Bus Station Demand Prediction Using Crowdsourced Data and Deep Learning Models

Seyed Hassan Hosseini
Faculty of Science, Technology and Medicine
University of Luxembourg,
Mobilab Transport Research Group, Esch-sur-Alzette, Luxembourg
seyedhassan.hosseini@uni.lu

Nima Darabi
Department of Civil, Constructional and Environmental Engineering
Università di Roma La Sapienza
Roma, Italy
darabi.1961599@studenti.uniroma1.it

Federico Bigi
Faculty of Science, Technology and Medicine
University of Luxembourg,
Mobilab Transport Research Group, Esch-sur-Alzette, Luxembourg
federico.bigi@uni.lu

Francesco Viti
Faculty of Science, Technology and Medicine
University of Luxembourg,
Mobilab Transport Research Group, Esch-sur-Alzette, Luxembourg
francesco.viti@uni.lu

Guido Gentile
Department of Civil, Constructional and Environmental Engineering
Università di Roma La Sapienza
Roma, Italy
guido.gentile@uniroma1.it

***Abstract*—As urban populations increase and traffic congestion escalates, public transportation systems, especially buses, provide a sustainable solution by decreasing the reliance on private cars and minimizing fuel consumption. However, operators must address passengers' concerns about long waiting times and overcrowded conditions to keep buses an attractive option. Therefore, real-time predictions of passenger demand are essential for optimizing scheduling, reducing headways, and enhancing service reliability. Despite its importance, short-term forecasting of bus passenger demand is still underexplored, facing challenges such as seasonal fluctuations, periodicities, and interactions with other transport modes. This paper introduces a new study to predict bus station demand patterns using Google Popular Times (GPT) data through a two-step deep learning approach. Drawing on real-world historical data from bus stations, we propose a predictive framework that starts by classifying passenger demand at each station into distinct clusters. Sequence-to-sequence (Seq2Seq) models are subsequently trained for each cluster to predict demand patterns for the next 24 hours, using the previous 72 hours of data as input.**

Keywords—Bus demand prediction, Google popular times, Deep learning models

I. Introduction

Bus stops are essential for city transport, linking people to wider travel networks. However, these stops often face problems such as servicing buses that do not run on time, insufficient space for waiting passengers, and poorly planned schedules. These issues lead to overcrowded stops and extended wait times. Understanding bus stations' demand patterns becomes crucial when addressing these problems. Moreover, improved demand prediction models enable transport companies to optimize bus schedules, allocate buses more efficiently, and adjust routes as needed. Rather than relying solely on ticketing data, which is no longer available in some countries, such as Luxembourg, where Public Transport (PT) is free and lacks entry or exit checkpoints, new mobile applications, and GPS methods have been developed to provide faster and more accurate predictions. However, these methods can be costly and may raise privacy concerns. In this study, we explore the potential of Google Popular Times (GPT) data, a source of data that captures location busyness based on aggregated user movement patterns, and as an under-explored resource for predicting passenger demand patterns at bus stops. Focusing on 84 bus stops with available historical GPT data in Esch-sur-Alzette (Luxembourg), we observed that each bus stop displays unique weekly GPT patterns, which are regularly updated to reflect current activity trends. By analyzing historical GPT data from past updates, we develop a predictive model to forecast future GPT metrics. Our study evaluates the accuracy and reliability of these GPT-based predictions, demonstrating a cost-effective approach that relies solely on historical GPT data. Unlike other methods such as those based on mobile network data, smart card data, and other sources, our approach avoids the high costs and time-consuming preparation typically associated with data collection and processing.

The primary goal of this paper is to validate GPT data as an effective tool for predicting passenger demand at bus stops, providing a privacy-sensitive and scalable solution for optimizing public transport systems. The outline of this paper is as follows: Part II provides a short literature review on demand prediction in public transport. Part III introduces the case study of Esch-sur-Alzette, Luxembourg, detailing the data used and the methodology applied. Part IV presents the results of our clustering and prediction models. Finally, Part V concludes the paper with a discussion of the findings, their implications, and future work.

II. Related Works

Understanding travel demand patterns is fundamental to effective urban planning and transportation management. In the urban context, developing accurate models to predict bus stop demand has become essential for optimizing public transit efficiency and enhancing the user experience. Traditional methods for predicting bus stop demand have relied on passenger counts and origin-destination surveys to establish baseline ridership trends. Smit M [1] developed a four-step model integrating spatial, demographic, and service data to offer a comprehensive framework for demand prediction. Carpio [2] applied multiple linear regression to forecast stop-level demand in Madrid, incorporating spatial

and urban variables to underscore the significance of contextual influences. However, these labor-intensive methods struggle with adapting to rapid travel pattern changes in urbanizing areas, as [3] noted. Limited real-time capture and zoning biases prompted the shift towards agile frameworks such as machine learning (ML) and big data analytics, aiming for precise spatiotemporal analysis of modern transit systems. Advancements in computational power have shifted the field toward Machine Learning (ML) and Deep Learning (DL) methodologies. Mariñas-Collado et al. [4] developed a hybrid framework for bus passenger demand forecasting that combines clustering, model integration, and cointegration techniques. Their enhanced SVM model significantly outperformed traditional approaches in Salamanca, Spain, while cointegration efficiently extrapolated predictions across clustered stops, creating a scalable solution for bus-centric transit systems Moreover, for bus ridership prediction, a multilayer perceptron (MLP) network was proposed by Farahmand et al. [5]. Weather conditions, alongside events, holidays, and cancellations, were incorporated as influential factors. Their experimental results demonstrated significant improvements in prediction accuracy when incorporating meteorological parameters, especially during extreme weather events. For regional bus passenger flow and revenue prediction, a deep learning framework incorporating LSTM, RNN and greedy layer-wise algorithm was proposed by Nagaraj et al. [6]. Their approach extended prior spatial-temporal models by integrating revenue forecasting and iterative regional clustering, offering actionable insights for resource allocation in bus-centric networks. Collectively, these studies elevated predictive precision but introduced challenges related to data volume requirements, computational intensity, and reduced interpretability for practical applications.

The advent of crowdsourced and system-generated data has transformed demand modeling in public transportation, facilitating detailed, real-time analysis of passenger behavior. Foundational work by Pelletier et al. [7] established the value of smart card data in examining boarding and alighting patterns, although its dependence on fare-collection systems constrained its capacity to capture latent demand or the behaviors of non-transit users. Expanding upon this, Hussain et al. [8] conducted a comprehensive review of approaches for constructing transit origin–destination (OD) matrices from smart card data, underscoring challenges in data validation, transfer identification, and zonal OD estimation, while noting persistent limitations in adapting stop-level OD to broader spatial frameworks. Addressing these challenges, Samaras et al. [9] combined automated vehicle location (AVL) and automated passenger counting (APC) systems to forecast stop-level demand, revealing that integrating sensor data and advanced feature engineering markedly enhances model applicability across diverse routes. In recent advances, artificial intelligence (AI) and edge computing have further refined predictive accuracy. Liyanage et al. [10] employed bidirectional long short-term memory (BiLSTM) networks to predict short-term bus demand with high accuracy using Melbourne's smart card data, demonstrating improved performance over conventional models in capturing temporal relationships. The transportation forecasting landscape has undergone a methodological evolution, transitioning from conventional approaches through machine learning advancements to the present exploration of crowdsourced data streams. Although traditional methods established fundamental principles, they are less robust under dynamic conditions. ML/DL techniques improved predictive power but introduced challenges in computational resources and resulted in interpretability. In addition, crowdsourced approaches encountered limitations related to coverage, privacy, and granularity. Within this progression, GPT data has emerged as a particularly promising metric, offering extensive geographic coverage coupled with hourly temporal resolution. Recent frameworks such as TransitCrowd demonstrated GPT's effectiveness in estimating passenger volumes at subway stations [11]. Subsequent research expanded this foundation by integrating surrounding activity data and establishing quantifiable relationships between points of interest (POIs) and usage patterns [12]. Our study builds upon this nascent but promising research trajectory, specifically examining GPT's potential for predicting hourly demand fluctuations at bus stops. By investigating this application, we aim to determine whether this widely accessible data source can provide actionable insights for bus service optimization while establishing a methodological framework applicable across diverse urban environments.

III. Case Study Area, Dataset and Methodology

GPT offers an approach to analyzing demand patterns across diverse POIs through anonymized smartphone location data. It quantifies the busyness of locations on a 0-100 scale, where zero indicates the area is closed and 100 shows it is open. 1 and 100 represent the relative lowest and highest hourly visit levels within a week. This relative scaling system shows temporal visitation patterns for any POI with sufficient user data while maintaining privacy thresholds, without requiring dedicated counting infrastructure. Several challenges emerge when analyzing this data across different types of locations. Data quality depends heavily on smartphone penetration rates, potentially undermining reliability at less-frequent places. Additionally, the relative normalization methodology also obscures absolute visitor volumes, complicating comparisons between POIs with different baseline activity levels. Moreover, precision in data attribution can be compromised for locations where visitors have brief dwelling times, reducing overall granularity.

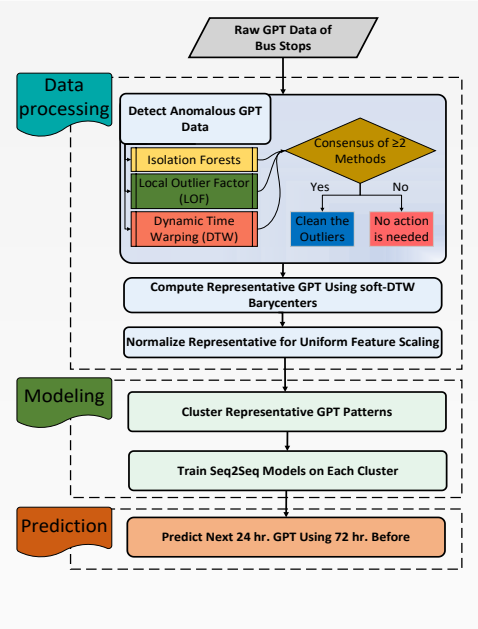

Fig. 1. Two-step predictive framework

Fig 1 illustrates a two-step predictive framework for forecasting bus stop passenger demand. The first step, data processing, involves detecting and cleaning anomalous GPT data using methods such as Isolation Forests, Local Outlier Factor (LOF), and Dynamic Time Warping (DTW), with outliers removed based on a consensus of at least two methods. Once cleaned, representative GPT patterns are computed using Soft Dynamic Time Warping barycenters and normalized for feature scaling. The second step, modeling and prediction, clusters these representative patterns and trains Seq2Seq models for each cluster. Finally, the trained models are used to predict the next 24 hours of GPT using data from the preceding 72 hours.

A. Data Acquisition and Preprocessing

In this research, we utilized historical GPT data from 84 bus stops, collected over two months, with six updates per stop during this period. We investigated the potential of employing past GPT patterns to predict future busyness patterns at these stops. To identify anomalous weeks in the time series data collected from various locations, a comprehensive outlier detection framework has been implemented using DTW [13], Isolation forests [14], and LOF [15]. A multi-method outlier detection framework was implemented to mitigate the limitations of individual techniques and address the heterogeneous nature of temporal anomalies. This framework was designed to address the inherent complexities of temporal data, such as non-linear patterns, varying densities, and potential misalignments while minimizing the limitations of any single method. Isolation Forest was applied to detect global outliers through recursive random partitioning of the feature space. Indeed, the computational efficiency and effectiveness of this method in identifying global extremes make it an ideal choice for our dataset. To complement the global perspective of Isolation Forest, the LOF method was applied to identify anomalies based on local density deviations in the flattened feature space. LOF quantifies the degree to which a point's local density differs from that of its neighbors, assigning negative outlier factor scores to signify anomaly severity. This approach excels at detecting subtle irregularities that may not stand out globally but are anomalous within their local context in the feature space. Moreover, LOF does not directly consider temporal structure, it provides a valuable perspective on local anomalies. The DTW-based method identifies structurally dissimilar time series by computing the mean DTW distance of each sequence to all others. In other terms, it is a measure reflecting global deviation from typical temporal patterns. Outliers exhibit elevated mean distances due to anomalous shapes or phase shifts undetectable by non-temporal methods. To robustly flag these deviations, we applied a Median Absolute Deviation (MAD) criterion, which compares distances to their median while resisting distortion from extreme values. Therefore, this method complements the previous mentioned approaches by targeting temporally misaligned or structurally distinct patterns. Recognizing that each method offers a distinct perspective on anomaly detection, a consensus-based strategy was adopted to synthesize their outputs. A time series was classified as an outlier only if at least two of the three methods flagged it as an outlier. This approach mitigates the risk of false positives inherent in individual techniques while leveraging their collective strengths. Additionally, combined anomaly scores were calculated by normalizing and integrating outputs from all three methods. In particular, scores from these methods were converted to uniform ranks in order to mitigate scale disparities. A composite score was derived as the arithmetic mean of these ranks to ensure equal weighting across methodologies. In addition, to resolve cases with no consensus outliers, the 95th percentile of composite scores was used as an adaptive threshold.

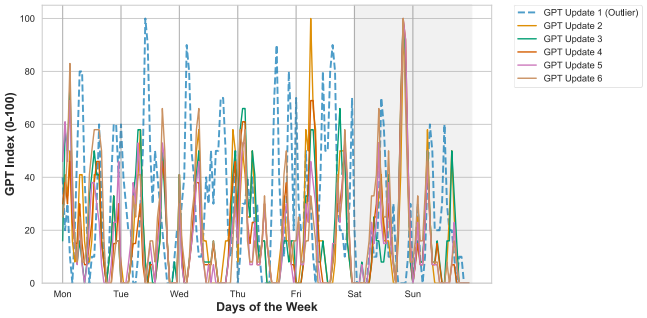

Fig. 2. Outlier detection results for a single bus stop

Fig 2 shows a representative example using GPT data for a bus stop with six historical updates. Following outlier detection, update 1 (dashed line), the oldest GPT entry in our dataset, was identified as an anomaly. Notably, this pattern aligns with broader observations. In fact, the oldest GPT entries often display notable deviations from later profiles. To ensure robustness, these early outliers were systematically excluded from analysis.

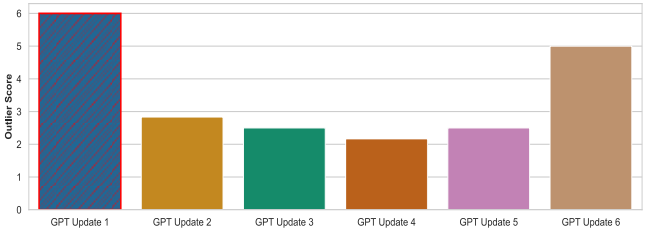

Fig. 3. Anomaly score across successive updates of GPT data for a single bus station

Fig 3 presents the outlier score for six updated weeks of time series data at each bus location, where higher scores indicate greater anomaly. GPT update 1 stands out significantly with an outlier score of approximately 6, suggesting a clear outlier. In contrast, other updates have much lower scores, ranging from around 2 to 3, indicating that their activity patterns are relatively typical and consistent with each other.

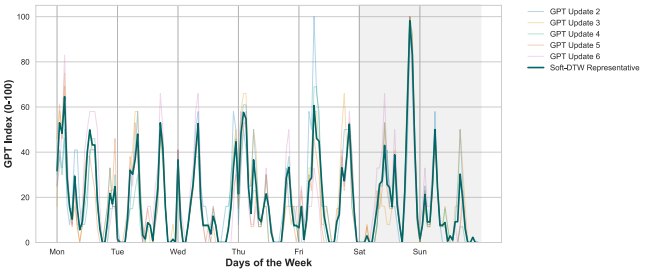

Fig. 4. Representative occupancy pattern for a bus stop using soft DTW barycenter

To cluster the GPT pattern of bus stations effectively, it is essential to derive a representative pattern after removing outliers. Soft Dynamic Time Warping barycenter [16] has been applied to arrive at one representative for each bus station. Fig 4 shows the representative that captures the pattern of barycenter across the input time series, accounting for temporal misalignments, which is particularly useful for summarizing typical activity patterns in time series data.

B. Clustering Approach

To categorize bus stations into distinct groups based on their GPT patterns, we utilized Time Series KMeans algorithm. This method was chosen for its effectiveness in capturing the temporal dependencies present in time-series datasets. Additionally, DTW has been used as the distance meature to account for temporal shifts and variations. Before clustering, the data were normalized using a standard scaler to ensure consistent feature scaling across all the time series. This preprocessing step transforms each time series to have a mean of zero and a variance of one, allowing to emphasize the shape of the temporal patterns rather than their absolute magnitudes. To determine the optimal number of clusters, four widely-used validation metrics: Silhouette Score, which quantifies how well-separated and cohesive clusters are (higher values indicate better clustering); Davies-Bouldin Index, which measures the average similarity between clusters (lower is better); Calinski-Harabasz Index, which evaluates the ratio of between-cluster to within-cluster dispersion (higher is better); Inertia, the sum of squared distances to cluster centroids, often used in the elbow method to identify a point of diminishing returns have been used. Furthermore, to enhance the accuracy in determining the optimal number of clusters, the average of RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and R^2 (Coefficient of Determination) were computed after training different deep learning models for each cluster number. This approach is supposed to help in selecting the best number of clusters. Out of 84 bus stations, the maximum number of clusters was set to half of the total bus stations.

C. Recurrent Neural Network Models

Recurrent Neural Networks (RNNs) are a specialized class of neural networks designed to process sequential data. These networks excel in natural language processing and time-series predictions, largely due to their unique feature of hidden states that preserve information from previous steps. Unlike traditional feedforward neural networks, RNNs can process sequences of varying lengths by maintaining an internal state that captures context across time steps. However, standard RNNs face significant challenges, particularly the vanishing gradient problem, which hinders their ability to effectively learn and capture long-term dependencies [17]. Among various RNN models, LSTM has gained significant recognition over traditional RNNs due to its ability to efficiently capture both long-term and short-term dependencies in time series data. LSTMs are a specialized type of RNN, integrated within the Sequence-to-Sequence (Seq2Seq) architecture, designed to effectively handle long- and short-term dependencies in sequential data [17].

D. Sequence-to-Sequence (Seq2Seq) Architecture

Seq2Seq models have significantly advanced the fields of natural language processing and machine learning by offering a robust framework for mapping input sequences to corresponding output sequences. Leveraging recent developments in deep learning and memory-augmented architectures, these models demonstrate exceptional performance in understanding and generating sequential data. Their ability to preserve and transmit contextual information across sequences makes Seq2Seq models particularly well-suited for a wide range of sequence-based tasks, including machine translation, video captioning, and code generation [18], [19]. To forecast the next day's bus station demand using a transformer model, we employ a sliding window approach with sequences of consecutive past demand values as input. Each sequence comprises historical demand data, such as [$D1$, $D72$], up to [Dk-1, Dk] with corresponding target values being the subsequent demand observations, denoted as $D73$, …, Dk+1. The model is trained on historical data spanning the previous 72 hours of GPT trends, enabling it to capture temporal patterns and dependencies within the time series for accurate prediction of future bus station demand.

E. Model Architecture and Training Process

The architecture comprises two main components: an encoder and a decoder. The encoder consists of two stacked LSTM layers with 128 and 64 hidden units. Both layers are regularized using L2 weight decay (λ=0.001), followed by batch normalization and a dropout layer with a dropout rate of 0.3 to mitigate overfitting. The final encoder output is passed through a repeat vector to align with the target output length. The decoder has two LSTM layers (64 and 32 units), also followed by batch normalization and dropout. The decoder outputs a sequence of vectors fed into a time-distributed dense layer with a linear activation function to produce the next 24-hour bus stop demand prediction. The model was trained using the Adam optimizer with a learning rate of 0.001. Each model was trained for 200 epochs with a batch size of 64, and early stopping was considered based on validation performance. The dataset for each cluster was split with 80% of the samples used for training and the remaining 20% for testing. To evaluate model performance, RMSE, MAE, and R^2 scores are calculated on the test set. For each prediction, we calculated 95% confidence intervals based on the standard error of the residuals using the t-distribution.

IV. RESULTS

This section presents two key results. First, the outcomes of the clustering process applied to bus stations are discussed. After determining the optimal number of clusters, a separate Seq2Seq model is trained for each cluster. The models' performance trained on each cluster is then evaluated and reported.

A. Bus Stations Clustering

To determine the optimal number of clusters for segmenting bus stations based on their GPT data, we evaluated several clustering configurations using different numbers of clusters. The quality of each clustering solution was assessed using standard internal validation metrics. These metrics provide insight into the compactness and separation of clusters, helping to identify the most appropriate clustering structure. TABLE I shows increasing the number of clusters generally led to improvements in some clustering validity metrics, such as a decrease in the Davies-Bouldin Index and inertia. The Davies-Bouldin Index decreased from 3.58 (with 10 clusters)

to 1.98 (with 30 clusters), indicating improved cluster separation. Following the clustering step, a dedicated Seq2Seq model was trained for each cluster configuration, and the average model performance was evaluated using RMSE, MAE, and R^2. Increasing the number of clusters to 30 results in reduced prediction accuracy, with RMSE rising from 11.35 to 16.14 and R^2 decreasing from 0.71 to 0.58. This highlights the trade-off between clustering granularity and model performance. Based on an evaluation of both clustering quality and predictive accuracy, the configuration with 10 clusters was selected as the optimal solution. An increase in the number of clusters resulted in marginal improvements in clustering validity metrics such as the Davies-Bouldin Index and inertia; however, a decline in prediction performance was also observed. In contrast, the 10-cluster configuration yielded an RMSE of 11.35, the highest R^2 score of 0.71, and a competitive MAE of 8.60.

TABLE I Clustering configurations and Seq2Seq performance

Cluster numbers	Silhouette	Davies-Bouldin	Calinski-Harabasz	Inertia	Avg RMSE	Avg MAE	Avg R^2
10	**-0.020**	**3.5896**	**1.60**	**26.199**	**11.35**	**8.60**	**0.71**
15	**-0.053**	**3.084**	**1.38**	**23.38**	**11.36**	**8.58**	**0.70**
20	**-0.063**	**2.624**	**1.348**	**20.84**	**11.46**	**8.48**	**0.67**
25	**-0.065**	**2.19**	**1.328**	**18.34**	**12.58**	**9.61**	**0.63**
30	**-0.068**	**1.981**	**1.308**	**18.34**	**16.14**	**10.11**	**0.58**

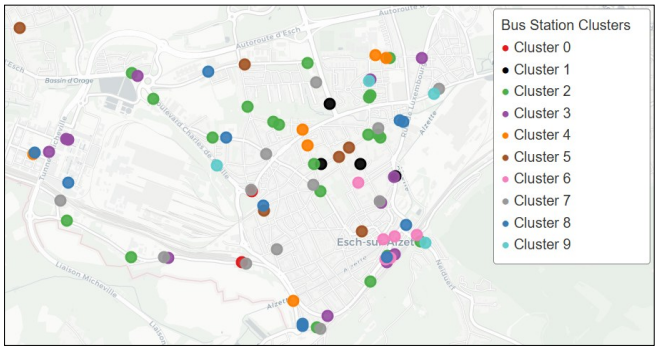

Fig. 5. Bus station clustering results

Fig 5 illustrates the spatial distribution of bus stations in Esch-sur-Alzette, Luxembourg, categorized into 10 distinct clusters based on the similarity of their historical demand patterns. Each colorful circle represents a bus station, and its color corresponds to its assigned cluster, as indicated in the legend. The clustering reveals that stations with similar temporal usage profiles are often located near each other, although some clusters span different parts of the city. This spatial segmentation provides a meaningful basis for developing cluster-specific forecasting models, enabling more accurate and localized predictions of future bus station demand.

B. Deep Learning Results

The clustering of 84 bus stations enables the development of individual Seq2Seq models for each cluster to predict future demand. Each model is trained using historical GPT-derived data specific to its assigned cluster, allowing for targeted and accurate forecasting.

TABLE II. Results of prediction models for each cluster

Cluster Number	Number of bus stations in each cluster	RMSE	MAE	R^2
0	2	14.108	10.66	0.563
1	4	13.680	10.392	0.665
2	21	11.887	9.104	0.720
3	12	12.134	9.22	0.658
4	6	12.291	9.167	0.709
5	6	6.0137	4.561	0.915
6	6	12.232	9.263	0.68
7	12	10.755	8.120	0.765
8	11	11.709	8.979	0.729
9	4	8.694	6.592	0.76

Table II presents the performance of individual Seq2Seq models trained for each 10 clusters. Each row corresponds to a specific cluster. The results demonstrate variability in model performance across clusters, which is partly influenced by the number of bus stations and the characteristics of their historical demand patterns. Cluster 5, which includes 6 stations, achieved the best performance with an RMSE of 6.01, MAE of 4.56, and an R^2 of 0.915, indicating a highly accurate model. In contrast, Cluster 0 and Cluster 1, with only 2 and 4 stations respectively, exhibited higher error rates and lower R^2 scores, suggesting limited predictive capability likely due to smaller training sets or more volatile demand patterns. Overall, the findings highlight the influence of cluster composition and size on model accuracy and emphasize the benefit of data-driven segmentation in enhancing short-term demand prediction.

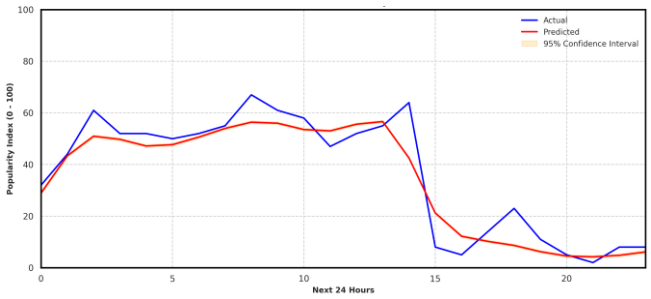

Fig. 6. Prediction of demand for a bus stop in cluster 5 (next 24 hrs)

Fig 6 presents the actual and predicted demand for the next 24 hours with a 95% confidence interval (CI) at a bus station in Cluster 5. The model was trained on GPT data from all bus stations within Cluster 5 and used to forecast future demand for this specific station. The results show that the transformer model effectively captures the overall trend of bus demand. The CI provides a range of expected demand, helping to decide increase or decrease the frequency of the bus line or optimize the lines. In this study, we demonstrate the potential of using GPT-derived data to predict short-term demand at bus stations. GPT data is widely available and easily accessible, making it a practical source for demand forecasting. Our results show that, with appropriate clustering and model training, future demand at bus stations can be predicted with promising accuracy. Future work will focus on enhancing model performance, extending the analysis to

include a larger set of bus stations, and integrating additional GPT signals related to surrounding activities.

V. CONCLUSIONS

This study introduced a novel framework for predicting bus station demand using crowdsourced data, specifically Google Popular Times (GPT), in combination with deep learning models. By leveraging the temporal patterns inherent in GPT data, we addressed key challenges in urban public transport planning, including the lack of ticketing-based data in fare-free systems such as Luxembourg. Our approach involved a two-step methodology: first clustering bus stations based on their weekly GPT patterns using time-series clustering techniques, and then training customized Seq2Seq models for each cluster to forecast demand over a 24-hour horizon using the previous 72 hours of data. The results demonstrated that deep learning models were able to effectively capture temporal dependencies and provided accurate predictions for each station group. Evaluation metrics such as RMSE, MAE, and R^2 confirmed the robustness of the proposed models across clusters. Our findings validate the potential of using crowdsourced data as a cost-effective, scalable, and privacy-preserving alternative for public transport demand forecasting. This approach not only eliminates the need for expensive infrastructure and data collection systems but also enables more dynamic and responsive transport planning, especially in urban environments undergoing rapid changes in mobility patterns.

VI. REFERENCES

[1] M. Smit, "Exploring the demand for bus transport," 2014.

[2] J. Carpio-Pinedo, "Urban bus demand forecast at stop level: Space Syntax and other built environment factors. Evidence from Madrid," *Procedia-Social and Behavioral Sciences*, vol. 160, pp. 205–214, 2014.

[3] X. Chu, "Ridership models at the stop level," University of South Florida. Center for Urban Transportation Research, 2004.

[4] I. Mariñas-Collado, A. E. Sipols, M. T. Santos-Martín, and E. Frutos-Bernal, "Clustering and forecasting urban bus passenger demand with a combination of time series models," *Mathematics*, vol. 10, no. 15, p. 2670, 2022.

[5] Z. H. Farahmand, K. Gkiotsalitis, and K. T. Geurs, "Predicting bus ridership based on the weather conditions using deep learning algorithms," *Transp Res Interdiscip Perspect*, vol. 19, p. 100833, 2023.

[6] N. Nagaraj, H. L. Gururaj, B. H. Swathi, and Y.-C. Hu, "Passenger flow prediction in bus transportation system using deep learning," *Multimed Tools Appl*, vol. 81, no. 9, pp. 12519–12542, 2022.

[7] M.-P. Pelletier, M. Trépanier, and C. Morency, "Smart card data use in public transit: A literature review," *Transp Res Part C Emerg Technol*, vol. 19, no. 4, pp. 557–568, 2011.

[8] E. Hussain, A. Bhaskar, and E. Chung, "A novel origin destination based transit supply index: Exploiting the opportunities with big transit data," *J Transp Geogr*, vol. 93, p. 103040, 2021.

[9] P. Samaras, A. Fachantidis, G. Tsoumakas, and I. Vlahavas, "A prediction model of passenger demand using AVL and APC data from a bus fleet," in *Proceedings of the 19th panhellenic conference on informatics*, 2015, pp. 129–134.

[10] S. Liyanage, R. Abduljabbar, H. Dia, and P.-W. Tsai, "AI-based neural network models for bus passenger demand forecasting using smart card data," *Journal of Urban Management*, vol. 11, no. 3, pp. 365–380, 2022.

[11] P. Vitello, C. Fiandrino, R. D. Connors, and F. Viti, "Transitcrowd: Estimating subway stations demand with mobile crowdsensing data," *Data Science for Transportation*, vol. 6, no. 2, p. 6, 2024.

[12] P. Vitello, R. D. Connors, and F. Viti, "Leveraging Crowdsourced Activity Information for Transit Stations Flow Estimation," *IEEE Access*, 2024.

[13] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[14] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, IEEE, 2008, pp. 413–422.

[15] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[16] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *International conference on machine learning*, PMLR, 2017, pp. 894–903.

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

[18] A. Nugaliyadde, "Extending Memory for Language Modelling," *arXiv preprint arXiv:2305.11462*, 2023.

[19] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4534–4542.