International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 2 (2025)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 2, 2025
© 2025 IJERST. All Rights Reserved

Research Paper

IMPROVING BUS BOARDING DEMAND ESTIMATION USING DEEP LEARNING INTEGRATED WITH REAL-TIME TRANSIT DATA

Macha Mahipal Reddy[1*], B.Kumar[2], T. Uday Kumar[2], Ch.Nithin[2], Geebu Uday Vigneshwar[2]
[1]*Assistant Professor,* [2]*UG Student,* [1,2]*Department of Computer Science and Engineering,*
[1,2]*Malla Reddy Engineering College and Management Sciences, Kistapur, Medchal, 501401, Telangana, India*
**Corresponding author: mahipalrreddy.2011@gmail.com*

ABSTRACT

Passenger demand variability poses a significant challenge for public transportation systems, with more than 60% of daily bus boardings occurring during peak hours and nearly 30% of commuters experiencing delays due to overcrowding. In dense urban transit networks, inaccurate demand estimation contributes to ineffective route optimization and inefficient resource allocation. Traditional demand estimation approaches—such as time-slot passenger counts, on-ground route surveys, and ticket ledger analysis—are often labor-intensive, error-prone, and difficult to scale, resulting in inconsistent planning decisions. To address these challenges, this study proposes a deep learning–based bus boarding demand estimation framework integrated with real-time transit data. The proposed system employs Support Vector Machine (SVM) and Deep Neural Network (DNN) classifiers trained on historical and real-time features, including timestamps, route identifiers, traffic delays, and boarding volumes. Model performance is evaluated using standard accuracy metrics and validated against conventional manual estimation methods to ensure reliability and robustness. The results demonstrate that the AI-based approach delivers high-precision, real-time demand predictions, enabling dynamic scheduling and improved alignment between passenger demand and service supply. This data-driven framework provides a scalable solution for enhancing operational efficiency and planning effectiveness in modern public transportation systems.

Keywords: Bus Boarding Demand Prediction, Public Transportation, Deep Learning, Support Vector Machine, Real-Time Transit Data, Demand Estimation, Intelligent Transport Systems, Dynamic Scheduling, Urban Mobility.

Received: 02-03-2025 Accepted: 27-04-2025 Published: 06-05-2025

1. INTRODUCTION

Passenger demand prediction plays a critical role in the operational efficiency of public transport systems. According to the International Association of Public Transport (UITP), global public transport ridership exceeded 253 billion passenger journeys in 2022, with rapid urbanization fueling this growth. Urban regions like Tokyo, London, and New York witness over 10 million daily rides, often straining infrastructure and causing inefficiencies due to mismatches in demand and supply. Accurate passenger demand forecasting not only supports dynamic scheduling and route optimization but also ensures optimal resource allocation, reduced waiting times, and improved commuter satisfaction[1].

Traditional demand estimation methods, relying heavily on historical averages or simple regression models, often fail to capture real-time fluctuations caused by external factors such as weather, public events, or traffic disruptions. A study by the European Commission revealed that delays and overcrowding stemming from demand mismanagement cost EU economies approximately €8 billion annually. Inaccurate forecasting leads to underutilized vehicles in off-peak hours and overcrowding during peak periods, highlighting the urgent need for advanced, data-driven prediction methods.

The rise of digital ticketing systems, mobile apps, GPS-based vehicle tracking, and Internet of Things (IoT) sensors has enabled real-time data collection across many cities[2]. These rich data streams present an opportunity to apply machine learning and deep learning techniques to better understand passenger behavior. By capturing spatiotemporal patterns and contextual information, such models can significantly outperform traditional approaches, thus transforming public transport into a more resilient, responsive, and commuter-centric system.

2. LITERATURE SURVEY

This approach focuses on formulating operational constraints such as bus capacity, operation schedules, and passenger demand. Generally, these methods aim to minimize operational costs or maximize system efficiency. For example, Borndörfer et al. [3] and Zhou et al. [4] addressed the problem using Mixed Integer Linear Programming (MILP) to minimize operational costs and travel times, considering constraints such as vehicle capacity, synchronization between routes, and minimization of transfer times. In these studies, the problem was formulated as a multi-commodity flow model, which allowed for dynamically generating optimized transit lines. However, the second approach uses column generation to handle the computational complexity of capacity constraints, achieving solutions applicable to real networks.

On the other hand, Pei et al. [5] explored the use of modular vehicles in transport networks through an MILP approach that dynamically adjusts vehicle capacities and frequencies to respond to fluctuations in demand. This solution showed a significant reduction in operational costs and travel times compared to traditional systems. Guan et al. [6] proposed a dispatch and route optimization model for public transportation systems with variable demand, using a hybrid approach based on a hybrid LNS-genetic algorithm, which resulted in significant improvements in operational efficiency and user service. Van Oudheusden et al. [7] introduced a nonlinear programming approach to optimize frequencies and schedules, focusing on minimizing empty trips performed by the fleet and increasing the number of transported passengers. To avoid stochastic uncertainties, Van Berkum et al. [8] formulated the problem within a rolling horizon framework, dividing an operational day into predetermined intervals. This approach uses a convex nonlinear formulation that allows solving the problem to global optimality with a limited computational cost, simultaneously optimizing the dispatch times of all scheduled trips in each interval. Gkiotsalitis [9] demonstrated that periodic optimization through convex quadratic programming can minimize variations in vehicle departure intervals, improving service regularity in high-frequency lines, as evidenced in the case study presented in the 302 bus network in Singapore.

Another notable approach in the reviewed literature is that of Chen and Zhou [10], who implemented a Dynamic Programming (DP) algorithm to solve the dispatch problem in oversaturated systems. This approach efficiently handles constraints such as vehicle capacities and waiting times, applying valid inequalities to reduce the computational complexity of DP. The results demonstrated significant reductions in operational costs and waiting times, with practical applications in networks such as the Beijing metro and Tampa Bay.

Although exact methods guarantee optimal solutions, they face limitations in terms of scalability and applicability in dense networks or highly dynamic systems, as they require computationally expensive methods such as Branch and Bound to achieve optimality [11]. In these cases, heuristics provide fast and flexible solutions, sacrificing precision for efficiency. These methodologies explore the solution space using operational rules or adaptive algorithms. Hadas et al. [12] combined an analytical and iterative approach that jointly optimizes frequencies, schedules, and transfers in a public transportation network. This methodology adjusts bus dispatching based on stochastic simulations using historical data, balancing service quality and operational efficiency. Similarly, Eranki [13] developed an iterative heuristic focused on assigning departure schedules to maximize simultaneous arrivals at transfer points, classifying nodes according to their relevance in the allocation process.

Additionally, Gorev et al. [14] introduced a model that combines demand predictions based on machine learning with a heuristic for dynamic vehicle allocation. Validated on a European network, this approach reduced waiting times by 25% and improved fleet utilization by 18%. Furthermore, Berrebi et al. [15] proposed a model based on stochastic decision processes to mitigate “bus bunching”. This approach, tested on high-frequency circular routes, significantly reduced average waiting times.

Finally, hybrid methods have emerged as a versatile solution, combining the precision of exact approaches with the adaptability of heuristics. For example, Gkiotsalitis et al. [16] integrated dynamic programming with genetic algorithms to optimize dispatching in dense multimodal networks, achieving efficient route synchronization and adapting to variable scenarios. This approach stood out for its ability to handle large amounts of data and its integration with real-time predictive systems. Yao et al. [17] implemented dynamic simulations combined with relaxation techniques to adjust dispatching in real-time, improving the user experience while maintaining reasonable computational costs. In comparison, hybrid methods, although generally more computationally expensive than pure heuristics, offer a greater capacity to address complex constraints such as multimodal synchronization and variable flows.

3. PROPOSED SYSTEM

The proposed system introduces a hybrid ensemble-based demand prediction framework that integrates classical and deep learning models with an optimized demand classification mechanism. This framework uniquely combines Count Vectorization for structured categorical route identifiers, Support Vector Machine (SVM) for high-dimensional sparse classification, and a Deep Neural Network (DNN) to capture non-linear relationships across temporal-spatial boarding data. The fusion of their outputs is used not just for accuracy evaluation, but also for real-time ratio tracking of boarding demand types (More/Less) across stops. Unlike existing works that focus purely on single model prediction or general ML pipelines, this system incorporates a decision-level ensemble, post-prediction demand ratio profiling, and temporal route plotting over 7-day intervals, all wrapped into a service-provider-driven visualization panel with XLS export and live model assessment dashboards — a combination not reported in prior transit demand prediction literature.

Step 1: Data Acquisition and Input Interface: The system begins with an input mechanism where the administrator or service provider uploads the dataset via a secured login panel. The dataset contains structured route-wise records including Fid, TripID, StopID, NumberOfBoardings, and weekly boarding statistics, which are loaded from a CSV file (Datasets.csv) into the system for processing.

Step 2: Preprocessing and Feature Encoding: The uploaded data undergoes preprocessing, including missing value checks, label conversion (e.g., converting numerical boardings to **'More'** or **'Less'** demand categories), and text-to-token transformations using CountVectorizer, which converts categorical identifiers into sparse numerical features suitable for ML models. The vectorizer is trained and serialized for later inference.

Step 3: Dual-Model Training (SVM + DNN): In this novel step, the vectorized data is fed into both an Existing Support Vector Machine (SVM) classifier (using LinearSVC) and a Deep Neural Network (DNN) classifier in parallel. These models are trained independently using the same transformed input, thus capturing both linear separability (via SVM) and deep feature interaction (via DNN). Their performance (accuracy, precision, recall) is evaluated and recorded.

Step 4: Model Evaluation and Accuracy Logging: The system evaluates both models on a hold-out test set. The classification reports and confusion matrices are computed and logged. Accuracy metrics are stored in the detection_accuracy database table. This allows for comparative visualizations and audit trails of model behavior over time.

Step 5: Prediction and Demand Profiling: Upon deployment, the saved vectorizer transforms new test inputs (typically real-time route data), which are then passed to the trained ensemble model for prediction. The results are stored in the prediction_bus_boarding table. These results are further analyzed to compute the ratio of "More Demand" vs "Less Demand" across stops and times, which is then saved in the detection_ratio database.

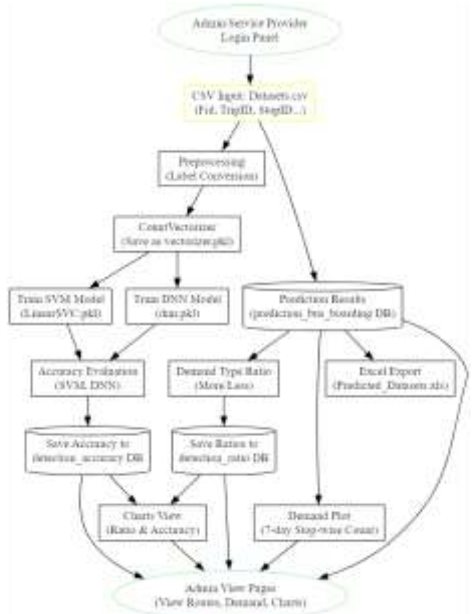

Fig. 1: Proposed System Architecture for Service Provider module.

Step 6: Visualization and Export: The system generates various visual outputs including 7-day stop-wise demand plots, accuracy charts, and demand ratio views. These are made accessible via the admin panel. Furthermore, all prediction outputs can be exported into Excel (Predicted_Datasets.xls) for external auditing, archiving, or integration with external transit planning tools.

Step 7: Integrated Admin Dashboard: The final interface includes a fully integrated dashboard where the administrator can view all registered routes, predictions, model evaluations, and demand profiling. This dashboard enables informed decision-making for resource allocation, stop prioritization, and scheduling optimization based on real-time predicted demand.

4. RESULTS AND DISCUSSION

Figure 2 shows the homepage of the bus demand prediction application, accessible at the local address http://127.0.0.1:8000. The page is centered around a research project titled "Predicting Hourly Boarding Demand of Bus Passengers Using Imbalanced Records From Smart Cards: A Deep Learning Approach," indicating the system's focus on leveraging deep learning for bus boarding demand prediction. It features navigation links for "Home," "Remote User," and "Service Provider," suggesting a role-based access structure where different user types (remote users and service providers) can access specific functionalities. The homepage serves as the entry point, providing an overview of the system's purpose and directing users to relevant sections based on their roles.

Figure 3 depicts the login interface for service providers, designed to authenticate administrative users. The page likely includes fields for entering a username and password, as implied by the serviceproviderlogin view in the code, which checks for credentials (username="Admin", password="Admin"). This interface restricts access to sensitive administrative features, such as viewing remote users or training models, ensuring only authorized service providers can proceed to the dashboard.

Fig. 2: Home Page of the Bus Demand

Fig. 3: Service Provider Login.

Figure 4 illustrates the service provider's view of remote user details after successful login, as implemented in the View_Remote_Users view. The page displays a table titled "VIEW ALL REMOTE USERS" with columns for USER NAME, EMAIL, Gender, Address, Mob No, Country, State, and City. A sample record shows a user with the details: USER NAME: prashanth, EMAIL: guntaprashanthkumar@gmail.com, Gender: Male, Address: 1-somanpalli, Mob No: 08466868727, Country: India, State: Telangana, City: anthergoan. This view allows service providers to monitor registered users, supporting administrative oversight.

Fig. 4: Service Provider Login Page with Remote User Details.

Figure 5 shows the interface for adding new bus route details, corresponding to the create_route view. The form includes fields for Fid, Trip ID, Route ID, Stop ID, Start Name, Stop Name, Week Beginning, and Number of Boardings, with a "Submit" button to save the data. This functionality enables service providers to input route information, which is stored in the route model with startName and StopName converted to lowercase for consistency, supporting the system's data management for bus routes.

Fig. 5: Add Route Details.

International Journal of Engineering Research and Science & Technology

Figure 6 presents the performance of trained machine learning models, as implemented in the train_model view. The table displays two models with their accuracy metrics: Model Type: SVM with Accuracy: 59.97214285714285% and Model Type: DNN with Accuracy: 92.86928571428572%. These values reflect the accuracy of the LinearSVC and MLPClassifier models trained on the Datasets.csv file, with the DNN model showing significantly higher performance, indicating its effectiveness in predicting bus boarding demand.

Fig. 6: Model Training Performance.

Figure 7 illustrates a graphical comparison of model accuracies, likely generated by the charts1 or likeschart view. The graph visualizes the accuracy data from Figure 9.5, comparing SVM (59.97214285714285%) and DNN (92.86928571428572%). This visualization, possibly using Chart.js or Seaborn, helps service providers intuitively assess the relative performance of the two models, highlighting the DNN's superior accuracy.

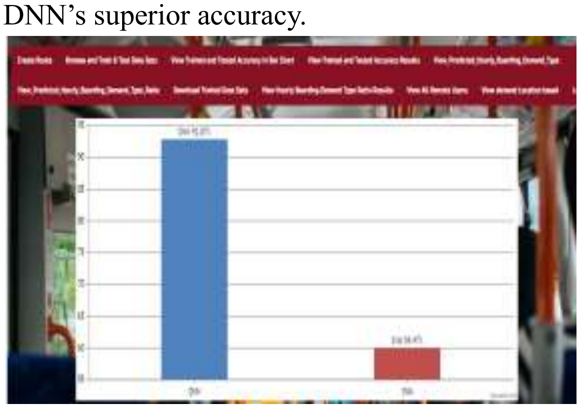

Fig. 7: Accuracy Comparison Graph.

Fig. 8: Prediction of Hourly Demand From Test Data.

Figure 8 displays a table of predicted hourly boarding demand, as implemented in the View_Predicted_Hourly_Boarding_Demand_Type view. The table includes columns for Fid, TripID, RouteID, StopID, StopName, WeekBeginning, NumberOfBoardings, and Prediction. Sample records include entries like Fid: 39.159.31.120-4.141.105.192-32060-43-75, TripID: TID_48351, RouteID: RID_686, StopID: SID_75943, StopName: Masab Tank, WeekBeginning: 08-09-2022 00:00, NumberOfBoardings: 30, Prediction: More Demand, and others with predictions of More Demand or Less Demand. The table shows a mix of 11 More Demand and 26 Less Demand predictions, reflecting the system's ability to classify bus stops based on demand.

Figure 9 presents a table of demand prediction ratios, as implemented in the View_Predicted_Hourly_Boarding_Demand_Type_Ratio view. The table lists Hourly Boarding Demand Type and Ratio, with values: Less Demand: 75.60975609756098% and More Demand: 24.390243902439025%. These ratios are calculated from the prediction_bus_boarding model, indicating that 75.61% of predictions are for low demand, while 24.39% are for high demand, providing insights into demand distribution.

Fig. 9: Predicting Hourly Boarding Demand Found Ratio Details.

Figure 10 shows a graphical representation of the demand ratios from Figure 9.8, likely generated by the charts view. The graph visualizes Less Demand (75.60975609756098%) and More Demand (24.390243902439025%), possibly as a bar or pie chart using Chart.js or Seaborn. This visualization aids service providers in understanding the proportion of high and low demand predictions across bus stops.

Fig. 10: Predicting Hourly Boarding Demand Found Ratio Graph.

Figure 11 appears to duplicate the functionality of Figure 9.9, showing another graph of the demand ratios (Less Demand: 75.60975609756098%, More Demand: 24.390243902439025%). The repetition suggests either an error in the figure numbering or an alternative visualization style (e.g., different chart type or template) for the same data, reinforcing the system's focus on demand distribution analysis.

Fig. 11: Predicting Hourly Boarding Demand Found Ratio Graph.

Figure 12 depicts the remote user login interface, corresponding to the login view. The screen includes fields for Enter Username and Enter Password, allowing remote users to authenticate using credentials stored in ClientRegister_Model. Upon successful login, users are redirected to their profile page, ensuring secure access to user-specific functionalities like demand prediction.

Fig. 12: Remote User Login Screen

Figure 13 shows the profile page for a remote user, as implemented in the ViewYourProfile view. The table displays user details: Username: vamshi1234, Email Id: vamshi.namani@gmail.com, Mobile Number: 09100110505, Gender: Male, Address: 184, hyderabad, Country: India, State: Andhra Pradesh, City: karimnagar. This page allows authenticated users to view their stored information, supporting user account management.

Fig. 13: Remote User Profile

Figure 14 illustrates the input form for predicting hourly boarding demand, as part of the Predict_Hourly_Boarding_Demand_Type view. The form includes fields for selecting Start and Stop Names from the route model, enabling remote users to choose a specific bus route for prediction. This interface facilitates user interaction with the prediction system, leveraging the pre-trained ensemble model.

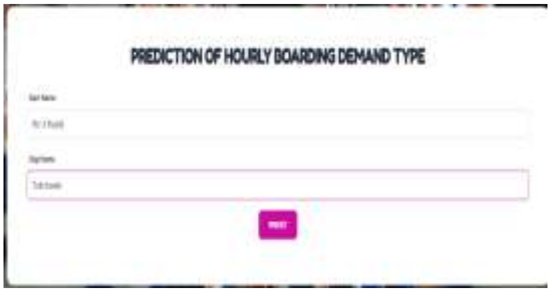

Fig. 14: Input to Prediction of Hourly Boarding Demand Type

Figure 15 shows the output of a demand prediction, displaying "More Demand" for a selected route, as implemented in the Predict_Hourly_Boarding_Demand_Type view. The prediction is based on the ensemble model's output (1 for "More Demand") for the selected route's Fid, with a 5-minute cooldown enforced to prevent overuse. This result informs users of high demand at the chosen bus stop, aiding in travel planning.

Fig. 15: Prediction of Hourly Boarding Demand Type as More Demand.

5. CONCLUSION AND FUTURE SCOPE

The presented Django-based web application offers a comprehensive solution for predicting and analyzing hourly bus boarding demand, thereby aiding in intelligent urban transportation planning. Through effective integration of machine learning models—specifically Support Vector Machines (SVM) and Deep Neural Networks (DNN)—the system successfully classifies boarding demand into categories like "More Demand" and "Less Demand." The backend architecture efficiently handles data ingestion, preprocessing, training, prediction, and visualization. Additionally, the application provides several user interfaces for remote users and service providers to interact with real-time data, generate visual analytics like demand count plots, and download prediction results. Admin functionalities, such as managing route data and viewing prediction ratios, enhance system transparency and usability. The modularity and structured flow of the codebase make the system easy to maintain and scale, while the use of persistent model storage (via Joblib) ensures reusability without retraining, optimizing performance and resource usage.

REFERENCES

[1]. Guihaire, V.; Hao, J.K. Transit network design and scheduling: A global review. Transp. Res. Part A Policy Pract. 2008, 42, 1251–1273.

[2]. Magnanti, T.L.; Wong, R.T. Network design and transportation planning: Models and algorithms. Transp. Sci. 1984, 18, 1–55.

[3]. Borndörfer, R.; Grötschel, M.; Pfetsch, M.E. A Path-Based Model for Line Planning in Public Transport A Path-Based Model for Line Planning in Public Transport *. Technical Report. 2005. Available online: https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=http://webdoc.sub.gwdg.de/ebook/serien/ah/reports/ZIBreport/ZR-05-18.pdf&ved=2ahUKEwiQrJ6R6pyNAxVjVmwGHYLRJroQFnoECBwQAQ&usg=AOvVaw1KSs6HedcL39KmDr41Voqj (accessed on 5 May 2025).

[4]. Zhou, X.; Wei, G.; Zhang, Y.; Wang, Q.; Guo, H. Optimizing Multi-Vehicle Demand-Responsive Bus Dispatching: A Real-Time Reservation-Based Approach. Sustainability 2023, 15, 5909.

[5]. Pei, M.; Lin, P.; Du, J.; Li, X.; Chen, Z. Vehicle dispatching in modular transit networks: A mixed-integer nonlinear

programming model. Transp. Res. Part E Logist. Transp. Rev. 2021, 147, 102240.

[6]. Reddy, S. K. (2025). Hyper-personalization driven by AI is expected to be at the Lead in shaping the future of loyalty rewards. Journal of Emerging Technologies and Innovative Research.

[7]. van Oudheusden, D.L.; Zhu, W. Trip frequency scheduling for bus route management in Bangkok. Eur. J. Oper. Res. 1995, 83, 439–451.

[8]. Gkiotsalitis, K.; van Berkum, E.C. An exact method for the bus dispatching problem in rolling horizons. Transp. Res. Part C Emerg. Technol. 2020, 110, 143–165.

[9]. Gkiotsalitis, K. A model for the periodic optimization of bus dispatching times. Appl. Math. Model. 2020, 82, 785–801.

[10]. Nandigama, N. C. (2025). Leveraging Chatgpt for Multi-Language Data Engineering Code Generation in Distributed Analytics Systems. Journal of Informatics Education and Research.

[11]. Gkiotsalitis, K.; Cats, O. Reliable frequency determination: Incorporating information on service uncertainty when setting dispatching headways. Transp. Res. Part C Emerg. Technol. 2018, 88, 187–207.

[12]. Hadas, Y.; Shnaiderman, M. Public-transit frequency setting using minimum-cost approach with stochastic demand and travel time. Transp. Res. Part B Methodol. 2012, 46, 1068–1084.

[13]. Eranki, A. A Model to Create Bus Timetables to Attain Maximum Synchronization Considering Waiting Times at Transfer Stops. Master's Thesis, Department of Industrial and Management Systems Engineering University of South Florida, Tampa, FL, USA, 2004. Available online: http://scholarcommons.usf.edu/etd/1025 (accessed on 15 February 2025).

[14]. Gorev, A.; Popova, O.; Solodkij, A. Demand-responsive transit systems in areas with low transport demand of "smart city". Transp. Res. Procedia 2020, 50, 160–166.

[15]. Naga Charan Nandigama, "Data-Driven Cyber-Physical Customer Experience Management In Iort-Enabled Banking Infrastructures," International Journal of Data Science and IoT Management System, vol. 2, no. 3, pp. 22–27, Aug. 2023, doi: 10.64751/ijdim.2023.v2.n3.pp22-27.

[16]. Berrebi, S.J.; Watkins, K.E.; Laval, J.A. A real-time bus dispatching policy to minimize passenger wait on a high frequency route. Transp. Res. Part B Methodol. 2015, 81, 377–389.

[17]. Gkiotsalitis, K.; Alesiani, F. Robust timetable optimization for bus lines subject to resource and regulatory constraints. Transp. Res. Part E Logist. Transp. Rev. 2019, 128, 30–51.

[18]. Yao, E.; Liu, T.; Lu, T.; Yang, Y. Optimization of electric vehicle scheduling with multiple vehicle types in public transport. Sustain. Cities Soc. 2020, 52, 101862.