visits. PNRL [33] proposed a predictive network representation for the structural link prediction. PTARL [31] explored the peer and temporal dependencies of driving behavior with GPS trajectories data.

However, those frameworks focus on individual status learning rather than learning similar or dissimilar representations from data organized into similar or dissimilar pairs.

5.3 Contrastive Learning.

Contrastive representation learning made a great success in practice in classifying groups of images unsupervisedly [7, 11–13, 18, 36]. It benefits to identify two key properties related to the contrastive loss: (1) alignment (i.e., closeness) of features from positive pairs, and (2) uniformity of the induced distribution of the normalized features [28, 32]. For example, SimCLR [11] proposed two major components to enable the contrastive prediction tasks to learn useful representations, including data augmentation and learnable nonlinear transformation. MoCo V2 [12] used an MLP projection head and more data augmentation with Momentum Contrast (MoCo), which outperformed SimCLR and did not require large training batches. BYOL [16] introduced a new framework for self-supervised representation learning, which relies on two neural networks, including online and target networks that interact and learn from each other. However, contrastive methods typically have real-time requirements and need many explicit pairwise feature comparisons, which incur a high computational cost. For efficiency, SwAV [7] is an online algorithm without being required to compute pairwise comparisons. SimSiam [13] simplified the BYOL framework by removing: (i) negative sample pairs, (ii) large batches, (iii) momentum encoders, and achieved surprising empirical results. BARLOW TWINS [36] did not require large batches nor asymmetry between the network twins, i.e., a predictor network, gradient stopping, or a moving average on the weight updates.

In summary, *Contrastive learning* is a great self-supervised approach that benefits learning similar or dissimilar representations from data. It is suitable to learn the similar or dissimilar degradation status of e-scooters without explicit status measures. In this work, we enhance the generic contrastive learning with a new data augmentation method for sequential data and introduce user preferences as implicit feedback to improve representation learning.

6 DISCUSSION

6.1 Lessons Learned

Based on the design, implementation, and evaluation of RLIFE, we learned the following lessons:

- **User behavior performs well as an implicit input to measure e-scooters status.** The key insight of RLIFE is that user behavior, i.e., user preferences, can be utilized as the implicit input to learn the e-scooters' degradation status. That is, a less-selected e-scooter (i.e., low selection probability) or longer idle time e-scooter (i.e., long idle intervals between consecutive trips) generally has a worse condition. Supported by Fig. 7, we found that introducing user preferences helps our model gain 24.96% and 7.95% improvement in the New Brunswick and Newark datasets, respectively.
- **Future usage dynamics should be considered in the remaining lifespan prediction.** Different from the existing lifespan prediction that the future usage generally is consistent with the historical usage, e-scooters' usage changes as the degradation status changes. Our ablation study validates the necessity of considering changed future usage for the remaining lifespan prediction. Supported by Fig. 7, we observed that the future usage prediction component leads to the performance improvement of 47.16% and 26.31%.
- **User preferences can be used to improve future usage prediction.** Predicting future usage can be challenging if a dynamic degradation process is involved. In our work, we use the learned status representation as an opportunity to estimate future user preference, which in turn supports future usage prediction. Supported by Fig. 7, we observed that the introduction of user preference estimation in the future usage prediction improves the performance by 27.25% and 26.39%.

6.2 Practical Implications of the results

In this work, we focus on modeling e-scooters' current status and future usage to provide a more accurate prediction about the remaining lifespan. The potential implications include that the results (i.e., estimated remaining lifespan) can be utilized to further study the e-scooters' re-balancing problem [8, 24]. For instance, we can re-balance the e-scooters with longer lifespan (i.e., good condition) to the areas with higher demand to increase the users' satisfaction. And we can also re-balance the e-scooters with shorter lifespan to low-demand areas to increase overall lifespan.

6.3 Ethics and Privacy

During the data analysis and data mining of the trip records, we took careful steps to address ethical and privacy concerns. First, all the e-scooter users have digested the Terms of Services and consent the platform can collect their trip trajectories for research and service improvement. Second, all the raw data has been pre-processed into aggregated anonymous statistics based on the privacy protection requirements during the data collection process. All the user identifiers are removed, and all the auxiliary information is strictly limited to GPS traces.

7 CONCLUSION

In this work, we design a framework called RLIFE for remaining lifespan prediction of e-scooters with user preferences consideration. Our RLIFE validates that the user preference is beneficial to be explored as the implicit input for the e-scooters' degradation status representation learning. Moreover, future usage prediction contributes to prediction performance. Based on the experiment results, RLIFE can improve the performance by up to 35.67% compared with the baseline methods. We also demonstrate the effectiveness of our RLIFE with different ablation studies and parameters analysis.

ACKNOWLEDGMENT

This work is partially supported by NSF 1932223, 1951890, 1952096, 2003874, 2047822, 2246080. We thank all the reviewers for their insightful feedback to improve this paper.

REFERENCES

[1] 2019. Lime. https://www.statista.com/outlook/mmo/shared-mobility/shared-rides/e-scooter-sharing/worldwide.

[2] 2019. Shared scooters don't last long. https://scooter.guide/how-long-do-electric-scooters-last/.

[3] 2021. E-scooter Speed. https://electricscootering.com/electric-scooters-fast/.

[4] 2022. Lime. https://www.li.me/en-US/home.

[5] 2022. NOAA. https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ncdc:C00684.

[6] 2022. User Agreement. https://www.veoride.com/user-agreement/.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems* 33 (2020), 9912–9924.

[8] Stefano Carrese, Fabio d'Andreagiovanni, Tommaso Giacchetti, Antonella Nardin, and Leonardo Zamberlan. 2021. A beautiful fleet: Optimal repositioning in e-scooter sharing systems for urban decorum. *Transportation Research Procedia* 52 (2021), 581–588.

[9] Daoquan Chen, Weicong Hong, and Xiuze Zhou. 2022. Transformer Network for Remaining Useful Life Prediction of Lithium-Ion Batteries. *IEEE Access* 10 (2022), 19621–19628. https://doi.org/10.1109/ACCESS.2022.3151975

[10] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[13] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15750–15758.

[14] Xinlei Chen, Saining Xie, and Kaiming He. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9640–9649.

[15] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. 2016. Multi-layer representation learning for medical concepts. In *proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1495–1504.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems* 33 (2020), 21271–21284.

[17] Baoshen Guo, Weijian Zuo, Shuai Wang, Wenjun Lyu, Zhiqing Hong, Yi Ding, Tian He, and Desheng Zhang. 2022. WePos: Weak-Supervised Indoor Positioning with Unlabeled WiFi for On-Demand Delivery. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 54 (jul 2022), 25 pages. https://doi.org/10.1145/3534574

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[19] Chongdang Liu, Linxuan Zhang, and Cheng Wu. 2019. Direct remaining useful life prediction for rolling bearing using temporal convolutional networks. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2965–2971.

[20] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. 2020. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 825–832.

[21] Meng Ma, Chuang Sun, and Xuefeng Chen. 2017. Discriminative deep belief networks with ant colony optimization for health status assessment of machine. *IEEE Transactions on Instrumentation and Measurement* 66, 12 (2017), 3115–3125.

[22] Annemieke Meghoe, Richard Loendersloot, and Tiedo Tinga. 2020. Rail wear and remaining life prediction using meta-models. *International Journal of Rail Transportation* 8, 1 (2020), 1–26.

[23] Hani G Melhem and Yousheng Cheng. 2003. Prediction of remaining service life of bridge decks using machine learning. *Journal of Computing in Civil Engineering* 17, 1 (2003), 1–9.

[24] Jesus Osorio, Chao Lei, and Yanfeng Ouyang. 2021. Optimal rebalancing and on-board charging of shared electric scooters. *Transportation Research Part B: Methodological* 147 (2021), 197–219.

[25] Lei Ren, Yaqiang Sun, Jin Cui, and Lin Zhang. 2018. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. *Journal of Manufacturing Systems* 48 (2018), 71–77.

[26] Yinxian Song, Huimin Li, Jizhou Li, Changping Mao, Junfeng Ji, Xuyin Yuan, Tianyuan Li, Godwin A Ayoko, Ray L Frost, and Yuexing Feng. 2018. Multivariate linear regression model for source apportionment and health risk assessment of heavy metals from different environmental media. *Ecotoxicology and Environmental Safety* 165 (2018), 555–563.

[27] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* (2018), arXiv–1807.

[28] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2495–2504.

[29] Guang Wang, Yuefei Chen, Shuai Wang, Fan Zhang, and Desheng Zhang. 2023. ForETaxi: Data-Driven Fleet-Oriented Charging Resource Allocation in Large-Scale Electric Taxi Networks. *ACM Trans. Sen. Netw.* 19, 3, Article 63 (mar 2023), 25 pages. https://doi.org/10.1145/3570958

[30] Guang Wang, Zhou Qin, Shuai Wang, Huijun Sun, Zheng Dong, and Desheng Zhang. 2022. Towards Accessible Shared Autonomous Electric Mobility with Dynamic Deadlines. *IEEE Transactions on Mobile Computing* (2022), 1–16. https://doi.org/10.1109/TMC.2022.3213125

[31] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Pengfei Wang, Yu Zheng, and Charu Aggarwal. 2018. You are how you drive: Peer and temporal-aware representation learning for driving behavior analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2457–2466.

[32] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[33] Zhitao Wang, Chengyao Chen, and Wenjie Li. 2017. Predictive network representation learning for link prediction. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*. 969–972.

[34] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3733–3742.

[35] Boyuan Yang, Ruonan Liu, and Enrico Zio. 2019. Remaining useful life prediction based on a double-convolutional neural network architecture. *IEEE Transactions on Industrial Electronics* 66, 12 (2019), 9521–9530.

[36] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*. PMLR, 12310–12320.

[37] Chaohe Zhang, Xin Gao, Liantao Ma, Yasha Wang, Jiangtao Wang, and Wen Tang. 2021. GRASP: Generic Framework for Health Status Representation Learning Based on Incorporating Knowledge from Similar Patients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 715–723.

[38] Jianjing Zhang, Peng Wang, Ruqiang Yan, and Robert X Gao. 2018. Long short-term memory for machine remaining life prediction. *Journal of manufacturing systems* 48 (2018), 78–86.

[39] Yongzhi Zhang, Rui Xiong, Hongwen He, and Michael G Pecht. 2018. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. *IEEE Transactions on Vehicular Technology* 67, 7 (2018), 5695–5705.