國立臺灣大學管理學院資訊管理學研究所

碩士論文

Graduate Institute of Information Management

College of Management

National Taiwan University

Master's Thesis

適用於物聯網機器服務業的深度學習輔助容量管理

Deep Learning Assisted Capacity Management for IoT-Enabled Services Industries

曹竣瑋

Chun-Wei Tsao

指導教授：陳靜枝 博士

Advisor: Chern, Ching-Chin, Ph.D.

中華民國 113 年 7 月

July 2024

doi:10.6342/NTU202401885

謝辭

能完成這篇論文，想先感謝指導教授陳靜枝老師，從確定題目的一年多的期間內，老師每個周四都會不厭其煩的幫助我們推進論文的進度，針對我們的進度，不斷的提出更多的方式來解決所遇到的問題。老師經常說:「論文很可能是你們此生最完美最大型的著作，一定要好好的完成。」因此每次遇到挫折和困難時，我都有不能退縮的勇氣。經過了數個月的資料整理以及實驗、不同的嘗試後，我得出了意想不到的結果，讓我覺得這些日子的忙碌以及準備，都很值得。

感謝實驗室的夥伴郁茹，經常在課程、助教和論文的部分幫助我，讓我在碩士班期間的工作都能更有效率，也能在忙的時候有人可以幫忙分擔，最重要的是我們一起經歷的很長一段時間的努力，最後把自己的著作完成。

最後想感謝我的家人，我的阿嬤總是給予我很多支持，擔心我太累，姑姑、叔叔、嬸嬸也都給我很多關心，幫助。我的哥哥除了聽我抱怨遇到的挫折，分擔煩惱，也經常給我很多有用的建議，甚至還花時間幫我進行口試演練。還要感謝我最聰明的同學們: 巨屌宗佑、宇如、智傑、家興、家安、博傑、柏昇、俊廷、啟安、明非、庭麒、中則、小孟、澔澔，如果沒有他們，我無法完成我此生最大的著作，真心的感謝。

曹竣瑋 謹識

于臺大資訊管理研究所

民國一一三年七月

doi:10.6342/NTU202401885

論文摘要

近年來，因為物聯網 (IoT, Internet of Things) 機器的出現，配備機器的服務業在收集資料的方式可以轉為自動化。除了可以精確的得到機器運轉的情況，也能過濾掉機器使用狀況異常的訊號。從物聯網感測器所收集到的資料具備大數據的性質，包含資料量(Volume)，以及速度(Velocity)，和多樣性(Variety)。這樣的資料性質可以針對特定機器的運轉情況作出客製化的處理，詳細記錄時間的資訊也能更有效地執行時間序列分析。

本研究使用洗衣租賃業所提供的物聯網機器資料，針對 700 多台洗衣機總共兩年的資料進行時間序列分析及預測，並期望幫助服務業避免使用經驗做出新增機器的決策，轉向使用資料輔助預測的方式達到資源分配的目的。我們首先利用物聯網機器狀態轉變的訊號進行處理，依照不同的時間段切分運轉情況。再針對所有的機器使用狀況建立深度學習模型，搭自編碼器(Autoencoder)技術處理複雜大量的物聯網資料，進行準確的時間序列預測，可以得出特定機器未來的運轉狀況。除此之外，我們提出兩階段的深度學習架構，使用分群演算法搭配分類演算法，可以使用所收集到的物聯網機器資料建立模型，幫助沒有歷史資料的機器，針對機器的性質，所要放置的地點進行預測，計算出未來可能的運轉情況。

根據本研究所提出的深度學習預測架構，企業可以針對物聯網機器所預測出的運轉情況和當前的資源分配做出比較，了解機器分配的情況並做出調整。不管特定機器是否有歷史資料，都能夠運用我們提出的架構，得到準確的運轉情況預測。

關鍵詞: 物聯網機器、大數據、時間序列分析、深度學習

doi:10.6342/NTU202401885

THESIS ABSTRACT

DEPARTMENT OF INFORMATION MANAGEMENT
NATIONAL TAIWAN UNIVERSITY

NAME: Tsao, Chun-Wei MONTH/YEAR: JULY/2024
ADVISOR: Chern, Ching-Chin

In recent years, the advent of Internet of Things (IoT) devices has revolutionized data collection in the service industry, enabling automation and precise monitoring of machine operations. IoT sensors generate large-scale data characterized by volume, velocity, and variety, which can be tailored to analyze specific machine performance and facilitate effective time series analysis.

This study leverages IoT data from a laundromat industry, analyzing over two years of data from more than 700 washing machines. The goal is to transition from experience-based decision-making to data-driven predictions for resource allocation. We process signals from IoT devices to segment machine operations over different temporal dependencies. Using deep learning models and autoencoder technique, we accurately forecast time series data to predict future machine performance.

Additionally, we propose a two-phase deep learning framework that combines clustering and classification algorithms. This approach allows us to build predictive models for machines without historical data, estimating future performance based on related features such as machine characteristics and deployment locations.

Our predictive framework enables businesses to compare forecasted machine operations with current resource allocation, optimizing machine distribution and adjustments. Whether or not a machine has historical data, our approach provides accurate performance predictions.

Keywords: IoT machine data, Big Data, Autoencoder, Deep Learning

doi:10.6342/NTU202401885

Table of Contents

doi:10.6342/NTU202401885

doi:10.6342/NTU202401885

doi:10.6342/NTU202401885

List of Tables

doi:10.6342/NTU202401885

List of Figures

doi:10.6342/NTU202401885

Chapter 1

Introduction

1.1 Background and Motivation

Recent advancements in technology have improved the accuracy of data-driven decision-making. By analyzing historical data, service providers can gain insights into the previous performance of their services, enabling them to make decisions based on the past experience. In service industries, proper resource allocation allows increasing efficiency, reducing service idle time and improving overall performance. With the aid of historical data, service providers no longer have to guess the capacity in a long period, as well as taking the risk of unforeseeable future demand.

A significant development of this field is the rise of the Internet of Things (IoT). Through IoT devices, it is possible to meticulously record every aspect of a service operation. Automated data collection via these devices eliminated errors from manually recording, ensuring the data's accuracy and completeness. Beyond simple data collection, these devices process and convert raw data into actionable insights that can guide decision-making and strategic planning. The precision of IoT data down to every individual operational task boosts our confidence in predicting future service demands.

Service industries integrated with IoT enable data can also benefit from Big Data Analytics. While Big Data Analytics emphasizes on rapid data processing, IoT messages passing techniques ensure real time data transmission. The convergence of these technologies allows for comprehensive data analysis, resulting in more precise demand prediction than traditional approaches.

In recent service sector, the usage patterns of service provided may not be recorded correctly. For instance, service company that rent machines as their business model can

doi:10.6342/NTU202401885

be struggled from knowing the actual usage condition of each machines. Thus, they could not be clear about the demand of their business, which casued inefficiency. With the aid of IoT-enabled message passing devices, every single operational records are allowed to be recorded in a flexible way. Based on the received data format, service industries can utilize these data to conduct further analysis and make data-driven decisions.

1.2 Research Objectives

Capacity management has long been a significant challenge for services with fixed capacities. Such services often struggled from achieving optimal resource allocation, especially given that usage patterns can vary substantially across different time frames. Once resources are initially deployed, adjusting them in response to shifting demands becomes a formidable task. Nevertheless, capacity management can lead to better customer satisfaction, optimized operational cost and rectify service inefficiencies. If one could predict resource needs using historical data, it would be possible to optimize initial resource allocation and facilitate dynamic adjustments as needed. In this study, we will present the methodology that combines IoT-driven data with deep learning techniques to assist in predicting future demands, using a rental washing machine service as a case study.

In scenarios where service lacks historical data, decision-making primarily relies on predefined features, such as location, type of appliance, and other characteristics determined prior to the service's launch. Our goal is to introduce a framework that harnesses the power of deep learning models to facilitate optimal capacity planning in the service sector. This model aims to forecast potential usage patterns even when only foundational features are presented, overcoming the challenges of a cold start.

doi:10.6342/NTU202401885

1.3 Research Scope and Limitation

(1) Capacity-Constrained Services

In this study, our primary objective is to explore optimal capacity planning through the integration of IoT-driven data and a deep learning framework. Even though capacity planning could relate to personnel, facilities and equipment, we are trying to narrow our research scope to the equipment aspect in order to manifest the power of IoT collected data. The case used by this study is to use data from washing machines with IoT devices attached to collect real-time data. By using these data, we intend to build a framework that is capable of analyzing the positive consequences resulting from equipment reallocation. Also, the essence of capacity planning could be managed by two distinct aspects: managing the supply for a fixed demand, or managing the demand for a fixed supply[31]. We only focus on the service industries that present the property that the service supply is fixed, but demand fluctuates over time.

(2) Multiple Geographical Area

For the scope of capacity planning, we envision services with several locations in different geographical areas that the resources could be reallocated. For instance, equipment may be allocated to three different geographical areas, where each area contains five sub area. In this case, the resource allocation becomes even more complex and require data-support information to efficiently allocate machine and appliance, avoiding biases while making decisions. A key consideration in our model is the incorporation of location as a feature, which recognized its potential influence on customer demand. By analyzing location-specific data, we could further discern usage patterns across different regions to inform resource decisions for each service location.

doi:10.6342/NTU202401885

(3) Features Collected

Our methodology leans heavily on time series analysis to construct the predictive framework, and trying to use statistical methods to apply on the service industries that lack historical data. However, if only sparse features are available, for instance, only location and service type are provided, it would be challenging to correctly predict the precise amounts of operational tasks. To avoid this, we introduce an alternative approach that classifies services into distinct groups and try to match the property of the services to their counterpart without historical data and offering valuable insights.

doi:10.6342/NTU202401885

Chapter 2
Literature Review

2.1 Capacity Planning

In the service sector, capacity typically refers to the availability of personnel, facilities, and equipment to deliver the service, and thus proper capacity planning ensures that resources could reflect on the instability of the demand and are not wasted at any time[40]. With the improvement of technology, the service sector is now able to rely on automated machines to deliver their service to customers, which emphasizes the importance of equipment capacity planning. Actually, the pure service sector could encounter multiple problems due to the complex structure of service delivery, including the problem of product mix, the problem of setup time, and the problem of varying efficiency[10]. Therefore, the more automation is involved, the less factors are considered when it comes to decision-making in service delivery. Applications regarding to the development of capacity planning emerged recently due to recent technological advancements. This trend is exemplified by various studies employing advanced algorithms for predictive modeling in different domains. For instance, Su[34] utilized a combination of Recurrent Neural Networks (RNN) and Firefly Genetic Algorithms to develop a prediction model for electric vehicle charging piles, aiming to optimize power supply and minimize waste. Similarly, Qiu et al[27]. introduced a model that leverage Genetic Algorithms and RNNs for efficient utilization of parking space capacity. Jung et al. [43] applied Long Short-Term Memory (LTSM) network for forecasting in solar photovoltaic systems, focusing on capturing temporal patterns in South Korea. This growing body of research addresses capacity challenges across various sectors, benefiting from cutting-edge technological advancements. In our study, we will delve into the

doi:10.6342/NTU202401885

intricacies of IoT message passing and explore the capabilities of deep learning, situating our research within this context of advanced computational techniques.

2.2 Combination of IoT message and Deep Learning

Capacity planning in the service sector has long been challenging in various type of services, including capacity-constrained services, hospital resource planning, and appliance rental services[20]. Previous studies used mathematical solution to locate the problem of capacity planning, including minimum staffing[1] and ICU capacity management [17] in hospital resources. This article introduces deep learning techniques to address the issue of the limitation on mathematical solutions, improving the predictive precision of the actual usage patterns.

Multiple strategies related to dynamically adjusting the allocation of resources also serves as an important role when dealing with this type of problem[44]. Some research also emphasized on the strategies to react to the temporary down of services[4]. Nevertheless, with IoT message passing devices attached, emergencies could be noticed in real time and further action could be taken place shortly, which allows the resource reallocation being more efficient and rapid.

2.3 Deep Learning for Time Series Analysis

Ideally, service firms should adjust capacity according to demand fluctuation[26]. Emphasizing the seasonality factor, it becomes evident that demand patterns require time series analysis for accurate forecasting. Therefore, we aim to apply deep learning techniques and time series analysis on IoT message-passing washing machines to forecast future demand.

Deep learning techniques for time series analysis have gained significant attention over an extended period. The Artificial Neural Network (ANN), comprising

doi:10.6342/NTU202401885

interconnected computing units, has been instrumental in addressing a myriad of challenges[29]. Each computing unit in the network processes data and passes it to subsequent units, facilitating continuous learning. The advent of ANNs have paved the way for comprehending intricate patterns, reducing the reliance on expert-driven data interpretation. Their adaptability and self-learning capabilities offer a robust alternative to traditional time series forecasting models, underscoring their utility in dynamic data environments[12].

In addition, Convolution Neural Network (CNN), which is a specialized class of deep learning model, has become a popular modeling technique to resolve even more complex data input, such as image, audio and video[3]. They leverage spatial hierarchies and local connectivity patterns, making them exceptionally proficient at tasks like image recognition and classification.

Recurrent Neural Network (RNN) has also become a widely used deep learning model that has the ability to recognize sequences of data, such as text, speech, or time series[45]. Unlike traditional feedforward networks, RNNs possess loops allowing information to persist, enabling them to maintain memory of previous input. However, due to issues like the vanishing gradient problem, variants like Long Short-Term Memory (LTSM) have been developed to address their limitations[37]. LTSM are a specialized type of RNN designed to alleviate the long-term dependency problem inherent in traditional RNN. It uses gating mechanisms, namely input, output and forget gate, to regulate the flow of information, enabling them to better learn and remember over long sequences without being significantly affected by the vanishing or exploding gradient problem[14].

Sequence-to-sequence (Seq2Seq) autoencoders have emerged as a powerful technique in time series analysis due to their ability to capture complex temporal

doi:10.6342/NTU202401885

dependencies and generate meaningful representations of sequential data[6]. When integrated with Long Short-Term Memory (LSTM) networks, Seq2Seq models significantly enhance the prediction accuracy and robustness of time series forecasting[21]. LSTMs, known for their capability to retain long-term dependencies, work harmoniously with autoencoders to manage and encode sequential data into a latent space effectively[18]. This combination allows for the reconstruction and prediction of time series data with high fidelity, enabling more accurate and reliable forecasting.

2.4 Clustering Algorithms

In service firms where historical data is scarce, traditional time series analysis may not be feasible. We propose an alternative deep learning-based clustering method to address this limitation. Conventional time series clustering prioritizes similarity measures and feature extraction, making it susceptible to noise and potentially overlooking local shifts. Incorporating deep learning techniques, such as Autoencoders, enhances noise reduction and feature extraction capabilities[2]. Furthermore, the inherent capacity of deep learning to discern complex time series patterns paves the way for more sophisticated analytical approaches. In the following section of this paper, we delineate our problem and introduce deep learning-based methods to address our problem.

Traditional clustering approaches such as K-means, K-medoids, and Fuzzy C-means, fall under the category of Partitioning Methods. While these methods are characterized by their simplicity, ease of implementation, and agility, they do come with a drawback. Specifically, the number of clusters must be predetermined, necessitating multiple experiments and often requiring expert judgement for optimal results. The random selection of initial centroids can lead to poor convergence or convergence to local minima, affecting the overall quality of the clustering. Moreover, conventional clustering methods often assume that the clusters are spherical and evenly sized, which makes it less effective

doi:10.6342/NTU202401885

for clusters of different shapes and densities.

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Application with Noise) is a clustering algorithm that extends DBSCAN by converting it to a hierarchical clustering algorithm. It's particularly well-suited for dealing with clusters of varying density and is robust in the presence of noise[28]. HDBSCAN builds upon DBSCAN by using a hierarchical tree of clusters, allowing it to find arbitrary shaped clusters and exclude points in low-density areas[23]. In the utilization of IoT messages, density clustering approach should gain more visibility on the data structure and may generate more robust clustering model by strategically remove noise information.

After exploring various approaches to group machine usage pattern into clusters, we would attempt on partitioning clustering such as K-means, and Density-based clustering method such as HDBSCAN, to apply robust clustering techniques and group machines with its given features, We will delve deeper into the handling of IoT messaging data and the construction of modeling frameworks, with the aim of deriving meaningful insights from the results.

doi:10.6342/NTU202401885

Chapter 3

Problem Description

In this chapter, we aim to describe the problem of our research by examining the preprocessing of raw data as a foundational step, followed by a discussion on feature extraction techniques that serve to augment the predictive capacity of our model. Subsequently, a comparative analysis of various deep learning architectures is presented, measuring their respective advantages and limitations to elucidate their applicability. Finally, the chapter introduces deep clustering strategies as innovative solutions for service sectors confronted with the challenge of limited historical dataset.

3.1 Raw Data Preprocessing

With the integration of IoT messaging devices, we can acquire real-time data specific to each transmitted signal. The data is collected in JavaScript Object Notation (JSON) for each record. To harness the potential of this IoT data, preprocessing is essential to convert the raw information into a format compatible with deep learning models. For instance, Table 3-1 shows the concept of identifying single operational run from the IoT collected data, when analyzing data from IoT-enable message passing devices, we identify the completion of an operational run by detecting a status change from Running to Ready. By organizing the raw data chronologically, we can pinpoint accurate operational tasks and filter out irrelevant records and erroneous status changes due to network failure to enhance the training process's feasibility as well as the data quality.

Additionally, for consistency in analysis, we standardize the time column values to the "HH:MM:SS" format and arrange them chronologically. In our study, we have collected specific temporal-scale of raw data from IoT-enable washing machines. By monitoring status change signal from these devices, we are able to track each operational

doi:10.6342/NTU202401885

cycle in detail down to second. This capability allows us to leverage the data for advanced deep learning techniques aimed at achieving the desired outcomes. Specifically, the clarity and robustness of this data enhance its utility in applications such as deep learning time series analysis and deep clustering algorithms, where clear, consistent patterns are crucial for effective analysis.

Table 3-1 Example status changed detection mechanism, machine with status changed from running to ready is consider one operational run

Time	Machine ID	Previous Status	Current Status	Run Completion
2023-06-30 22:57:07	XXXXXXXX	Ready	Running	False
2023-06-30 23:37:33	XXXXXXXX	Running	Ready	True

3.2 Feature Extraction

To effectively use preprocessed data in deep learning models, it is imperative to generate additional features that enhance the model's ability to discern the time series patterns of our target. This necessitates feature extraction, which augments the data with detailed attributes related to individual operational runs. Prior to this, we need to ascertain the granularity of our data analysis. For example, if usage patterns remain relatively consistent on a weekly basis, we introduce a 'weekday' column to note the specific day of each run. While if usage patterns fluctuate throughout a whole year, we may use a monthly column to denote the seasonal differences. Moreover, the precision of IoT messaging data, accurate to the second, prompted us to add another column reflecting the time period within a day. For example, we can segment a day into four intervals, each spanning six hours, enabling a robust analysis of the washing machine's usage behavior.

The concept of seasonality in analyzing IoT data can vary significantly depending on the nature and context of the data being examined. Different IoT devices and their

doi:10.6342/NTU202401885

operational environments dictate the need for distinct seasonal definitions to accurately capture relevant patterns. For instance, for an industrial machine in a manufacturing setting, the 'seasons' might be better defined in terms of production cycles or maintenance schedules, which could span different timeframes, such as quarterly or biannually. Additionally, in environments where human activity plays a crucial role, such as in residential energy usage, the definition of seasonality might incorporate social patterns like holidays, school calendars, or typical vacation periods. Each of these scenarios demands a tailored approach to defining seasonality, ensuring that the time series analysis aligns with the specific operational rhythms and external influences impacting the IoT data. This customization allows for more accurate and meaningful insights, as it aligns the data analysis with the real-world factors that most significantly affect the device's usage patterns. Therefore, determining the appropriate seasonal definition is a critical step in preprocessing IoT data for deep learning models, as it directly influences the model's ability to detect and learn from the most pertinent patterns in the data."

Last but not least, additional features including the contractor information of the machine, number of years this machine has been employed, and the location where this machine is placed. Due to the fact that we could not determine which feature mostly affects the usage pattern of the machine, different aspects of attribute should be imported to ensure comprehensive analysis has been conducted. Furthermore, we can group the machine by location and calculate the sum of the machine in a single location to understand the supply and improve capacity planning process.

To effectively incorporate the additional attributes — contractor information, years of machine employment, and location — with the previously collected time series data, a multifaceted analytical approach is required. The time series data, which primarily tracks the usage patterns of the machines over time, provides a dynamic view of how these

doi:10.6342/NTU202401885

machines are operated. By integrating this data with the additional static attributes, we can gain a more holistic understanding of the factors influencing these usage patterns. For instance, combining time series data with the number of years a machine has been employed can reveal trends in machine efficiency or maintenance needs over time. Similarly, analyzing usage patterns in conjunction with the machine's location can uncover geographical influences on machine utilization. The contractor information adds another layer, potentially highlighting how different contractors' operational strategies impact machine usage. To achieve this integration, advanced data analysis techniques such as multivariate time series analysis or machine learning models can be employed. These methods can handle the complexity of combining time-based patterns with static attributes, allowing for a more comprehensive analysis. This integrated approach not only aids in understanding the usage patterns more deeply but also enhances capacity planning processes by identifying key factors that influence machine utilization across different locations and operational contexts.

When analyzing temporal features, it is crucial to account for the varying operational periods of real-world machine. In the IoT service sector, the dynamic updating of machine allocations frequently occurs, which must consider to prevent data preparation from compromised by differing recording time slots. Figure 1 illustrates the duty periods of various IoT machines. The data collection process can be further complicated not only by system shutdowns but also by the distinct onboard time of the machines, making it challenging to achieve perfect IoT message collection.

doi:10.6342/NTU202401885

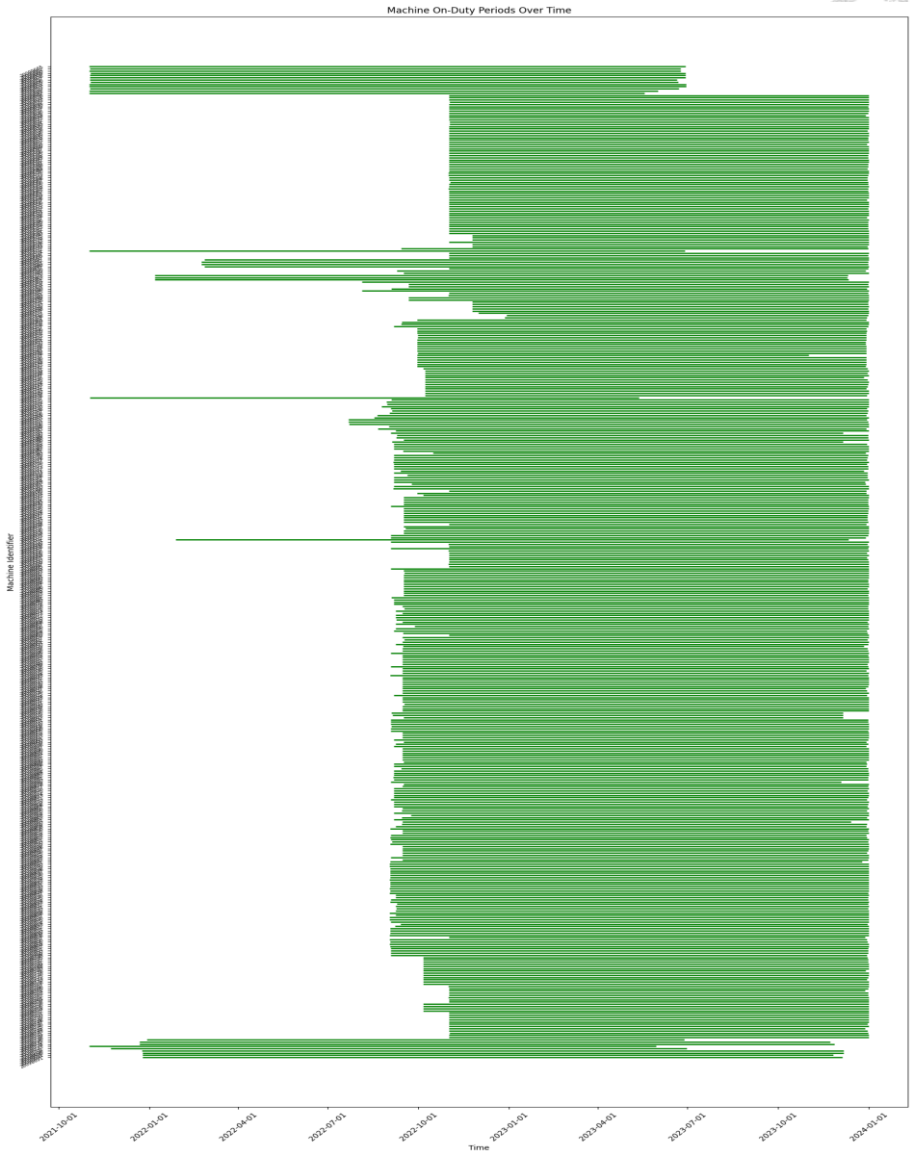


Figure 1. Real world IoT message-passing device actual operational time frame, x axis represents the date range that the machine is onboard, and each y tick stands for one single machine identifier.

3.3 Time Series Model Selection and Evaluation

In this section, we explore deep learning models to evaluate their performance on training and validating time series data, aiming to achieve optimal predictions of subsequent time periods. The efficacy of each model will be determined using specific machine learning metrics, which will be elaborated upon later in this chapter.

doi:10.6342/NTU202401885

3.3.1 Time Series Data Model Selection

Tree-based deep learning models, such as decision trees, random forests, and gradient boosted trees, each offer unique advantages and challenges in the context of time series data. A notable strength of these models is their interpretability. The transparent decision-making framework of tree-based models provides valuable insights into the importance of individual features. However, tree-based models do not fit for the times series data with complicated features and thus are not included in this study.

In our analysis, we consider neural network architectures, specifically artificial neural networks (ANNs) and convolutional neural networks (CNNs). Due to their strong computational power, these networks can efficiently handle vast data sets, allowing intricate usage patterns to be discerned.

Finally, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LTSMs) play an integral role in our study. RNNs themselves have the strong ability to process input data and are also capable of maintaining previous information captured. However, to overcome long dependency problems associated with RNNs, LTSMs are employed to address gradient-related problems inherent to RNNs.

As we mentioned in Chapter 2, the mechanism of RNN involves recognizing sequences by maintaining a memory of previous inputs through their looped connections. This implies that each input in an RNN is processed in the context of the preceding inputs, creating a chain of dependent operations. However, RNN's ability to maintain information from earlier in the sequence diminishes over time, which leads to vanishing gradient problem. On the other hand, LSTMs, while following the sequential processing principle of RNNs, introduce a more sophisticated approach through their unique gating mechanisms. The input, output, and forget gate allow LTSM to selectively remember and forget information, thus efficiently managing the flow of data through the network. This

doi:10.6342/NTU202401885

means that LSTMs can maintain relevant information over longer sequences and discard non-essential data, addressing the long-term dependency issue inherent in traditional RNNs. Consequently, while both models process data sequentially, LSTMs provide a more nuanced control over what is remembered and what is discarded, making them better suited for tasks where long-term contextual information is crucial.

Furthermore, we will incorporate autoencoders into our framework to effectively capture crucial information from our dataset. Utilizing an encoder-decoder structure, autoencoders can filter and identify important features while minimizing reconstruction loss[5]. While IoT message-passing machine often generate vast amounts of feature data to record machine properties and transactional information, autoencoders focus on essential elements that influence our predictions, thus enhancing the model's efficiency and accuracy.

Our approach involves generating different datasets, evaluating the foundational performance of each model, and examining insights they generated. Subsequently to this preliminary assessment, we will fine-tune model parameters, guided by observed data patterns, to maximize their performance.

3.3.2 Time Series Data Model Evaluation

For architectures like RNNs and LTSMs, which are tailored to time series data, it is crucial to use evaluation metrics that consider their spatial hierarchies and temporal dependencies. Therefore, Mean Squared Error (MSE) and Mean Absolute Error (MAE) are appropriate for continuous output, as they assess the model's ability to capturing trends and magnitudes within the time series data.

$$\text{MSE (Mean square error)} = \left(\frac{1}{n}\right) \Sigma (Y_i - \hat{Y}_i)^2 \quad (1)$$

$$\text{MAE (Mean absolute error)} = \frac{1}{n} \Sigma \mid Y_i - \hat{Y}_i \mid \quad (2)$$

doi:10.6342/NTU202401885

where n is the number of data, Y_i stands for the predicted value and the $\hat{Y}_i$ stands for the actual value.

To effectively evaluate the performance of deep learning models on regression task, particularly with architecture like RNNs and LSTMs tailored to time series data, it is essential to choose metrics that capture both the temporal dependencies and spatial hierarchies inherent in the data. MSE and MAE are foundational for assessing trends and magnitudes, which allow deep learning model assessment through comparison.

3.4 Clustering Data Model Selection and Evaluation

Service firms with limited historical data can leverage deep embedding time series clustering techniques to discern usage patterns closely aligned with their operations. Initially, deep clustering categorizes the data into well-defined groups, facilitating the discovery of these patterns.

3.4.1 Clustering Data Model Selection

Once clusters are established, cluster labels could be defined and help further analysis new onboarded IoT machines to predict their future usage patterns. This process required two essential phases of actions to fulfilled our expectation: **(1) Firstly, performing unsupervised clustering to create clusters and generate labels; (2) Secondly, employed these labels yielded from the first phase in a supervised deep learning classification to categorize new machine within the existing clusters.** Subsequently, the average number of operational runs for the machine can be calculated, providing a predictive measure for new machine over a specified timeframe. This method allows service firms to utilize existing dataset attributes to predict operational activities, thereby empowering them with predictive insights as new IoT machines are integrated.

As we mentioned in Chapter 2, traditional and recent clustering approaches both

doi:10.6342/NTU202401885

have the ability to successfully conduct clustering on time series data. K-means and K-medoids are praised for their expediency but are hindered by the prerequisite to predetermine the number of clusters, which often lead the model to stuck in the local minima. Moreover, these traditional clustering methods are focus on the numerical representation of data, other meaningful categorical features could not be included in the clustering dataset. On the other hand, given the ability to handle sequence data, RNN and LTSM can be employed for clustering by extracting features or representation from time series and form embedded data, which can then be clustered using traditional methods.

In addition to conventional methods, we could employ deep clustering methodologies such as HDBSCAN (Hierarchical Density-Based Spatial Clustering of Application with Noise) and Autoencoders combination. While HDBSCAN transform time series data into hierarchical tree cluster representation in order to find varying density between each data point, Autoencoder compresses the data into a compact, lower-dimensional space. Without knowing which approach is better for resolving our problem, we expect to employ the approaches mentioned above and attempt to understand the essence of our data.

Each approach mentioned above brings unique perspectives to handling time series data. For instance, K-means and K-medoids require a predetermined number of clusters and typically work with vectorized representations of time series data, while RNN and LTSM first extract features or representations from time series before these features can be clustered using traditional methods. HDBSCAN, elevate this further by transforming time series data into hierarchical tree cluster representations, especially useful for sequences of varying lengths. Autoencoders, on the other hand, compress data into a more compact, lower-dimensional space to relief curse of dimensionality and simplify the input data that would fit into the clustering model. Each method fundamentally alters the nature

doi:10.6342/NTU202401885

of the input data, making it more amenable for clustering in a transformed feature space.

3.4.2 Clustering Data Model Evaluation

For the service sectors lack historical data, we employed various evaluation metrics corresponding to clustering methods to assess the quality and relevance of discovered patterns. The Silhouette Coefficient measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation), which is crucial for assessing the appropriateness of the cluster assignments, especially when the number of clusters is not predefined.

Additionally, the Davies-Bouldin Index evaluate clustering validity by measuring the average similarity between clusters, where lower values indicate better clustering[8]. For deep embedding clustering approaches like autoencoders, the reconstruction error provides insight into the compressed representations.

To further classified target machine without historical data, we introduce supervised classification to take the clustering result as target variable and conduct classification. In supervised classification tasks, evaluation metrics such as tend to examine the amounts of labels that have correctly be predicted. Evaluation metrics including Accuracy, Recall, Precision, and F1-score are variables that examine the model performance. Accuracy measures the proportion of correctly predicted instances among the total instances. Recall evaluates the model's ability to identify all relevant instances. Precision calculates the proportion of true positive predictions among all positive predictions. F1-score is the harmonic mean of Precision and Recall, providing a single metric that balances both concerns[25]. These metrics offer comprehensive insights into the effectiveness and reliability of the model.

doi:10.6342/NTU202401885

Chapter 4

Model Building

In this chapter, we aim to further process our dataset to include more features in order to fit our models. Different methodologies of time series forecasting and deep learning clustering are introduced separately by the detailed steps of processing data and conducting model training and evaluation to provide solution for a large scale IoT formed dataset.

4.1 Feature Engineering

First of all, before feeding dataset into deep learning model, we should retrieve all meaningful information regards to our data as described below in this subsection.

4.1.1 Feature Selection

We aim to realize the essence of IoT passed dataset and add other features based on the machine identifier in this study. While trying to combine every useful column in our dataset, such as location and machine properties, it is essential that every feature in our generated dataset is not correlated to other features in order to avoid linear correlation. For instance, if both the location identifier and the location property columns contain nearly identical values, indicating that each location possesses unique property, it may be cautious to either eliminate the location property column or seek an alternative feature. This alternative should offer distinct value variations and align with domain-specific knowledge.

Since IoT machine lending services often require collaboration with several contractors to boost demand across large geographical areas, we suggest including data related to these contractors in our dataset as well. By incorporating the columns that

doi:10.6342/NTU202401885

capture contractor-specific attributes, deep learning models can effectively discern variations in user perceptions and the condition of equipment provided by different contractors.

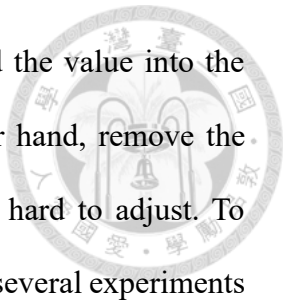
While categorical features often require one-hot encoding, which can lead to the curse of dimensionality, numerical features are recommended to enhance the richness of the dataset[15]. Therefore, we can enhance our dataset by implementing feature engineering techniques that generate numerical columns, thereby enriching the data variety. For instance, we can aggregate IoT-enabled machines based on their its location identifiers to count the total amount of machines in each location. This metric can then be used to gauge the demand for each machine based on its location density. Additionally, given that our framework accommodates varying usage patterns of machines, it is crucial to record the actual operating duration of each machine as a feature. By capturing the timestamp of the initial and final operational signals from the IoT devices within a specific timeframe, we can calculate the total active period in days for each machine.

In addition to the feature mentioned above, it is essential to incorporate external data that could influence machine usage patterns, thereby capturing information beyond the control of service provider and the contractors. For instance, service providers with IoT-equipped devices located outdoors may experience a decrease in demand during severe weather events. Accordingly, leveraging domain knowledge, we have integrated weather-related features such as precipitation and the number of rainfall days into our dataset as numerical variables to further enrich our analysis.

4.1.2 Machine Identifier

For the machine identifier, we should carefully conduct experiment on the influence when different decision is chosen. The effect that we add or remove the machine identifier could cause significant impact on the way that the model learning the patterns. Due to the

doi:10.6342/NTU202401885



fact that the machine identifier is not a general feature, simply feed the value into the model may mislead the model's view of the data[16]. On the other hand, remove the machine identifier may let the model facing unstructured data and hard to adjust. To decide whether the model needs this column or not, we've conducted several experiments to know whether it is proper to add it into our dataset. We suggest to use mechanism to help the model realize the machine identifier in the first place, which could help the model avoid misunderstanding the meaning of this column.

4.2 Time Series Dataset preparation

After we transformed the raw data to records that could be identified as a single operational run, or a unit of usage, more features should be utilized to help our model recognizes specific patterns. The reason that we introduced features such as weekday and multiple periods within a day previously in Chapter 3 would be addressed here.

Different from original time series analysis which use date as a feature, service sector with IoT message passing machine equipped may not gain a large number of usage amount in a single day. In deep learning field, an insufficient amount of data could lead to overfitting, where a model learns the training data too well, including the noise and outliers, and fails to generalize to new, unseen data.

On the other hands, recent research has demonstrated that deep learning prediction mechanisms have significantly improved inventory forecasting[30]. Both the retail and food industries can utilize these advancements by recording the number of parts or materials consumed in specific time slot and applying deep learning techniques to predict future needs. Similarly, data captured from IoT message-passing devices, which share data patterns with inventory prediction, can be integrated into this framework. Instead of tracking the quantity of parts or materials consumed, these devices record the number of operational runs within specific time slots. This approach allows us to leverage existing

doi:10.6342/NTU202401885

deep learning models and achieve accurate prediction outcomes.

Therefore, we abandon the date feature, instead of making each data point down to a single day, we tend to aggregate the time series data by summing it with different temporal feature to feed model more comprehensive data, and compare different permutation to choose the best data representation referred to our dataset.

Data captured by IoT machines can be detailed down to minute, it is an advantage to further divide several patterns within a day to gain more information from the analysis as well. In addition, in real world condition, the data obtained from IoT machines may not always be consistent. Several reasons such as maintenance, system downtime, or machine not onboard at the same time could cause segmentation when it comes to IoT message passing devices. Therefore, we should pick suitable timeframe for our captured IoT data when conducting time series analysis. The criteria of choosing the right timeframe for our data is to ensure enough amount of data could be recorded and avoid unnecessary complexity. In our framework, we aggregate each period of data based on the temporal features mentioned above to make every single record representative. After appropriate time slice chosen, we aggregate the data simply by summing the records with the same temporal features.

After generating the aggregated datasets, we then connect each dataset from different aspects and pack specific number of records into sequences, which also referred to as time steps in deep learning mechanism. Due to the fact that we would further combine specific number of records into sequence to help fit into the model, there are several approaches to align the records. By arranging the desire order of records and binding specific amounts of records prepared as time steps for model fitting process, different permutations generate distinct time series prediction model. When different scope of temporal feature existed in the dataset, the way we arrange the records stands for different aspect while the

doi:10.6342/NTU202401885

model processed the data. Table 4-1 shows the example arrangement of time series dataset to illustrate our thought.

Table 4-1. Time Series Dataset illustration, the records vary from period first, then goes weekday and year

Machine Identifier	Year	Weekday	Period Number	Number of runs	Other Features
Machine A	1st	Monday	1	19	value
Machine A	1st	Monday	2	20	value
Machine A	1st	Monday	3	32	value
Machine A	1st	Monday	4	53	value
Machine A	1st	Tuesday	1	17	value
Machine A	1st	Tuesday	2	24	value
Machine A	1st	Tuesday	3	33	value
Machine A	1st	Tuesday	4	68	value
Machine A	1st	Wednesday	1	29	value
Machine A	1st	Wednesday	2	31	value
Machine A	1st	Wednesday	3	62	value
Machine A	1st	Wednesday	4	65	value
...	...	...	...	...	...
Machine N	4th	Sunday	1	6	value
Machine N	4th	Sunday	2	0	value
Machine N	4th	Sunday	3	5	value
Machine N	4th	Sunday	4	26	value

4.3 Deep Clustering Dataset preparation

As mentioned in Section 3.4, to allow IoT devices without historical data could benefits from our approaches, two phase of model building require to complete this task. Below we would first address on the first phase unsupervised clustering and illustrate our concept with a table. Next, we would explain how to conduct second phase supervised classification to achieve our goal.

4.3.1 First Phase Unsupervised Clustering

For clustering data, first of all, we conduct data transformation process to every machine using identical technique as the previous time series data. To allow most

doi:10.6342/NTU202401885

insightful records prepared for our clustering model, we align the weekday feature and period feature into columns to represent the real usage patterns in specific time frame. Table 4-2 visualizes our concept of the clustering dataset, the column represents each temporal features combination, where the row indicates the machine ID, and each record stands for the number of operational runs in specific timeframe for a single machine. The ultimate goal is to use one record to represent individual machines, therefore, we apply aggregation on the different time frame of a single machine. For instance, integrating temporal feature into columns to stands for the usage patterns of a machine.

However, when too many temporal features are chosen, the resulting larger number of columns may cause dimensionality curse, leading to increased computational complexity and a higher risk of overfitting, especially when the dataset is not proportionately large. Therefore, we transform appropriate number of temporal features into columns, tailored to the specific requirements and the characteristic of the dataset. To avoid the temporal feature combination leads to dimensionality curse or the number of records not in proportion to the number of features, a practical approach involves computing the mean of the number of occurrences within distinct time frame. This transformation enables the dataset to adopt a new representation whose records encapsulate the usage pattern of an individual machines. Consequently, this facilitates the ability of time series clustering models to discern and classify distinct patterns of machine usage.

To enrich the information contains in the data representation, in additional to numerous features that constituted by temporal features indicating the number of operational runs, we apply feature engineering to captured deterministic factors that could successfully distinguish the usage condition of individual machines. Due to the fact that solely using numerical features that stands for the number of operational runs may not

doi:10.6342/NTU202401885

yield desire output while conducting clustering, categorical variables that points out the characteristic of the machines could contribute to this end. However, linear clustering algorithm have its limitation regarding to categorical variables and may lose the advantages when sparse feature included. Therefore, we see this dataset in different aspect and would explain more in the next paragraph.

Table 4-2. Clustering Dataset Illustration, each row stands for one single machine, and the columns indicates the number of operational runs in specific temporal feature

Machine identifier	Monday period 1	Monday period 2	Monday period 3	Monday period 4	Tuesday period 1	Tuesday period 2	temporal features	Sunday period 3	Sunday period 4
Machine 1	20	16	26	45	18	18	…	30	46
Machine 2	4	3	6	19	5	3	…	6	18
Machine 3	13	4	12	28	12	2	…	15	26
Machine 4	6	10	18	41	6	7	…	28	47
Machine 5	6	7	16	41	6	4	…	24	41
Machine 6	12	10	28	45	10	6	…	34	46
Machine 7	8	6	12	32	4	5	…	18	35
Machine 8	6	5	13	31	6	4	…	19	36
Machine 9	6	11	24	37	6	9	…	27	37
…	…	…	…	…	…	…	…	…	…
Machine N	5	4	6	24	6	2		7	20

4.3.2 Second Phase Supervised Deep Learning Classification

Additionally, as discussed in Chapter 3, IoT machines lacking historical data required two phases of implementation to calculate the average number of operational runs in specified timeframe. Once the desired clusters and labels are determined, these labels can be reassigned to their corresponding machine identifier. Consequently, the dataset would then include the machine identifier, the number of operational runs during specific periods, and other categorical variables that describe the machine's properties. Meanwhile, cluster labels, which grouped machines with similar usage patterns, are recorded, marking the completion of the first phase of implementation.

With the clustering labels and features we have introduced, the second phase supervised deep learning classification process can now accurately assign new machine

doi:10.6342/NTU202401885

to their respective categories based on the result. These results illustrate the aggregated usage patterns of IoT message-passing machine with the same cluster, which are then used to calculate the mean operational runs across various temporal features. Consequently, this approach enables us to identify groups of machines with similar usage patterns based solely on machine properties. This capability provides the service sector with predictive insights right from the initial state.

4.4 Time Series Model building and Evaluation

In the era of Big Data, the variety and volume of data collected have reached unprecedented level. While high-quality, large volume datasets can provide deep learning architectures with valuable information, facilitating more accurate outputs, traditional forecasting approaches often struggle to efficiently process the diverse and massive datasets generated by modern big data techniques and IoT devices[9]. This phenomenon necessitates a reevaluation of conventional machine learning algorithms and offers a new perspective on data analysis. As such, there is a growing need to innovate and adapt these models to better handle the complexity and scale of data, particularly with regard to IoT message passing and real-time data processing. This chapter will explore the limitations of traditional models and introduce alternative strategies that are more suitable to the demands of Big Data and IoT environments.

4.4.1 Time Series Forecasting Model building

While dealing with time series data, recent work generally use date as timestamp to predict the value of the target variable[19]. In our approach, we recommend aggregating the number of operational runs using specific temporal features to eliminate the date column from the dataset. By doing so, we aim to process the data with a more comprehensive perspective.

doi:10.6342/NTU202401885

Figure 2 and Figure 3 illustrate one machine usage distribution using date as a timestamp and one machine usage pattern aggregated by temporal features, respectively. By comparing the data distribution before and after aggregation, we can infer that the data distribution in Figure 4-2 is likely to achieve better performance during training due to its clearer patterns. Thus, we believed that using specific time slots to record the number of runs and conduct time series prediction can yield more accurate outcomes.

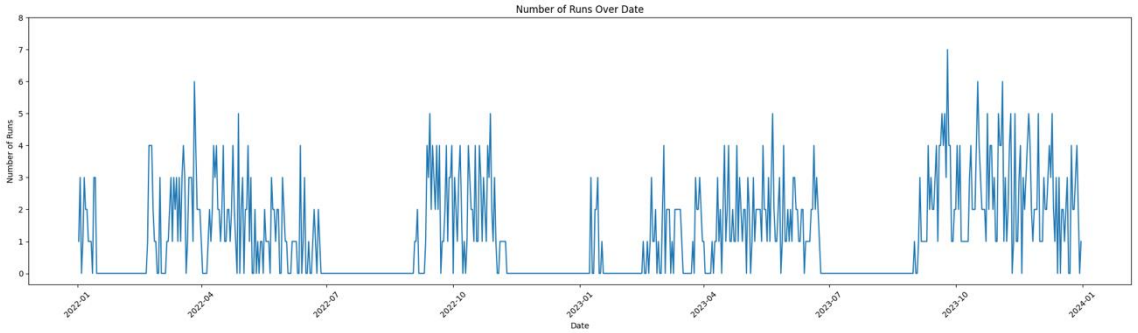


Figure 2. Machine usage patterns shown by date

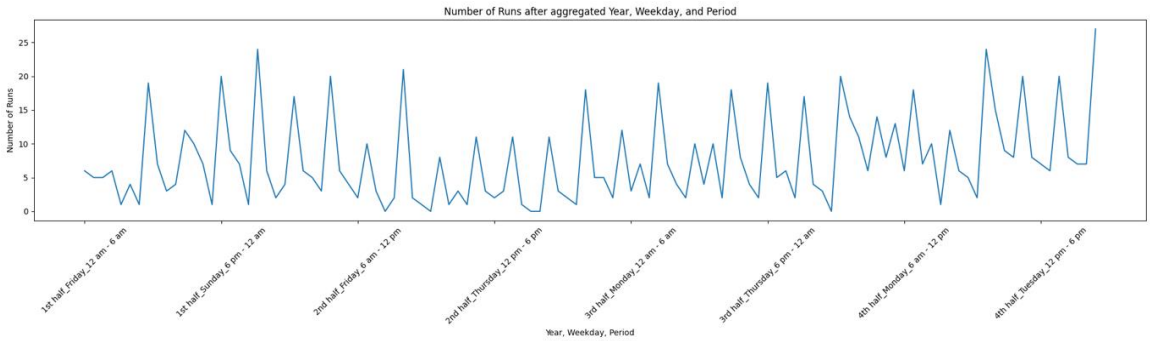


Figure 3. Machine usage patterns after aggregate year, weekday and period

Additionally, time series data requires defining time steps before fitting into models to determine how many records to consider during training process[35]. Because different arrangements of temporal features can lead to distinct training and prediction performances, our research accommodates time series data that may include periods of system shutdowns, machine maintenance, or the onboarding of new machines at different times. Thus, determining the time steps parameter during the training process is crucial

doi:10.6342/NTU202401885

for helping the model recognize machine usage patterns effectively.

In the approach that rearrange data sequence with different time step configured, it is essential to identify the smallest unit that can represent machine usage patterns. For instance, if the dataset is structured by fixed year, fixed month and varying weekday values, the desired time steps should encompass at least one week's worth of records without exceeding or crossing over into another month.

In time series analysis domain, several deep learning models have proved own strong capability to process enormous amount of data and minimize errors. As we have stated in Chapter 2, RNN, LSTM, GRU, are all powerful deep learning models when it comes to time series analysis. With the proposed aggregation method, new formed data appears clear and meaningful patterns that is suitable to fit in above-mentioned time series analysis models. However, the visualization of machine patterns distribution only present one single machine. With hundreds of lines, namely, hundreds of machines required to be processed during training, the data that will fit in the model is actually much more complex.

Autoencoder, employed encoder-decoder architecture, is good at denoising and dimensionality reduction[7]. Furthermore, sequence-to-sequence autoencoder, designed for handling sequence data, combines the capabilities of autoencoder and recurrent neural network like LSTMs[11]. This architecture is particularly useful for time series prediction tasks because it can effectively capture temporal dependencies in the data while also learning a compressed latent representation. Figure 4 demonstrates the concept of autoencoder integrated with LSTM.

In order to process the huge amounts of records and columns result from aggregating the data with several temporal features and the native machine properties, we can utilize the advantage of Autoencoder to learn a compressed latent representation that captures

doi:10.6342/NTU202401885

the essential features and patterns in the time series data. In our study case, deep learning time series models are not capable of dealing with the enormous amount of usage patterns collected from the IoT message-passing devices. Therefore, without providing baseline approach to compare with our proposed framework, we would demonstrate the model building and evaluation step with autoencoder-employed architecture. We list the experiment results that we have conducted in the Appendix.

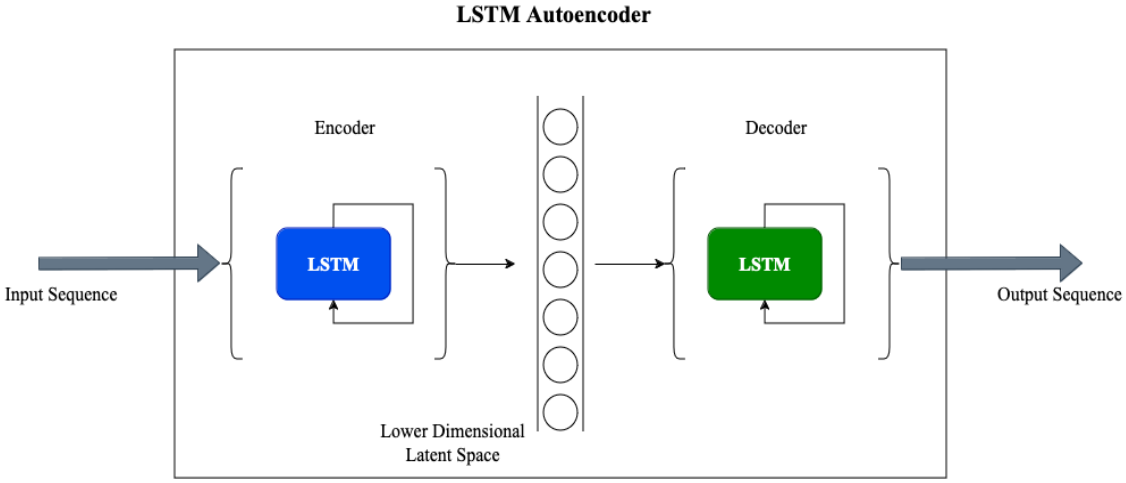


Figure 4. LSTM autoencoder architecture diagram

4.4.2 Time Series Forecasting Model Evaluation

For time series data, we explore a variety of evaluation metrics tailored for time series forecasting models in Chapter 3.4. Key metrics such as MAE, MSE are utilized to assess the performance of the selected models.

Moreover, RMSE, which served large errors more costly and attempting to minimize them, also play an important role here. The calculation of this metrics leaves more penalty on the larger errors by squares the errors before averaging, which makes it more sensitive to larger errors. Last but not least, RMSE is widely used in regression tasks to measure the standard deviation of prediction errors or residuals, providing a clear measure of

doi:10.6342/NTU202401885

model accuracy. The formula of RMSE is listed as follows.

$$\text{RMSE (Root mean square error)}=\sqrt{\text{MSE}}=\sqrt{\left(\left(\frac{1}{n}\right)\sum(Y_i-\hat{Y}_i)^2\right)} \quad (3)$$

Where n is the number of data, Y_i stands for the predicted value and the $\hat{Y}_i$ stands for the actual value.

We leverage the advantages of MAE and RMSE in our analysis to compute the residuals between the predicted values and the true values. While being effective in evaluation, we should also be aware of the limitations of these metrics, such as less robustness for outliers and over-penalization of large errors. Thus, both metrics would be employed to evaluate our model forecasting values to help assess the result objectively.

4.5 Deep Clustering Model Building and Evaluation

In this section, we focus on the IoT message passing machine that lack historical operational data, to perform two phase tasks: (1) unsupervised clustering model building and (2) supervised deep learning classification to provide result for time series forecasting.

4.5.1 Deep Clustering Model Building

4.5.1.1 First Phase Unsupervised Clustering

Data generated by IoT devices typically consists of complex logic and numerous features derived from machine operations. Beyond the basic properties of machines, data may contain concealed information. In our work, we aim to transform temporal features related to operational runs into structured columns. This approach attempts to enrich the data with detailed insights into the actual conditions of machine usage, thereby increasing the complexity of data representation.

Traditional clustering algorithms, such as K-means and Gaussian Mixture Models

doi:10.6342/NTU202401885

(GMM), primarily analyze numerical features to measure the similarity among data points, which are then grouped around centroids[38]. These methods typically model data distributions within a two-dimensional latent space. While effective for certain datasets, these traditional techniques often assume the data is linearly separable and normally distributed, which can limit their applicability to complex real-world data that may exhibit non-linear patterns and multi-modal distributions[42]. To address these limitations, more sophisticated approaches should be involved to incorporate dimensionality reduction techniques or kernel methods to capture deeper insight from the data structure.

Density-Based Clustering methods, such as DBSCAN and HDBSCAN, operates on the principle of identifying dense region of data points[13]. These regions are considered clusters, while sparser areas are treated as noise. These approaches are particularly good at handling dataset where clusters are not uniformly distributed. HDBSCAN, unlike DBSCAN, which requires the specification of a globally density threshold (eps)[33], it builds a hierarchy of clusters and doesn't need a predefined distance threshold. Besides, a minimal spanning tree is employed by calculating the mutual reachability distance between all pairs of points, and further build the hierarchical tree for clustering.

In the service sector, which frequently handles complex and disorganized data, traditional partitional clustering methods such as K means and GMM may not effectively distribute machines usage patterns. Additionally, partially collected time series data often contain extraneous information that would mislead the model while conducting clustering. Consequently, we would utilize a Density-Based clustering method to determine the optimal number of clusters for our data. We will then compare these results with those obtained from conventional clustering methods and provide a detailed justification for our choice.

While choosing the number of clusters that will be separated by the clustering

doi:10.6342/NTU202401885

models, we should be cautious to decide a proper number of clusters that would then become labels in the second phase classification task. For partitioning clustering methods like K-means, which requires predefined number of clusters, we suggest minimizing the number of machines within the cluster to help decrease the loss while conduct average to yield the predicted usage pattern. A cluster may contain five to twenty machines as most to ensure the predicted result would still make sense even conduct mean on several machines. In addition, for the density-based clustering method, we would not need to predetermine the number of clusters in advance, instead, we let the model decide what would be the optimal number of clusters with one noise cluster that we tend to remove from our dataset.

General service sector that has employed IoT message passing equipment, often contains large amounts of organized properties that could be used as features in our clustering framework. Following the feature engineering process described in the previous section, the prepared dataset may suffer from curse of dimensionality and inefficiency. Moreover, the clustering model has struggled to accurately assign data points to the appropriate clusters and to distribute the data evenly.

To address these challenges, Deep Embedding Clustering (DEC) has demonstrated significantly improved results when handling complex data structure[41]. Our approach incorporates an embedding layer that utilized Autoencoder technique within our clustering process, aiming to transform the data to lower-dimensional representation. Initially, we construct an Autoencoder to compress the high-dimensional data into a dense, embedding space. This transformation not only reduces dimensionality but also enhances the clustering process by highlighting intrinsic data pattern more effectively. Subsequently, we apply density-based clustering method to organize the embedded data into distinct groups, optimizing both the feature extraction and clustering phases

doi:10.6342/NTU202401885

concurrently for superior accuracy and efficiency.

4.5.1.2 Second Phase Supervised Deep Learning Classification

In the second phase of our supervised classification process, we have opted for tree-based deep learning models to their significant advantages. Tree-based models are proficient in handing both numerical and categorical variables, demonstrate robustness against outliers, and are highly interpretable[24]. These features are crucial given the diverse and segmented operational records introduced in Chapter 4.1 and our two phases implementation strategy for machine without historical data. Moreover, the interpretability of tree-based models allows us to analyze feature importance post-training, ensuring the accuracy of our classification task based on domain knowledge.

4.5.2 Deep Clustering Model Evaluation

As for clustering data, we treated two phases model performance measurement separately, since each phase employed different deep learning models to conduct the training process.

4.5.2.1 First Phase Unsupervised Clustering Evaluation

Partitioning clustering tasks usually focus on the mathematical evaluation metrics to measure the geographical distance, for instance, Euclidean distance, between centroid and each data point. Silhouette Score is another robust metrics used to assess the effectiveness of partitioning clustering method like K-means. It evaluates how well each data point fits within its cluster compared to other clusters, quantifying the compactness and separation of clusters[32].

$$\text{Silhouette Score} \quad s(i)= \frac{b(i)-a(i)}{\max(a(i), b(i))} \tag{4}$$

where a(i) is average distance from i to all other data points in the same cluster and b(i) stands for minimum average distance from i to points in a different cluster, for all

doi:10.6342/NTU202401885

clusters i is not a member.

Evaluating density-based clustering methods like HDBSCAN often involves specific metrics tailored to assess cluster quality in the context of variable density patterns. Metrics such as Silhouette Score may be less effective due to HDBSCAN's handling of noise and varied cluster densities, potentially leading to low score even with good clustering. Alternatively, the Davies – Bouldin Index and Calinski – Harabasz Index are more advantageous as they consider intra-cluster distances and inter-cluster separation[22]. However, their effectiveness can still vary depending on the dataset's structure and the inherent distribution of data points. The Davies – Boudlin Index and Calinski – Harabasz Index equation is listed below:

$$\text{Davies-Boudlin Index (DB)} = \left(\frac{1}{s}\right) \Sigma \max(R_{ij}) \quad (5)$$

$$\text{Calinski-Harabasz Index (CH)} = \frac{\Sigma\, n_i \cdot \|c_i - c\|^2 / (k-1)}{\Sigma\, \Sigma\, \|x - c_i\|^2 / (n-k)} \quad (6)$$

where s is the number of clusters and R_{ij} is a measure of dissimilarity between cluster I and the cluster most similar to i. For the Calinski-Harabasz Index, k is the number of clusters, n_i is the number of points in the i-th cluster, c_i is the centroid of the i-th cluster, c is the overall mean of the data, and n is the total number of points.

4.5.2.2 Second Phase Supervised Deep Learning Classification

As we mentioned in Chapter 3, the second phase of time series prediction suite for IoT message-passing machine without historical operational records, could be classified to specific group of machines that we have already separated with the first phase method. Therefore, tree-based deep learning classification models would then be introduced to help our cluster assign process. In deep learning classification field, evaluation metrics mainly focus on the difference between predicted label and the actual label. The metrics typically involved Accuracy, Precision, Recall, and F1-Score, each catering to different

doi:10.6342/NTU202401885

aspects of model performance. Accuracy measures overall correctness, but might not be reliable alone, especially in imbalanced dataset. Precision and Recall provide insights into the model's performance regarding positive class predictions, while F1-Score harmonizing the balance between Precision and Recall, making it crucial for datasets where false negative and positive are equally costly[39].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (7)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (8)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (9)$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

where TP (True Positives) represents cases correctly identified as positive; TN (True Negatives) represents cases correctly identified as negative; FP (False Positives), also known as Type I error, involves cases incorrectly labeled as positive; and FN (False Negatives), known as Type II error, involves cases incorrectly labeled as negative.

4.5.2.3 Two Phase Evaluation Trade off

Since two phases required to assign similar machine group to the new onboards IoT device, the performance of both models should take into consideration together. As we mentioned in Chapter 4.3, the number of clustering defined is essential to make the process more robust. Both the clustering model and the classification model should perform well enough to achieve the usage pattern prediction for IoT machine without historical data. However, the contradictory of the clustering loss and the classification accuracy make it difficult to successfully choose the right models as we observed that while assign more clusters to minimize the average loss effect, the label increase in classification would cause decreasing accuracy. We recommend setting threshold for each phase task to ensure both procedures are implemented correctly. For instance, for the

doi:10.6342/NTU202401885

unsupervised clustering task, the baseline training and testing loss, and the Silhouette Score can be recorded to compare with further advance methodologies with partitioning clustering methods. It is worth to notice that Silhouette Score applied to density-based clustering methods may leads to misunderstanding of the clustering result. For the classification task, about 80% above accuracy and 70 % above recall are acceptable for the supervised classification model in IoT data classification[36]. While both the model passed two phases of evaluation, we could then calculate the mean number of operational runs in specific periods to finish our time series forecasting process.

doi:10.6342/NTU202401885

Chapter 5
Experiment Result

In this section, we apply the model building procedure introduced in Chapter 4 to a real case, a group of IoT-enabled washing machines in the laundromat industry. These machines, equipped with sensors, are primarily located in college dormitories across various regions of Taiwan. We analyze approximately 700 machines over a two-year period, applying time series analysis and a deep clustering approach introduced in Chapter 4.

5.1 Feature Selection

In both time series and deep clustering approaches, relevant features regarding IoT-enabled washing machine, no matter numerical or categorical variables, should be integrated within our dataset. Therefore, we leverage existing variables which is yielded from the IoT message-passing process and domain knowledge regarding to the laundromat industry to form our informative dataset.

In our framework, specific amounts of temporal features and their arrangements are crucial to conduct a comprehensive work on machine usage pattern prediction and machine clustering. It is worth to mention that correctly slice specific time frame with the collected data is the key point to yield accurate forecasting result during model training. In our case, since the duty periods of each washing machine vary a lot, an inconsistent length of records occurs in our collected data. We slice two-year period of data from each machine to ensure enough information contained in the dataset. Furthermore, we grouped each half year data and assign feature “year” to represent 4 distinct values: 1st half, 2nd half, 3rd half, and 4th half.

In IoT laundromat industry, the usage pattern varied every single day. We have

doi:10.6342/NTU202401885

employed three kinds of temporal feature to capture the usage pattern difference between each temporal dependency: year, weekday and period. For period feature, we decided to divided four periods a day, each expanding six hours, to distinguish the usage pattern within a day. Figure 5 shows the usage pattern of seven days a week, four periods a day drawn of a washing machine by a bar plot with each color stands for a different period.

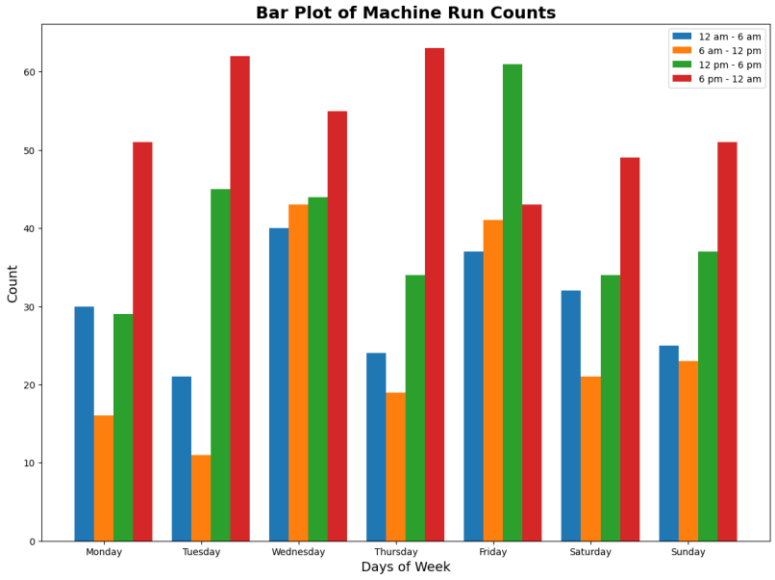


Figure 5. The Usage Pattern with Regards to Number of Operational Runs within 7 Weekdays and 4 Periods a Day.

Since the fluctuated rate of equipment utilization of washing machine in laundromat industry may lead to decrease profit, we aim to produce forecasting based on each of the four periods within a day. Consequently, companies could take action to address the low usage period and minimize machine ideal time. Furthermore, we have included variables such as contractor's information, the physical properties of the machine, and the weather information of each location based on domain knowledge. Table 5-1 shows the features that have be employed during our model training process.

doi:10.6342/NTU202401885

Table 5-1. Features in our dataset, including temporal features, location, contractor information, and weather

Variables	Description	Type
data_cid	Machine identifier	integer
year	Indicate which half year time slice was the operation run happened	categorical, 4 options
weekday	Indicate which weekday was the operation run happened	categorical, 7 options
period	Indicate which period within a day was the operation run happened	categorical, 4 options
location	Physical location of the machine	categorical, 60 locations
duty period	Whole period each machine has been operated	numerical, unit day
machine count	Number of machines within the same location	numerical
machine type	Brand or type of the washing machine	categorical, 6 types
contractor id	Contractor identifier with machine equipped in their facility	categorical
floor	Physical floor the machine is equipped	categorical
property	Indicate the machine location properties such as indoor/outdoor, etc	categorical
gender	Gender of the user within specific location	categorical
city	Large geographic area that the machine is resided in	categorical
avg_temperature	Average temperature of the city in 2 years, record each half year	numerical, float
precipitation	Precipitation of the city in 2 years, record each half year	numerical, float
rel_humidity	Relative humidity of the city in 2 years, record each half year	numerical, float
rainfall_days	Rainfall days of the city in 2 years, record each half year	numerical, integer
sun_duration	Sunshine duration of the city in 2 years, record each half year	numerical, float

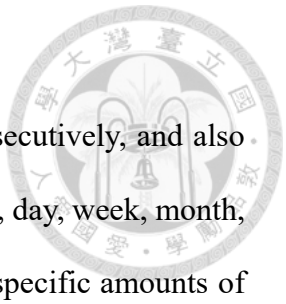
5.2 Time Series Model Performance Evaluation

For time series model experiment, we have prepared four types of arrangement of temporal features, each pattern provides distinct information during training process. We would first illustrate the different arrangement of the data, and then demonstrate the model training result tables.

5.2.1 Dataset Characteristic

Since we have observed different data arrangement, namely, different sequence of records would be packed with specific time steps, could cause distinct training outcome. Therefore, we demonstrate four types of dataset arrangement and their training result to prove our idea.

doi:10.6342/NTU202401885



(1) Contiguous periods

The first way is to align the smallest scale temporal feature consecutively, and also sort the other temporal features in an ascending order, e.g., period, day, week, month, quarter, etc. By using this kind of arrangement, we further pack specific amounts of records to feed in the model so that the model would receive the smallest scale of data arrangement. In this approach, we expect the model to realize the usage pattern in a concentration aspect, namely, the model would be more focus on the present usage and try to learn the usage pattern of the near future. The advantage of this approach is that the model learns from the contiguous usage pattern of the small-scale timeframe. However, the model lacks of the ability to realize seasonality pattern and even the usage pattern of data in the far future.

Table 5-2. Contiguous Period Dataset illustration, you can see the records vary from period, weekday, year

Machine Identifier	Year	Weekday	Period Number	Number of runs	Other Features
Machine A	1st half	Monday	1	19	value
Machine A	1st half	Monday	2	20	value
Machine A	1st half	Monday	3	32	value
Machine A	1st half	Monday	4	53	value
Machine A	1st half	Tuesday	1	17	value
Machine A	1st half	Tuesday	2	24	value
Machine A	1st half	Tuesday	3	33	value
Machine A	1st half	Tuesday	4	68	value
Machine A	1st half	Wednesday	1	29	value
Machine A	1st half	Wednesday	2	31	value
Machine A	1st half	Wednesday	3	62	value
Machine A	1st half	Wednesday	4	65	value
…	…	…	…	…	…
Machine N	4th half	Sunday	1	6	value
Machine N	4th half	Sunday	2	0	value
Machine N	4th half	Sunday	3	5	value
Machine N	4th half	Sunday	4	26	value

doi:10.6342/NTU202401885

(2) Weekday based series

In this approach, we suggest the meaningful feature, weekday, to be the temporal feature that varied first. Based on the domain knowledge of the laundromat industry, weekday feature appears to be the main difference of the machine usage pattern. For the other temporal features, we again align them in an ascending order to help the model focus on the contiguous change on the weekday so that the model emphasizes more on what we are caring about. The pros of this representation enforce the model to learn the main variation of the machine usage, while unseen data emerged, the model would own the ability to know the completely different pattern that exist in the real condition. The cons of this approach are similar to the Contiguous period approach, where the model could not gain visibility of the whole timeframe and know the real usage condition in the data we provided.

Table 5-3. Weekday Based Dataset illustration, the records vary from weekday first

Machine Identifier	Year	Weekday	Period Number	Number of runs	Other Features
Machine A	1st half	Monday	1	19	value
Machine A	1st half	Tuesday	1	17	value
Machine A	1st half	Wednesday	1	29	value
Machine A	1st half	Thursday	1	30	value
Machine A	1st half	Friday	1	34	value
Machine A	1st half	Saturday	1	23	value
Machine A	1st half	Sunday	1	19	value
Machine A	1st half	Monday	2	20	value
Machine A	1st half	Tuesday	2	24	value
Machine A	1st half	Wednesday	2	31	value
Machine A	1st half	Thursday	2	31	value
Machine A	1st half	Friday	2	34	value
...	...	...	...	...	...
Machine N	4th half	Thursday	4	36	value
Machine N	4th half	Friday	4	13	value
Machine N	4th half	Saturday	4	13	value
Machine N	4th half	Sunday	4	26	value

doi:10.6342/NTU202401885

(3) Half year scope prediction dataset

For half year scope prediction, we arrange the data by the largest scope temporal feature, for instance, year, half year, quarter. Furthermore, align the other temporal features in descending order. This kind of data would provide the model with large scale data realization, to let the model understand in large timeframe, the variation of number of operational runs. We expect that if the model knows the whole timeframe data in each training batch, it gains the ability to understand the usage condition more thoroughly.

Table 5-4. Half year scope Dataset illustration, the records vary from year, weekday, and then period

Machine Identifier	Year	Weekday	Period Number	Number of runs	Other Features
Machine A	1st half	Monday	1	19	value
Machine A	2nd half	Monday	1	24	value
Machine A	3rd half	Monday	1	21	value
Machine A	4th half	Monday	1	15	value
Machine A	1st half	Tuesday	1	17	value
Machine A	2nd half	Tuesday	1	14	value
Machine A	3rd half	Tuesday	1	24	value
Machine A	4th half	Tuesday	1	18	value
Machine A	1st half	Wednesday	1	29	value
Machine A	2nd half	Wednesday	1	17	value
Machine A	3rd half	Wednesday	1	27	value
Machine A	4th half	Wednesday	1	21	value
...	...	...	...	...	...
Machine N	1st half	Sunday	4	19	value
Machine N	2nd half	Sunday	4	15	value
Machine N	3rd half	Sunday	4	19	value
Machine N	4th half	Sunday	4	26	value

doi:10.6342/NTU202401885

(4) Half year scope prediction dataset – varied continuous period

Last but not least, we could arrange data with the contiguous period firstly, then align other temporal features in a descending way. For instance, the data aligned would vary the period first. After iterated through the option of period, the next sequence of data would change the largest scope. This approach suggests compromising the ability to know the comprehensive view of the dataset and the variation of the smallest timeframe, which means that we are trying to balance the pros and cons exist in the way to arrange data records.

Table 5-5. Half Year Scope Dataset, varied period Dataset illustration, the records vary from year, period, finally weekday

Machine Identifier	Year	Weekday	Period Number	Number of runs	Other Features
Machine A	1st half	Monday	1	19	value
Machine A	2nd half	Monday	1	24	value
Machine A	3rd half	Monday	1	21	value
Machine A	4th half	Monday	1	15	value
Machine A	1st half	Monday	2	20	value
Machine A	2nd half	Monday	2	14	value
Machine A	3rd half	Monday	2	13	value
Machine A	4th half	Monday	2	18	value
Machine A	1st half	Monday	3	32	value
Machine A	2nd half	Monday	3	18	value
Machine A	3rd half	Monday	3	32	value
Machine A	4th half	Monday	3	24	value
...	...	...	...	...	...
Machine N	1st half	Sunday	4	19	value
Machine N	2nd half	Sunday	4	15	value
Machine N	3rd half	Sunday	4	19	value
Machine N	4th half	Sunday	4	26	value

We suggest to adapt different arrangement of dataset and conduct time series analysis model training separately so that we can understand the differences between distinct data representation on time series forecasting results.

doi:10.6342/NTU202401885

5.2.2 Time Series Model Performance Evaluation

The LSTM autoencoder architecture in our case including input layer, two LSTM layer for encoder and decoder, output Layer, and repeat vector. While the model is fed with sequence data from input layer with time steps defined, the LSTM encoder layer compress the input into lower dimensional latent representation, which will then go through a repeat vector and attempt to capture crucial features of the sequence. The repeat vector be processed by LSTM decoder layer to maintain the temporal structure. Last but not least, output layer applies dense transformation to each time step individually to reconstruct the sequence back to its original feature size.

Figure 6 illustrates the model architecture that we employ in our deep learning time series prediction framework. While the second parameter in the input sequence shape indicates time steps selection, we packed four records as a sequence to represent the half year scope prediction dataset usage patterns.

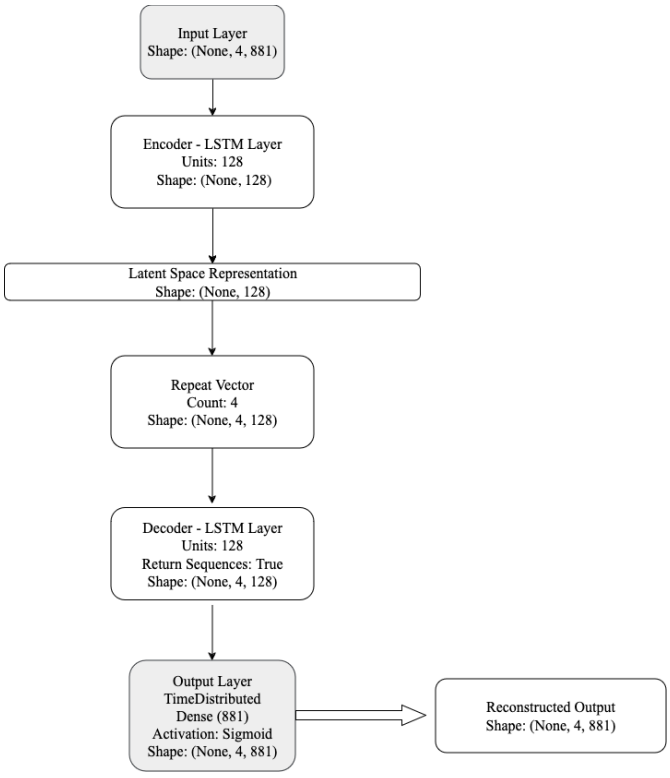
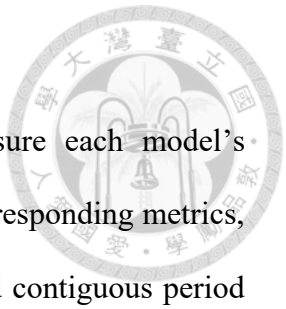


Figure 6. LSTM Autoencoder Time Series Prediction Model Architecture

doi:10.6342/NTU202401885



In the evaluation steps, we use MAE and RMSE to measure each model's performance. Table 5-6 shows the assessment of each model and corresponding metrics, demonstrate the actual performance of our experiments. We denoted contiguous period dataset as **DS1**, weekday-based dataset as **DS2**, half year scope dataset as **DS3** and half year scope – varied period dataset as **DS4**. The number of rows and columns of these datasets before encoded features using one-hot encoding is 40,684 rows and 19 columns, respectively. The time step column indicates the length of sequence that we group the records in the dataset, which may vary based on different dataset and needs. DS1 yielded MAE 0.022 and RMSE 0.106, which is good for regression forecasting task. DS2 improved the model prediction with lower MAE and RMSE. While we varied the largest temporal dependency half year first, which is represented by DS3 and DS4, the training and validation MAE enhance even more, which inferred that the comprehensive view of IoT devices data may contains information that helps the model training process. Besides, the relatively small value of time steps may also contribute to better performance.

Table 5-6. Assessment of each dataset with MAE and RMSE

Dataset	Time step	MAE	RMSE
DS1	28	0.0220	0.1060
DS2	7	0.0192	0.0934
DS3	4	0.0070	0.0578
DS4	4	0.0067	0.0588

Moreover, we have observed the training and validation loss drop from current condition after certain epochs. Figures 7 to 10 demonstrate the training and validation MAE of each dataset. We can see that the DS2 have a small drop on the training and validation MAE, while the DS3 and DS4 have significant drop of training and validation MAE. The decrease after certain epoch demonstrated by the red circle further improves

doi:10.6342/NTU202401885

the model performance and are considered escaping the local minima during training process, which indicate that the models have gain even more information after specific training epochs.

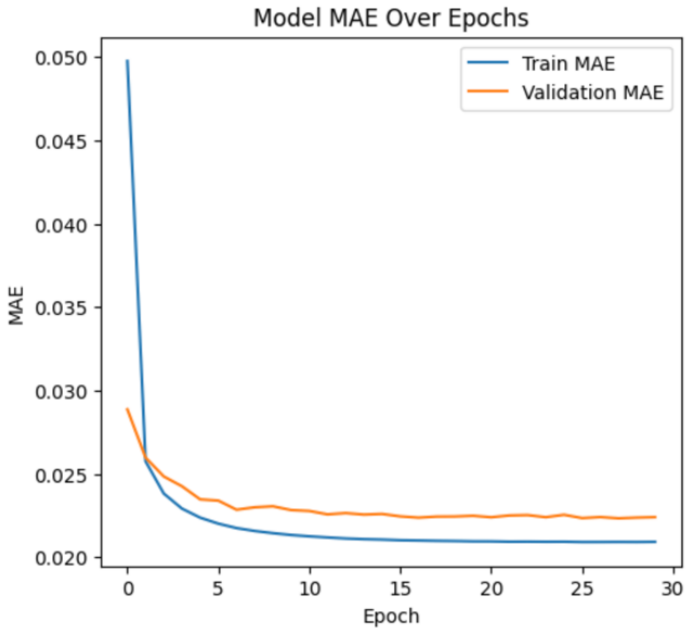


Figure 7. Training and Validation MAE -- DS1

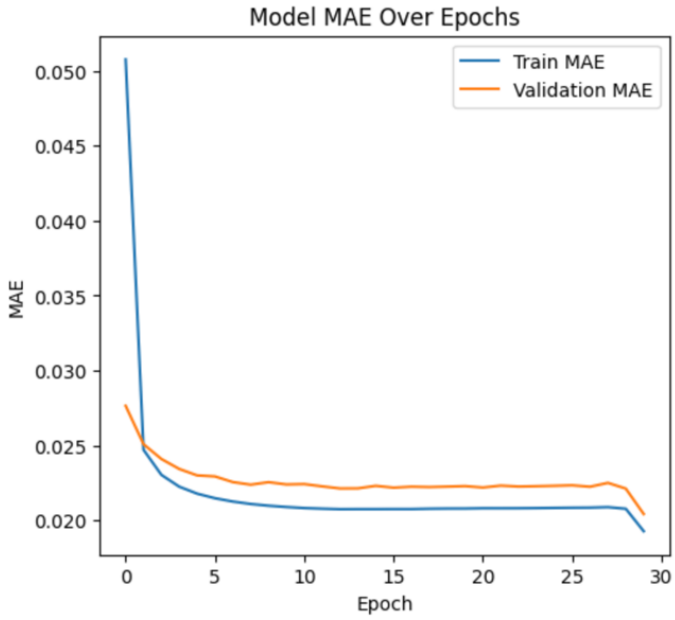


Figure 8. Training and Validation MAE – DS2

doi:10.6342/NTU202401885

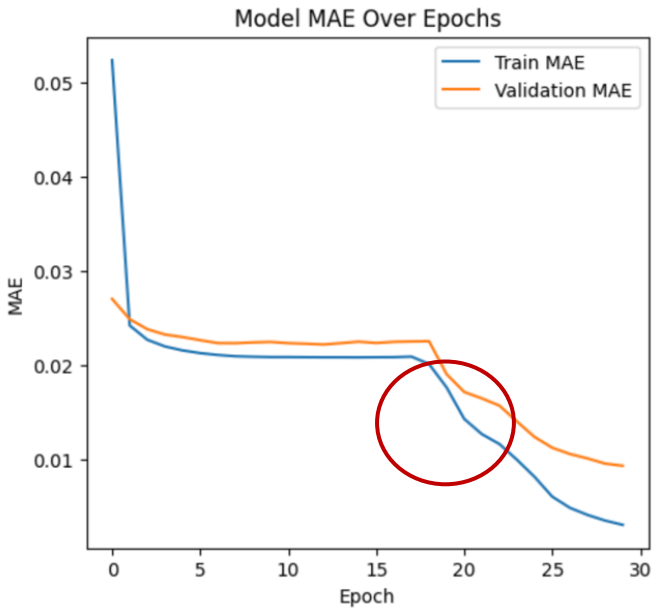


Figure 9. Training and Validation MAE – DS3

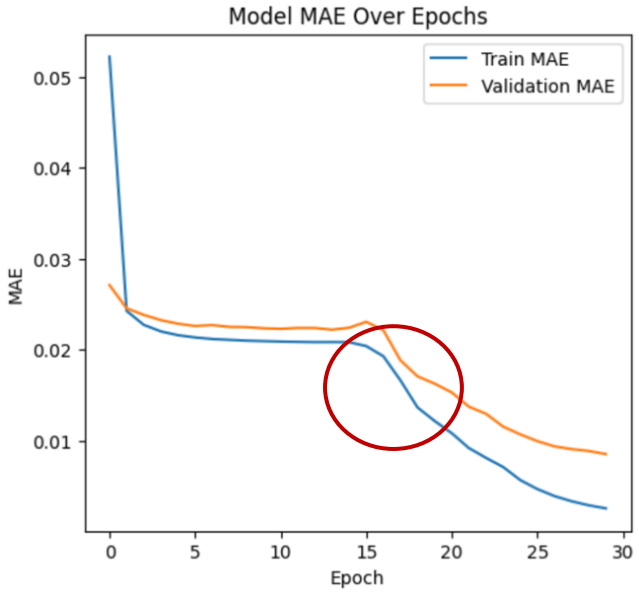


Figure 10. Training and Validation MAE – DS4

doi:10.6342/NTU202401885

In our research, the complex and noisy data collected from IoT message-passing machines have precisely been addressed by the advantage of autoencoder handling noise data and learn essential information by minimizing the reconstruction loss during training process. Furthermore, we have discovered that different arrangement of data could lead to distinct model performance and forecasting precision. By aggregating time series data with certain temporal features, we could yield strong prediction capability with the proposed framework.

5.3 Deep Clustering Model Performance Evaluation

As for IoT-enabled washing machines lack historical data, with only the properties that we have demonstrated in section 5.1, deep clustering methods enable grouping machines into similar clusters. Furthermore, supervised classification task could define the target machine to specific group based on its original essences. Below we would introduce the two phases framework separately, aiming to provide a thorough procedure to conduct usage pattern forecasting with machines without historical operational data.

5.3.1 First Phase: Unsupervised Learning Model Performance

In the first phase, we aim to form dataset like Table 4-2, to represent single machine usage patterns within two years in one record. We convert the temporal features weekday and period to columns to indicate the temporal dependency of the machine in our time frame. For a machine with full four half year data, 4 records are yielded, each indicating the usage pattern within one half year. Since not every machine contains full time data, we calculate the average number of operational runs among each half year record, aiming to aggregate the machine usage pattern to present by a single row.

We have addressed the power of deep embedding clustering methods that combines traditional partitioning clustering algorithm or density-based clustering approaches with

doi:10.6342/NTU202401885

autoencoder to retrieve crucial information within lower latent space. Here, we utilized K-means and HDBSCAN with autoencoder, to conduct experiments on our dataset. While K-means requires predefined number of clusters to group machine around the centroid, HDBSCAN automatically distributed the data point in the perspective of density, and also help isolate the noise records within our dataset. We have conduct K-means clustering with three different number of clusters defined, and also apply grid search to fine tune the optimal structure of autoencoder before feed into HDBSCAN clustering model, the clustering result of Silhouette Score, Davies-Boudlin Index and Calinski-Harabasz Index are shown in Table 5-8.

In order to find a suitable number of clusters that could minimize the average loss while we calculate the mean of number of operational runs in the second phase, we conduct experiments and show the clustering performance with ten and twenty clusters. Since we have around 700 machines, each cluster should contain no more than 100 machines to minimize the loss while compute average number of operational runs.

We first list a distribution of the numbers of machines and its corresponding cluster when using K-means with autoencoder to conduct the clustering in Table 5-7. The large number of machines in one group may increase the loss while conducting average on the number of operational runs significantly, leading to poor forecasting result. Therefore, while choosing proper number of clusters to group the machines, few experiments are required to achieve the optimal result in distribution as well as the second phase classification.

doi:10.6342/NTU202401885

Table 5-7. Number of machines in each cluster using K-means with 20 clusters predefined.

Cluster	Number of Machines
Cluster 0	34
Cluster 1	22
Cluster 2	34
Cluster 3	60
Cluster 4	39
Cluster 5	17
Cluster 6	39
Cluster 7	25
Cluster 8	54
Cluster 9	54
Cluster 10	13
Cluster 11	51
Cluster 12	29
Cluster 13	19
Cluster 14	56
Cluster 15	52
Cluster 16	23
Cluster 17	24
Cluster 18	37
Cluster 19	28

Table 5-8. Clustering result assessment for K-means with different number of clusters defined and HDBSCAN with grid search fine-tuned optimal autoencoder structure.

Dataset	Silhouette Score	Davies-Boudlin Index	Calinski-Harabasz Index
K-means, 10 clusters	0.118	2.040	80.675
K-means, 20 clusters	0.103	2.095	59.499
K-means, 30 clusters	0.130	1.853	52.792
HDBSCAN, 31 clusters	-0.112	2.476	75.134

The clustering result have proved that while the number of defined clusters increase, the Silhouette Score and Davies-Boudlin Index are better performance, which infer that the encoded features that comes from the autoencoder require large number of clusters to

doi:10.6342/NTU202401885

distribute each data point. However, the Calinski-Harabasz Index decreases as the number of clusters increases, indicating that the difference between did not increase against the increasing number of clusters. Moreover, the result of HDBSCAN have automatically distribute 31 clusters in the first phase operation, which is more feasible while evaluating with Calinski-Harabasz Index.

Figure 11 illustrates the data point distribution while using HDBSCAN plot by the t-Distributed Stochastic Neighbor Embedding (t-SNE) representation, which is a dimensionality reduction technique particularly well-suited for the visualization of high-dimensional dataset. It maps multi-dimensional data to two dimensions for visualization. Without showing all 31 clusters data, the t-SNE plot illustrates the overview of data distribution performed by HDBSCAN. Although a large amount of noise point presented, we can see that most data point can correctly gather into the cluster with similar machine operation pattern in dimensional reduction format.

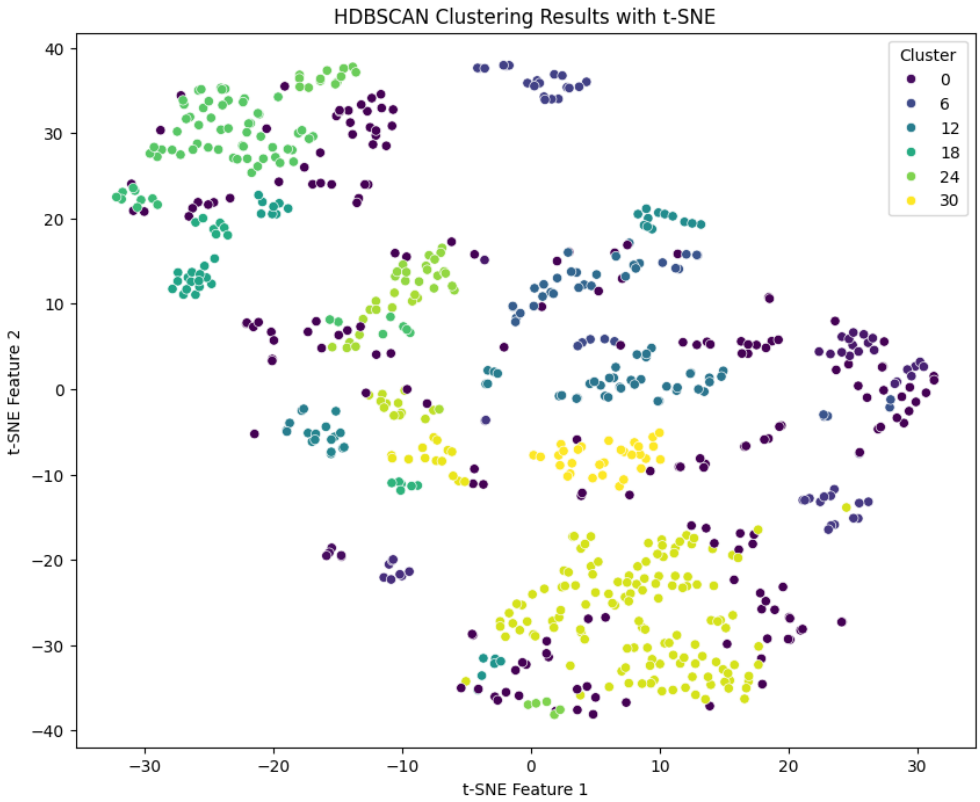


Figure 11. t-Distributed Stochastic Neighbor Embedding plot of HDBSCAN clustering

doi:10.6342/NTU202401885

5.3.2 Second Phase Supervised Learning Model Performance

In second phase evaluation process, we have conduct supervised classification on the result of clustering model from the first phase. Both random forest and XGBoost tree-based classification model are employed to understand the performance of the clustering results. We list the evaluation metrics including precision, recall, and F1-score in weighted average for each classification result to better understand each model's ability.

In addition, we have conduct cross validation in five folds to calculate the average accuracy of each model with random forest, presenting a robust evaluation on the performance of both tree models. The feature importance plot in classification results also demonstrate the features that determines the relation between value in each record and it label assigned.

Table 5-9.Classification performance of Random Forest and XGBoost apply on the clustering result in the first phase.

Dataset	Random Forest				XGBoost		
	Precision	Recall	F1-Score	Cross Accuracy	Precision	Recall	F1-Score
K-means, 10 clusters	0.87	0.85	0.86	0.75	0.83	0.82	0.83
K-means, 20 clusters	0.84	0.81	0.80	0.75	0.81	0.82	0.80
K-means, 30 clusters	0.77	0.73	0.73	0.70	0.81	0.76	0.77
HDBSCAN, 30 clusters (remove noise)	**0.92**	**0.93**	**0.91**	**0.89**	**0.94**	**0.93**	**0.93**

From the experiment result, the HDBSCAN with 30 cluster after removing the machine records in noise cluster performed the best. It yielded 0.89 on five folds cross validation from random forest model and 0.94 precision using XGBoost model. Therefore, we recommend use density-based clustering methods such as HDBSCAN while conduct

doi:10.6342/NTU202401885

clustering on complex IoT message-passing machine data. In addition, we find that the performance of the second phase task from random forest decrease while the number of clusters increase, infers that the trade-off between the number of clusters and the result of classification task exists.

Last but not least, with the aid of tree-based algorithms, it is possible to visualize the tree structure to know how the model process input data and make classification decisions. The feature importance plot shown in Figure 12 illustrates the notable features that have high score during the classification task, which help understand the variable's contribution during the training process. The duty period and location feature appear to be the deterministic element during the training process in our case, which also make sense while consider domain knowledge.

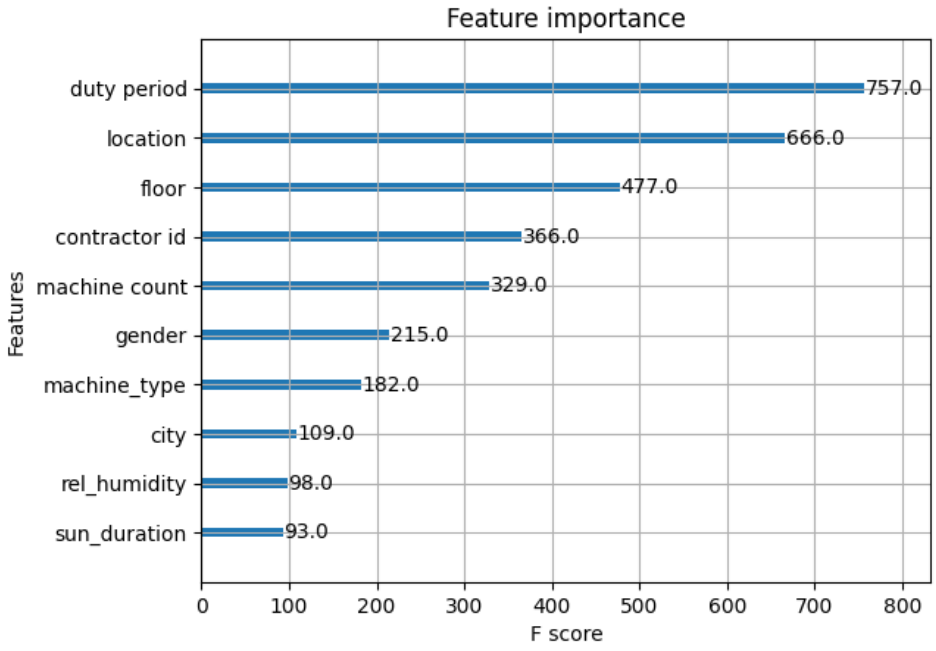


Figure 12. Feature Importance Plot

doi:10.6342/NTU202401885

Chapter 6
Conclusion

6.1 Conclusion

In this paper, we have explored various methods to optimize prediction outcomes from IoT-enabled collected data, acknowledging the complexities of data collection and processing due to varying data formats. A crucial aspect of enhancing our framework's forecasting capability is ensuring the long-term validity and accuracy of IoT data collection to reflect actual machine usage patterns. By contiguously integrating with new data, we can build comprehensive models that improve the robustness and accuracy of predictions.

We have proposed a framework that aggregates the data into specific temporal dependencies, enabling deep learning time series forecasting on IoT message-passing machines. This framework employs deep learning models and autoencoders, demonstrating the ability to handle complex IoT data of varying lengths and providing insights real-world collected raw IoT data.

For the machines lacking historical data, a deep embedding clustering approach combined with supervised classification methods forms a two-phase procedure to predict the future usage patterns based solely on machine properties and related features. This framework offers robust prediction outcomes, aiding the service sector in making data-supported business strategies.

Moreover, the application of this framework extends beyond business insights, offering significant benefits for human being. For example, implementing time series data forecasting on light bulbs equipped with sensors can predict the indoor periods of the elderly, enabling automatic lighting within precise time slots, enhancing safety without

doi:10.6342/NTU202401885

wasting electricity. Moreover, the allocation of electric vehicle supply equipment can also take advantage of our framework to avoid low occupy rate and high traffics in certain area.

In conclusion, the proposed framework not only provides the forecasting capabilities for service industries that utilize IoT-enabled machines but also leverages high-volume and high-quality data to offer forecasting advantages regardless of the presence of historical data. Beyond the service sectors, any field with IoT sensors and data collection mechanism can benefit from our framework, gaining greater visibility for future planning.

6.2 Managerial Implication

The integration of IoT data and advanced deep learning models presents numerous managerial implications across various sectors. In manufacturing, predictive usage patterns powered by IoT data provides forecasting ability on future machine usage., while accurate inventory forecasting ensures optimal spare parts availability. In healthcare, continuous patient monitoring through IoT sensors allows for early detection of potential health issues, improving patient outcomes and operational efficiency. Smart facility management optimizes the use of hospital equipment, leading to better resource utilization. Energy management benefits from IoT-enabled smart lighting systems that enhance energy efficiency by automating lighting schedules based on usage patterns, and predictive analytics optimize the placement and operation of electric vehicle charging stations, reducing congestion and improving user satisfaction. In the retail and supply chain sectors, IoT data-driven inventory forecasting reduces stockouts and overstock situations, while cold chain management ensures the quality of temperature-sensitive goods during transit. These applications demonstrate how leveraging IoT data for predictive analytics enhances decision-making, optimizes operations, and fosters innovation, ultimately driving business success and competitive advantage.

doi:10.6342/NTU202401885

6.3 Future work

We have explored various methods to achieve optimal prediction outcomes from the collected IoT messages. However, data collection and processing are complex tasks that require dynamical adjustments based on varying data formats. To enhance the forecasting capability of our framework, it is crucial to ensure that long-term IoT data collection is valid and accurately represents the actual usage patterns of individual machines. Consequently, by contiguously adding new data, we can build more comprehensive models with robust data quality, thereby improving the accuracy of our prediction outcomes.

Thanks to modern technology and computing power, pretrained models that can be applied to various data sources can be developed effectively. If a robust data collection procedure is established and sufficient data is prepared, it is recommended to build pretrained model based on our framework. With high-quality data, these pretrained models can achieve flexibility and be easily adapted for use in various domains.

For IoT machines without historical data, the deep embedding clustering approach allows these machines to be classified into clusters with similar usage patterns based solely on their properties. In our case, we compute the mean usage pattern from the machine within the same cluster to forecast the future usage pattern of the target machine. Alternatively, is it possible to build a deep learning time series forecasting model based on the data from machines in the assigned cluster. The prediction outcomes generated from this model should be more accurate and robust compared to simply computing the mean usage pattern of these machines.

doi:10.6342/NTU202401885

Reference

[1] Adenso-Díaz, B., González-Torre, P., and García, V., "A capacity management model in service industries," *International Journal of Service Industry Management,* vol. 13, no. 3, pp. 286-302, 2002.

[2] Alqahtani, A., Ali, M., Xie, X., and Jones, M. W., "Deep Time-Series Clustering: A Review," *Electronics,* vol. 10, no. 23, p. 3001, 2021.

[3] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., and Farhan, L., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data,* vol. 8, no. 1, 2021.

[4] Armistead, C. G. and Mapes, J., "Supply Networks and the Changing Role of Operations Managers," in Achieving Competitive Edge: Getting Ahead Through Technology and People, D. Bennett, C. Lewis, Eds. Springer, London, pp. 48-53, 1991."

[5] Baldi, P., "Autoencoders, unsupervised learning, and deep architectures.," in Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, PMLR 27, pp. 37-49, 2012."

[6] Borovykh, A., Bohte, S., and Oosterlee, C. W., "Conditional time series forecasting with convolutional neural networks.," *arXiv:1703.04691,* 2017.

[7] Dasan, E. and Panneerselvam, I., "A novel dimensionality reduction approach for ECG signal via convolutional denoising autoencoder with LSTM," (in English), *Biomedical Signal Processing and Control,* vol. 63, p. 102225, Jan 2021.

[8] Davies, D. L. and Bouldin, D. W., "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-1, no. 2, pp. 224-227, 1979.

[9] Elhanashi, A., Dini, P., Saponara, S., and Zheng, Q., "Integration of Deep Learning into the IoT: A Survey of Techniques and Challenges for Real-World Applications," *Electronics,* vol. 12, no. 24, p. 4925, 2023.

[10] Elmaghraby, S. E., "Manufacturing capacity and its measurement: A critical evaluation," *Computers & Operations Research,* vol. 18, no. 7, pp. 615-627, 1991/01/01/ 1991.

[11] Ghimire, S., Deo, R. C., Wang, H., Al-Musaylh, M. S., Casillas-Pérez, D., and

doi:10.6342/NTU202401885

Salcedo-Sanz, S., "Stacked LSTM Sequence-to-Sequence Autoencoder with Feature Selection for Daily Solar Radiation Prediction: A Review and New Modeling Results," *Energies,* vol. 15, no. 3, p. 1061, 2022.

[12] Zhang, G., Hu, M. Y., "Forecasting with artificial neural networks:: The state of the art," *International Journal of Forecasting,,* vol. 14, no. 1, pp. 35-62, 1998.

[13] Kriegel, H., Kröger, P., Sander, J., Zimek, A., "Density-based clustering," *WIREs Data Mining and Knowledge Discovery,* vol. 1, pp. 231-240, 2011.

[14] Hochreiter, S. and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[15] Jia, W., Sun, M., Lian, J., and Hou, S., "Feature dimensionality reduction: a review," *Complex & Intelligent Systems,* vol. 8, no. 3, pp. 2663-2693, 2022/06/01 2022.

[16] Karingula, S. R., Ramanan, N., Tahmasbi, R., Amjadi, M., Jung, D., Si, R., Thimmisetty, C., Polania, L. F., Sayer, M., Taylor, J., and Coelho, C. N., "Boosted Embeddings for Time-Series Forecasting," in *Machine Learning, Optimization, and Data Science*, Cham, G. Nicosia *et al.*, Eds., 2022// 2022: Springer International Publishing, pp. 1-14.

[17] Kim, S.-C., Horowitz, I., Young, K. K., and Buckley, T. A., "Analysis of capacity management of the intensive care unit in a hospital," *European Journal of Operational Research,* vol. 115, no. 1, pp. 36-46, 1999/05/16/ 1999.

[18] Lai, G., Chang, W.-C., Yang, Y., and Liu, H., "Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks," presented at the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 2018. [Online]. Available: https://doi.org/10.1145/3209978.3210006.

[19] Liu, X. and Wang, W., "Deep Time Series Forecasting Models: A Comprehensive Survey," *Mathematics,* vol. 12, no. 10, p. 1504, 2024.

[20] Pullman, M., "Capacity management for hospitality and tourism: A review of current approaches,," *International Journal of Hospitality Management,,* vol. 29, no. 1, pp. 177-187, 2010.

[21] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G., "LSTM-based encoder-decoder for multi-sensor anomaly detection," *arXiv:1607.00148.,* 2016.

doi:10.6342/NTU202401885

[22] Mary, S. A. L. S., An, A. N., Rani, M. U., "Cluster validity measures dynamic clustering algorithms.," *ARPN Journal of Engineering and Applied Sciences,*, vol. 10.9: 4009-4012., 2015.

[23] MCINNES, L., et al., "hdbscan: Hierarchical density based clustering," *J. Open Source Softw.,* 2017.

[24] Moshkovitz, M., Yang, Y.-Y., and Chaudhuri, K., "Connecting Interpretability and Robustness in Decision Trees through Separation," presented at the Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2021. [Online]. Available: https://proceedings.mlr.press/v139/moshkovitz21a.html.

[25] Mrabet, M. A. E., Makkaoui, K. E., and Faize, A., "Supervised Machine Learning: A Survey," in *2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 3-5 Dec. 2021 2021, pp. 1-10, doi: 10.1109/CommNet52204.2021.9641998. [Online].Available:https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=9641998&ref=

[26] Ng, I. C. L., Wirtz, J., and Sheang Lee, K., "The strategic role of unused service capacity," *International Journal of Service Industry Management,* vol. 10, no. 2, pp. 211-244, 1999.

[27] Qiu, J., Tian, J., Chen, H., and Lu, X., "Prediction Method of Parking Space Based on Genetic Algorithm and RNN," Springer International Publishing, 2018, pp. 865-876.

[28] RAHMAN, M. F., et al., "Hdbscan: Density based clustering over location based services.," *arXiv: 1602.03730,* 2016.

[29] S Agatonovic-Kustrin, R. B., "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis,* vol. 22, no. 5, pp. 717-727, 2000.

[30] Seyedan, M., Mafakheri, F., and Wang, C., "Order-up-to-level inventory optimization model using time-series demand forecasting with ensemble deep learning," *Supply Chain Analytics,* vol. 3, p. 100024, 2023/09/01/ 2023.

[31] Shemwell, D. J. and Cronin, J. J., "Services Marketing Strategies for Coping with Demand/Supply Imbalances," *Journal of Services Marketing,* vol. 8, no. 4, pp. 14-24, 1994-12-01 1994.

doi:10.6342/NTU202401885

[32] Shutaywi, M. and Kachouie, N. N., "Silhouette Analysis for Performance Evaluation in Machine Learning with Applications to Clustering," *Entropy*, vol. 23, no. 6, doi: 10.3390/e23060759.

[33] Stewart , G. and Al-Khassaweneh, M., "An Implementation of the HDBSCAN* Clustering Algorithm," *Applied Sciences*, vol. 12, no. 5, doi: 10.3390/app12052405.

[34] Su, S., "Method of Location and Capacity Determination of Intelligent Charging Pile Based on Recurrent Neural Network," *World Electric Vehicle Journal,* vol. 13, no. 10, p. 186, 2022.

[35] Surakhi, O., Zaidan, M. A., Fung, P. L., Hossein Motlagh, N., Serhan, S., AlKhanafseh, M., Ghoniem, R. M., and Hussein, T., "Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm," *Electronics,* vol. 10, no. 20, p. 2518, 2021.

[36] Vakili, M. G., Rezaei, M., "Performance analysis and comparison of machine and deep learning algorithms for IoT data classification.," *arXiv preprint,* vol. arXiv:2001.09636, 2020.

[37] Van Houdt, G., Mosquera, C., and Nápoles, G., "A review on the long short-term memory model," *Artificial Intelligence Review,* vol. 53, no. 8, pp. 5929-5955, 2020.

[38] Velmurugan, T. S., "A survey of partition based clustering algorithms in data mining: An experimental approach.," *Information Technology Journal,* pp. 478-484, 2011.

[39] Vujović, Ž., et al., "Classification model evaluation metrics.," *International Journal of Advanced Computer Science and Applications,* vol. 12.6: 599-606., 2021.

[40] W.Earl Sasser, S. P. A., "Selling Jobs in the Service Sector," *Business Horizons,* vol. 19, no. 3, pp. 61-65, 1976.

[41] Xie, J. G., Ross; Farhadi, A., "Unsupervised deep embedding for clustering analysis," *International conference on machine learning,* pp. p. 478-487, 2016.

[42] Xu, D. and Tian, Y., "A Comprehensive Survey of Clustering Algorithms," *Annals of Data Science,* vol. 2, no. 2, pp. 165-193, 2015/06/01 2015.

[43] Jung, Y., Kim, B., Han, S., "Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar PV

doi:10.6342/NTU202401885

facilities: Case study of South Korea,
Journal of Cleaner Production,," *Journal of Cleaner Production,* vol. 250, p. 119476, 2020.

[44] Yu, Y., Chen, X., and Zhang, F., "Dynamic Capacity Management with General Upgrading," *Operations Research,* vol. 63, no. 6, pp. 1372-1389, 2015.

[45] Yu, Y., Si, X., Hu, C., and Zhang, J., "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation,* vol. 31, no. 7, pp. 1235-1270, 2019.

doi:10.6342/NTU202401885

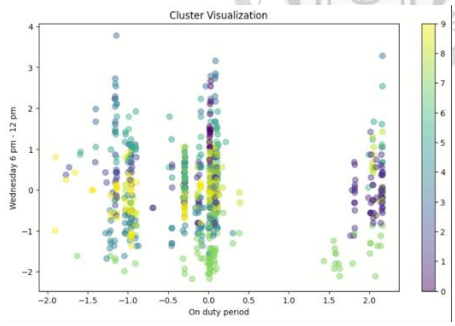
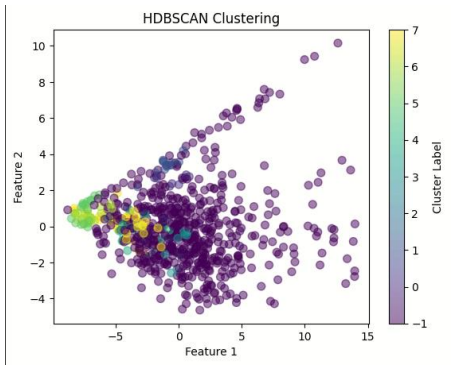
Appendix

Table - A 1. The training result for four dataset and recurrent neural network models

Dataset, Model / Evaluation Metrics	Testing MSE	Prediction MAE	Prediction RMSE
Contiguous Period, RNN	92.96	7.08	14.07
Weekday Based, RNN	109.37	7.61	13.08
Half Year Scope, RNN	251.02	13.04	17.95
Half Year vary Contiguous Period, RNN	85.33	7.05	14.72
Contiguous Period, RNN, without machine identifier	154.03	8.22	17.11
Weekday Based, RNN, without machine identifier	210.15	10.81	19.47
Half Year Scope, RNN, without machine identifier	144.25	8.45	19.01
Half Year vary Contiguous Period, RNN, without machine identifier	136.16	8.41	18.07
Contiguous Period, LSTM, with Embedding layer for machine identifier	155.26	10.21	12.46
Weekday Based, LSTM, with Embedding layer for machine identifier	152.85	10.01	12.35
Half Year Scope, LSTM, with Embedding layer for machine identifier	151.73	9.98	12.34
Half Year vary Contiguous Period, LSTM, with Embedding layer for machine identifier	151.32	9.92	12.31

doi:10.6342/NTU202401885

Table - A 2. Clustering Result without autoencoder employed

	Silhouette Score	Data Point Distribution
K-means without autoencoder, 10 clusters	0.159	
HDBSCAN without autoencoder, 8 clusters	-0.214	

doi:10.6342/NTU202401885