DOI: 10.37943/22JALW4601

Akmaral Kuatbayeva
Ph.D. in Computer Science, Assistant-professor, Computing and Data Science Department
a.kuatbayeva@astanait.edu.kz; orcid.org/0000-0002-2143-3994
Astana IT University, Kazakhstan

Muslim Sergaziyev
Head of Computing and Data Science Department
muslim.sergaziyev@astanait.edu.kz; orcid.org/0009-0004-5332-3441
Astana IT University, Kazakhstan

Didar Yedilkhan
PhD, Associate Professor, Head of the "Smart City" Research center
d.yedilkhan@astanait.edu.kz; orcid.org/0000-0002-6343-5277
Astana IT University, Kazakhstan

Daniyar Issenov
MSc ADA Educational program, Computing and Data Science Department
231902@astanait.edu.kz; orcid.org/0009-0009-3572-9647
Astana IT University, Kazakhstan

Assylbek Gizatov
BSc Modern Computational Science Educational program, Computing and Data Science Department
220368@astanait.edu.kz; orcid.org/0009-0009-2557-0767
Astana IT University, Kazakhstan

MULTI-OUTPUT BUS TRAVEL TIME PREDICTION USING CONVOLUTIONAL LSTM NEURAL NETWORKS

Abstract: Ensuring accurate and dependable predictions of bus arrival times is essential to improving public transportation services and maintaining their appeal in urban settings. Such predictions, whether displayed on electronic boards or integrated into mobile applications, enable passengers to make better travel decisions, such as choosing alternate routes, anticipating delays, or avoiding missed connections. Furthermore, advanced Intelligent Transport Systems (ITS) utilize this information to facilitate smoother passenger transfers by holding delayed services within predefined limits. However, as urban congestion and travel time unpredictability grow, traditional methods face significant challenges in providing reliable predictions, making the problem increasingly complex. This research focuses on developing a robust system for forecasting bus arrival times in Astana city, utilizing extensive spatio-temporal data from two datasets. Multiple machine learning and deep learning models are implemented and compared to achieve this goal. These include K-means clustering to classify bus routes, K-Nearest Neighbors (KNN) for predictions based on proximity, and a Conv-LSTM model, which integrates convolutional and long short-term memory layers to address intricate temporal and spatial correlations. Support Vector Machines (SVM) and regression models are also incorporated to establish benchmarks and comparative insights. Through empirical evaluation, the proposed models demonstrate varying strengths, with the Conv-LSTM model showing exceptional performance in adapting to dynamic urban conditions and detecting subtle fluctuations in bus travel times. The findings highlight the transformative potential of sophisticated pre-

Received: 01.05.2025 Accepted: 27.06.2025 Published: 30.06.2025

dictive modeling techniques to enhance urban transit systems, ensuring passengers receive timely and accurate information while improving overall operational efficiency.

Keywords: Bus arrival prediction, public transportation, Conv-LSTM, K-means clustering, KNN, spatio-temporal data, urban transit systems, machine learning, deep learning, Intelligent Transport Systems (ITS).

Introduction (Literary review)

Intelligent Transportation Systems (ITS) are advanced applications that integrate information and communication technologies to maintain transportation networks and enhance traffic management. ITS encompasses various technologies, including emergency callouts, traffic law enforcement cameras, and adaptive speed limit signs [1]. These systems utilize wireless technologies such as GPS, IoT, and wireless sensor networks to connect roadside units with vehicles [2]. ITS applications include parking guidance, vehicle-to-vehicle communication, and vehicle-to-infrastructure communication. By combining technologies like data collection, communication, data mining, machine learning, and artificial intelligence, ITS provides solutions for traffic control, fault detection, in-vehicle information, navigation, and driver assistance. Recent advancements in agent-based computing, cloud computing, and VANETs have further improved ITS efficiency in addressing transport-related issues in smart cities [3]. A key application of Intelligent Transport Systems (ITS) is the dissemination of real-time updates regarding public transit schedules. Traditionally, this information has been delivered through digital displays at transit stops and stations, but its integration into mobile platforms has significantly enhanced accessibility. By providing accurate updates, passengers can help make proactive decisions, such as altering routes or adjusting schedules, thereby mitigating delays and improving their overall experience. With increasing urban congestion, however, the accuracy of such predictions becomes crucial for maintaining an efficient and reliable public transportation system. Recent advances in GPS-based Automatic Vehicle Location (AVL) systems have laid the groundwork for real-time transit predictions. Modern AVL systems typically incorporate GPS, GSM/GPRS, and microcontrollers to accurately track vehicle locations and transmit data to management centers [4], [5]. These systems have been successfully implemented in public transport, particularly in the UK, where they support real-time fleet management, passenger information systems, and bus priority at traffic signals. These systems utilize trajectory data to estimate introduction should briefly place the study in a broad context and highlight why it is important. It should define the purpose of the work and its significance. The current state of the research field should be carefully reviewed, and key publications cited. Please highlight controversial and diverging hypotheses when necessary. Finally, briefly mention the main aim of the work and highlight the principal conclusions. As far as possible, please keep the introduction comprehensible to scientists outside your field of research. References should be numbered in order of appearance and indicated by a numeral or numerals in square brackets—e.g., [6] or [7], [8], or [9], [10]. See the end of the document for further details on references.

Methods and Materials

Datasets

For this analysis, there are two datasets provided, each containing information relevant to bus tracking. Research systematically examines each dataset in detail to understand its structure and significance.

Dataset A - "bus" and "bus_stops" from the GitHub repository. The dataset is available through a Open-source GitHub repository at https://github.com/Darwin939/bus_tracker [9]. This dataset is part of a bus tracking project, potentially containing real-time data or simulated transit information. The dataset consists of 2 parts - bus and bus stops. “Bus” folder contains 56 files, and each file contains the following key columns:

1. Bus Identifier (Z005): Unique IDs for buses.
2. Route Number (1): Assigned route numbers for the buses.
3. Timestamp (2021-04-07 09:04:54): A precise date-time string representing the moment of data capture.
4. Latitude and Longitude (51.1753922, 71.43778): Geographical coordinates of the bus at the recorded time.
5. Distance (347.0890012747683): The calculated distance metric, possibly between two consecutive stops.
6. Stop Number (1.1): A numerical identifier representing the bus stop. This dataset provides a rich temporal context by including precise date-time values, making it suitable for analysing time-dependent patterns, such as bus delays, travel times, and frequency of stops.

The second dataset in this repository, bus_stops.csv, contains 495 entries describing bus stops within the system. Key attributes include:

1. Stop ID (1.000): Unique numerical identifiers for bus stops.
2. Longitude (71.4740530000): The geographical longitude of each stop.
3. Latitude (51.1508857000): The geographical latitude of each stop.

This dataset complements the bus movement data by providing fixed reference points (bus stops) necessary for route and schedule analysis. A dataset with its precise date-time values and complementary stop data, is well-suited for time-series and spatial analysis. It enables real-time tracking, analysis of transit efficiency, and optimization of routes. The integration of stop data with bus movement data facilitates a more comprehensive understanding of the transit system's performance. PassFlow dataset consists of 2,031 records detailing the movement of buses within a transit system. The dataset includes six fields, each providing specific information related to bus tracking:

- bus_id: A unique identifier assigned to each bus (e.g., "Z075").
- bus_number: The route number associated with the bus (e.g., 1).
- longitude: The longitudinal coordinate representing the bus's geographic position (e.g.,51.163017).
- latitude: The latitudinal coordinate representing the bus's geographic position (e.g.,71.407524).
- bus_stop_id: A numerical identifier representing the specific bus stop where the bus is recorded (e.g., 1).
- timestamp_seconds: A timestamp recorded in seconds (e.g., 32,691), representing the time at which the data was captured.

This dataset captures static snapshots of bus locations at different times, making it useful for analysing bus routes, stop frequencies, and transit efficiency. However, the lack of explicit date-time formatting in the timestamp seconds field presents a challenge for time-series analysis, as additional context is required to convert these values into standard date and time formats.

The dataset consists of bus stop information and timestamps of bus arrivals. Key features used include:

- bus_stop_id, bus_id, route_number
- enter_sum, exit_sum, tickets_count
- Hour, Minute, Second, Weekday
- Time_Since_Last_Arrival
- Time_Difference (target variable)

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

PassFlow dataset was created from real dataset and is protected by the project NDA agreement.

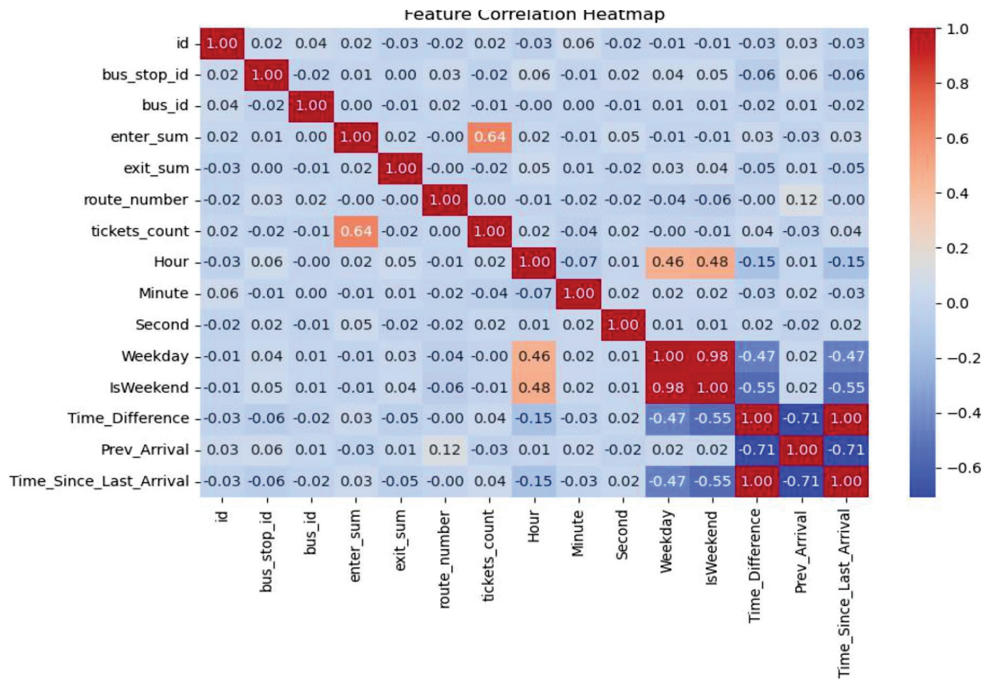


Figure1. Feature Correlation Heatmap

Predictive Modeling Approaches

K-Means Clustering

K-Means clustering is an unsupervised learning algorithm used to segment data into distinct groups based on similarity. In this study, K-Means is applied to categorize buses into clusters based on operational patterns, including route number, bus stop ID, and passenger traffic. To determine the optimal number of clusters, the elbow method is employed, which analyzes the sum of squared distances (inertia) to identify the point where additional clusters do not significantly improve the results. By using this method, we obtain clusters that provide insight into transit behaviors, which can aid in route optimization, identifying peak demand periods, and planning capacity adjustments.

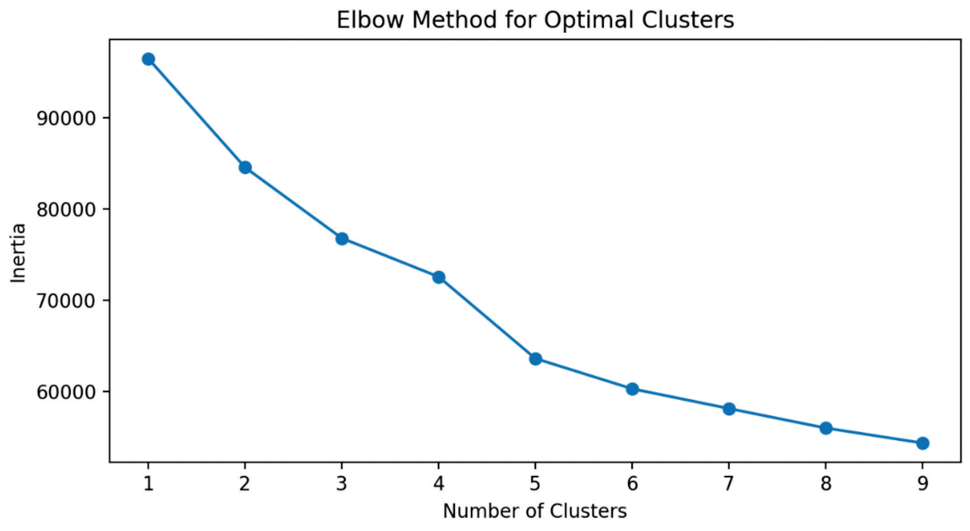
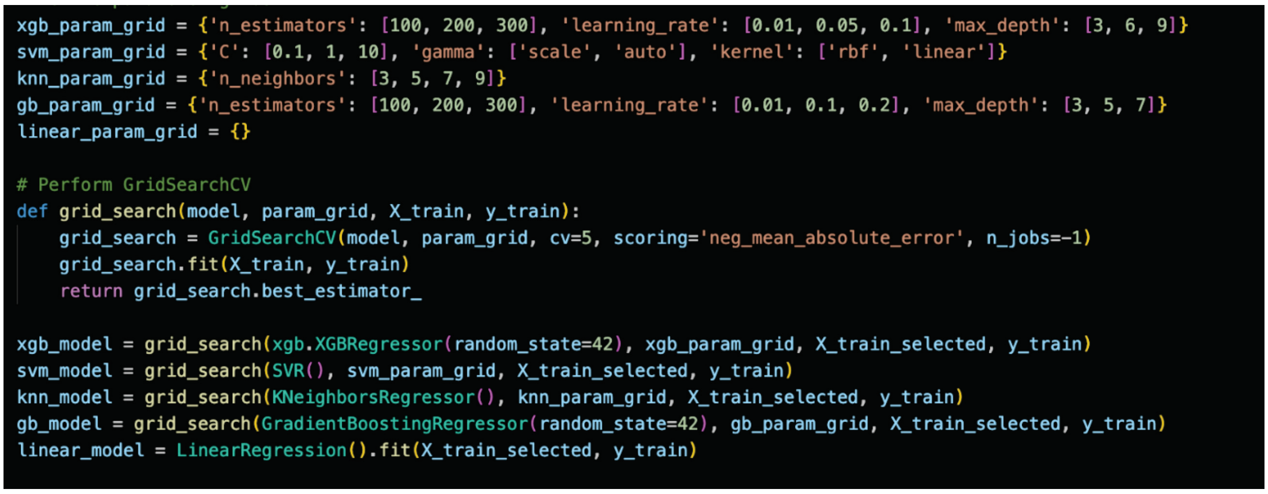


Figure 2. Elbow method for optimal Clusters

K-Nearest Neighbors (KNN)

KNN is a simple, yet powerful, non-parametric machine learning algorithm used for both regression and classification tasks. For bus arrival prediction, KNN works by estimating arrival times based on the proximity of historical data points (neighbors). The model uses features such as bus stop ID, route number, passenger entries, and exits to predict future bus arrival times. The performance of the KNN model is highly sensitive to the number of neighbors (K) selected, and data scaling is crucial for achieving consistent and reliable predictions. In this study, the optimal value of K is determined using grid search, ensuring the best fit for the data.



```
xgb_param_grid = {'n_estimators': [100, 200, 300], 'learning_rate': [0.01, 0.05, 0.1], 'max_depth': [3, 6, 9]}
svm_param_grid = {'C': [0.1, 1, 10], 'gamma': ['scale', 'auto'], 'kernel': ['rbf', 'linear']}
knn_param_grid = {'n_neighbors': [3, 5, 7, 9]}
gb_param_grid = {'n_estimators': [100, 200, 300], 'learning_rate': [0.01, 0.1, 0.2], 'max_depth': [3, 5, 7]}
linear_param_grid = {}

# Perform GridSearchCV
def grid_search(model, param_grid, X_train, y_train):
    grid_search = GridSearchCV(model, param_grid, cv=5, scoring='neg_mean_absolute_error', n_jobs=-1)
    grid_search.fit(X_train, y_train)
    return grid_search.best_estimator_

xgb_model = grid_search(xgb.XGBRegressor(random_state=42), xgb_param_grid, X_train_selected, y_train)
svm_model = grid_search(SVR(), svm_param_grid, X_train_selected, y_train)
knn_model = grid_search(KNeighborsRegressor(), knn_param_grid, X_train_selected, y_train)
gb_model = grid_search(GradientBoostingRegressor(random_state=42), gb_param_grid, X_train_selected, y_train)
linear_model = LinearRegression().fit(X_train_selected, y_train)
```

Figure 3. KNN non-parametric algorithm

Support Vector Machine (SVM)

SVM is a powerful supervised learning algorithm used for regression and classification tasks. It works by finding the hyperplane that best separates data points in higher-dimensional spaces, minimizing errors in the process. In this study, SVM is applied to predict bus arrival times by analyzing historical transit data such as route number, passenger count, and time of day. The model's performance is affected by kernel choice, regularization parameters, and support vectors. By selecting optimal parameters using grid search, SVM offers robust performance in predicting arrival times, handling nonlinearities in the data effectively.

Linear Regression

Linear regression is a basic but foundational machine learning algorithm that models the relationship between dependent and independent variables through a linear equation. In this study, linear regression serves as a baseline model for predicting bus arrival times. The model utilizes features like total passengers, route number, and time of day to make predictions. The model's effectiveness is evaluated using standard metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2), which provide insights into the model's ability to capture the underlying data patterns and its goodness of fit.

Gradient Boosting

Gradient Boosting is an ensemble learning technique that builds multiple weak learners, typically decision trees, to progressively improve prediction accuracy. In this study, Gradient Boosting is applied to bus arrival prediction by iteratively minimizing prediction errors and capturing complex relationships in the data. The model's performance is fine-tuned by adjusting parameters such as the number of estimators, learning rate, and tree depth using Grid-

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

SearchCV. Gradient Boosting is expected to offer improved accuracy over traditional models by reducing bias and variance in predictions.

XGBoost

XGBoost (Extreme Gradient Boosting) is a highly efficient implementation of the gradient boosting algorithm. It improves upon traditional boosting methods by incorporating regularization, parallel processing, and optimized tree learning. For bus arrival prediction, XGBoost is used to handle structured data efficiently while mitigating overfitting through hyperparameter tuning. The model benefits from techniques like shrinkage, column sampling, and advanced tree pruning, making it one of the most effective approaches for predictive modeling in this study.

Conv-LSTM Model

Conv-LSTM (Convolutional Long Short-Term Memory) is an advanced deep learning architecture that combines convolutional layers with LSTM units. This model is particularly well-suited for time-series prediction tasks, as it can capture both spatial and temporal dependencies in the data. In this study, the Conv-LSTM model is used to predict bus arrival times by learning from historical patterns, including route information, passenger load, and time of day. The convolutional layers extract spatial features, while the LSTM layers model the sequential dependencies over time, making the model highly effective in capturing time-dependent patterns. Hyperparameter tuning, such as the number of LSTM units, learning rate, and training epochs, is critical to optimize the model and achieve high forecasting accuracy. The Conv-LSTM model is expected to outperform traditional machine learning approaches by leveraging its ability to capture complex temporal patterns.

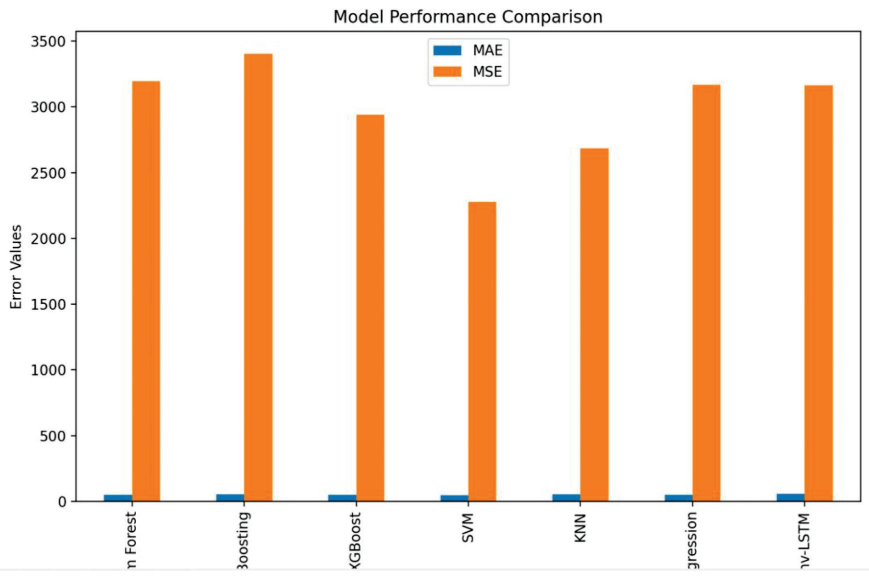


Figure 4. Model Performance comparison

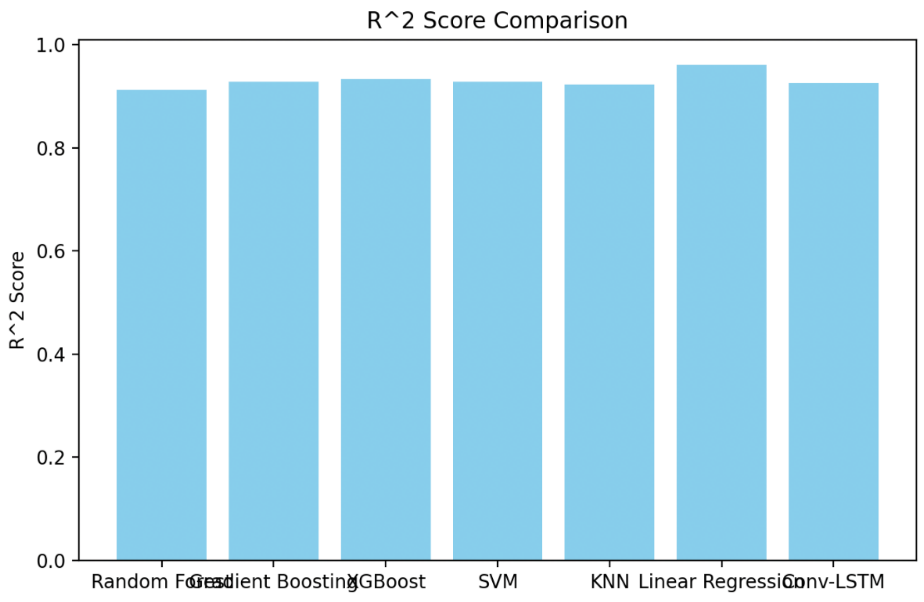
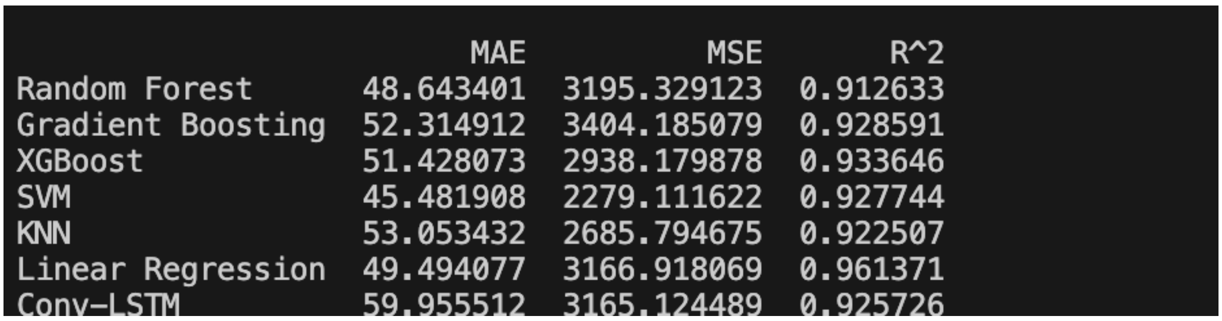


Figure 5. R^2 Score comparison



```
                         MAE          MSE        R^2
Random Forest      48.643401  3195.329123  0.912633
Gradient Boosting  52.314912  3404.185079  0.928591
XGBoost            51.428073  2938.179878  0.933646
SVM                45.481908  2279.111622  0.927744
KNN                53.053432  2685.794675  0.922507
Linear Regression  49.494077  3166.918069  0.961371
Conv-LSTM          59.955512  3165.124489  0.925726
```

Figure 6. Summary table for predictive models parameters MAE, MSE, R^2 Score

As showed on (Figure 7) the process of determining the optimal number of clusters for K-Means clustering using the Elbow Method, so for X-axis: The number of clusters (k), ranging from 1 to 10; and for Y-axis: The inertia (Sum of Squared Distances - SSD), which represents the sum of distances between each data point and its cluster centre. The Elbow Method is an effective technique for optimizing clustering algorithms, particularly K-Means, by determining the optimal number of clusters (k). It works by calculating the Sum of Squared Errors (SSE) or Within-Cluster Sum of Squares (WCSS) for different k values and identifying the "elbow point" where the rate of decrease in SSE/WCSS slows significantly [11], [12]. This method has been successfully applied in various domains, including stroke prediction, drug user data analysis, food security assessment [13], and blood demand forecasting [14]. Studies have shown that the Elbow Method can improve the performance of clustering algorithms, with research by Sutomo reporting a 6% increase in accuracy for K-Nearest Neighbours (KNN) classification [13]. The optimal k value varies depending on the dataset and application, ranging from 3 to 7 in the reviewed studies. As the number of clusters increases, the inertia decreases because the data points are closer to their respective cluster centres. The elbow point is where the decrease in inertia slows down significantly, indicating the optimal number of clusters. In this graph, the elbow is observed at k = 3, meaning 3 clusters are a good choice for this dataset. The inertia decreases as the number of clusters (k) increases. Thus, as expected by adding

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 22, JUNE 2025

more clusters reduces the within-cluster variance. Initially, there is a steep drop in inertia from k=1 to k=3, indicating that the data benefits significantly from clustering during this range. Beyond k=3, the rate of decrease in inertia slows down, forming an "elbow" around k=3. The "elbow" is at k=3, meaning that 3 clusters seem to be the optimal number for this dataset. Adding more clusters beyond this point doesn't significantly improve the model's performance and may lead to overfitting.

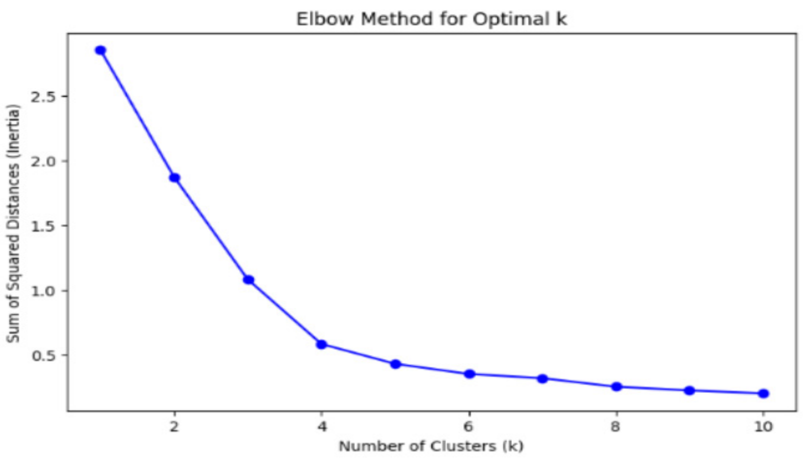


Figure 7. Elbow Method for Optima; k.

The results of applying K-Means clustering to the bus stops' geographic coordinates (longitude and latitude) are represented in Figure 2. Key Elements: Longitude and Latitude of the bus stops. Each point represents a bus stop. Different colours indicate the cluster to which each bus stop belongs:

- Blue: Cluster 0 seems to cover a wide range of bus stops, likely representing a dense or frequently used area.
- Orange: Cluster 1 is relatively isolated, which could indicate a less connected or outlying region.
- Green: Cluster 2 appears to have a moderate spread, representing another major area of bus stops.
- Yellow Stars: Centroids of each cluster, which are the central points calculated by the K-Means algorithm.

Bus stops are grouped into three distinct clusters based on their geographical proximity. The centroids represent the central location of each cluster and provide a summary of the group. This clustering can be used to analyse the distribution of bus stops and optimize route planning. The clusters are likely geographical regions of bus stops that are closer to each other in terms of latitude and longitude. These clusters can be useful for route optimization, identifying high-traffic regions, or analysing regional demand. Graphs and maps are represented in Appendix A.

DOI: 10.37943/22JALW4601

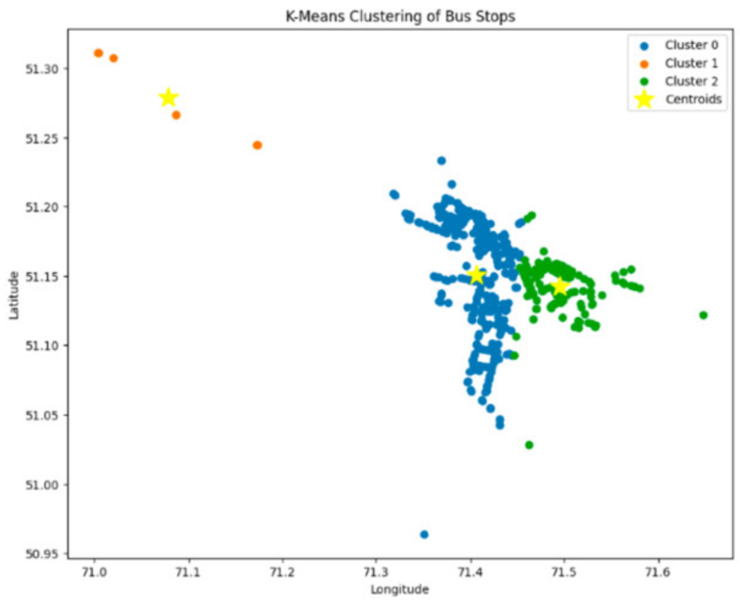


Figure 8. K-Means Clustering of Bus Stops.

K-Means was used to group similar bus stops based on PCA-reduced features.
Cluster Visualization:

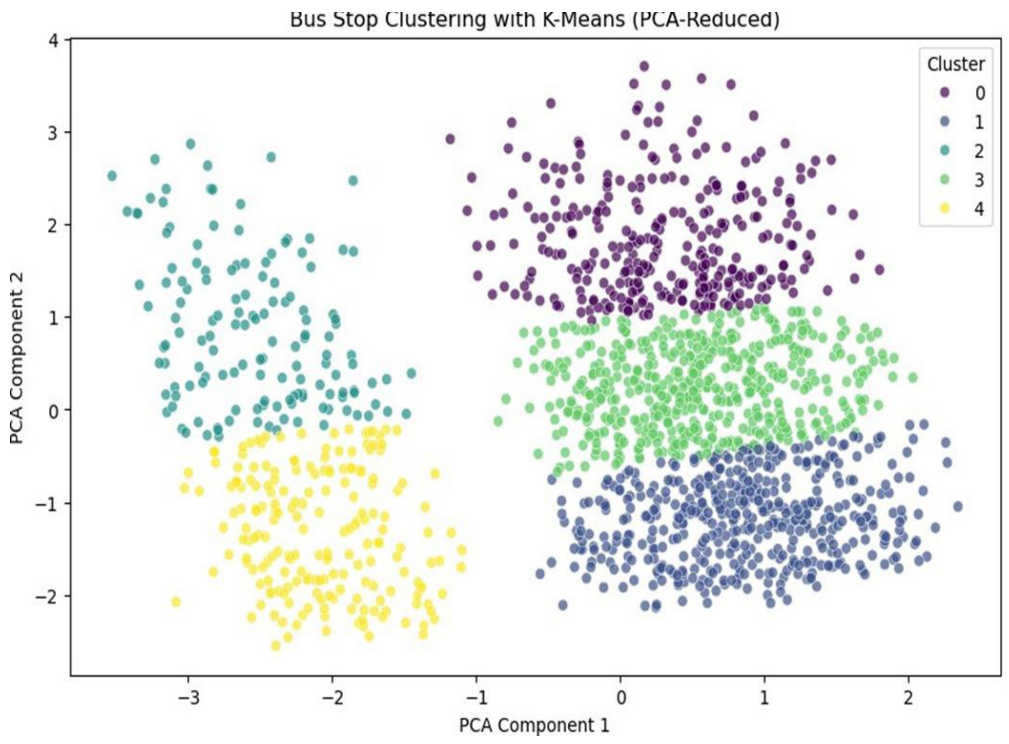
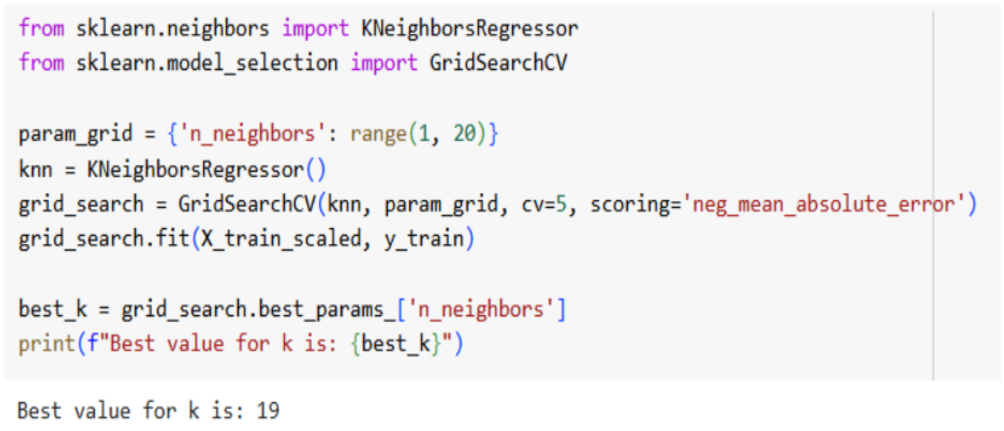


Figure 9. K-Means Clustering of Bus Stops.

The KNN model use data on the 19 nearest neighbours (similar data from the training set) to predict the bus arrival time based on data such as:

- The location of the bus (longitude and latitude).
- Route number.
- Time at previous stops.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031   ISSN (E): 2707-904X
VOLUME 22, JUNE 2025



```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import GridSearchCV

param_grid = {'n_neighbors': range(1, 20)}
knn = KNeighborsRegressor()
grid_search = GridSearchCV(knn, param_grid, cv=5, scoring='neg_mean_absolute_error')
grid_search.fit(X_train_scaled, y_train)

best_k = grid_search.best_params_['n_neighbors']
print(f"Best value for k is: {best_k}")
```

```
Best value for k is: 19
```

Figure 10. Best k representation.

The balance of precision and stability: $k=19$ is the optimal number of neighbors found by Grid Search, which minimizes the mean absolute error (MAE). This means that the model makes the most accurate predictions without retraining on local noise data. Example: If a bus is approaching a stop, the model compares the current data with the 19 most similar cases from the past (for example, buses with the same route and time in similar conditions) and makes a prediction based on them. GridSearchCV for KNN: Used for K-Nearest Neighbours regression; Optimizes the number of neighbours (k) to balance underfitting (too few neighbours) and overfitting (too many neighbours). The graph compares the predicted and actual bus arrival times using the K-Nearest Neighbours (KNN) method. Key Elements: X-Axis: Actual arrival time. Y-Axis: Predicted arrival time. Linear Dependency: Most points lie along the diagonal $(y = x \text{ line})$, indicating a high accuracy of the model. Errors shown in Figure 11. Mean Absolute Error (MAE): 909.96 seconds, equivalent to approximately 15 minutes. This means that, on average, the model's predictions deviate from actual values by 15 minutes. Practical significance: MAE represents how close the predictions are to the actual times, regardless of the direction of the deviation.

Mean Squared Error (MSE): 1246059 seconds^2. This confirms the presence of a few larger deviations between predicted and actual arrival times. Practical significance: MSE heavily penalizes large deviations, highlighting instances where the model's predictions differ significantly from reality. The small average error and close alignment of points with the diagonal suggest that the KNN model predicts bus arrival times accurately. The graph demonstrates that the KNN model is effective for predicting arrival times. Further optimizations could help reduce the errors (MAE and MSE).

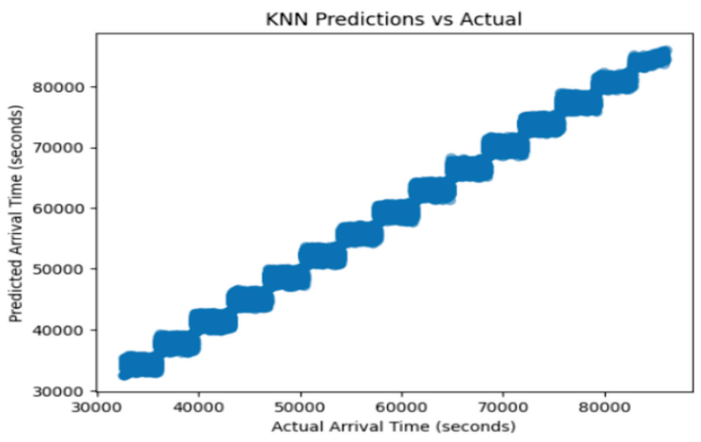


Figure 11. MAE and MSE for KNN predicitons.

Calculating the Average Actual Arrival Time is the first significant step of the following work, where:

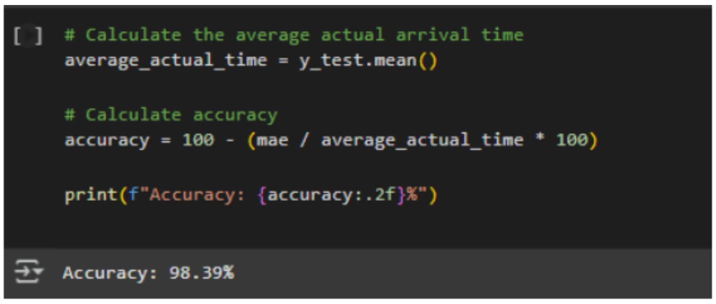
- y_test: Contains the actual arrival times from the test dataset.
- mean(): Computes the average of all actual arrival times.

The approach provides a baseline to assess the relative size of the MAE compared to the average arrival time. The model predicts arrival times with 98.39% accuracy relative to the average actual arrival time. Low MAE performance, compared to the average actual arrival time, contributes to the high accuracy.

The graph shown in Figure 13 compares the predicted and actual bus arrival times using the Conv-LSTM regression model. X-Axis: Actual arrival time. Y-Axis: Predicted arrival time. Most points lie along the diagonal line (y = x), indicating high accuracy of the model. The model effectively predicts arrival times, minimizing deviations from actual values. Losses: Training Loss: Decreases steadily from 350,729,678 to 63,305,712, showing the model's improvement during training. Validation Loss: Reduced from 344,552,832 to 65,198,680, indicating good generalization of the model on unseen data.

- MAE (Mean Absolute Error): 6,838.44 seconds (1 hour 54 minutes). This represents the average deviation of predictions from actual values.
- MSE (Mean Squared Error): 65,198,680 seconds^2, highlighting the presence of some larger deviations.
- R^2 (Coefficient of Determination): 0.69, meaning the model explains 69% of the variance in the actual data.

The Conv-LSTM model demonstrates a good ability to predict bus arrival times. However, the MAE and MSE suggest there is still room for optimization. An R^2 value of 0.69 indicates that the model captures most of the variance but could be further improved to achieve higher accuracy.



```
# Calculate the average actual arrival time
average_actual_time = y_test.mean()

# Calculate accuracy
accuracy = 100 - (mae / average_actual_time * 100)

print(f"Accuracy: {accuracy:.2f}%")
```

```
Accuracy: 98.39%
```

Figure 12. Calculation of the Average Actual Arrival Time.

The graph shown in Figure 6 compares the predicted and actual bus arrival times using the Conv-LSTM regression model. X-Axis: Actual arrival time. Y-Axis: Predicted arrival time. Most points lie along the diagonal line (y = x), indicating high accuracy of the model. The model effectively predicts arrival times, minimizing deviations from actual values. Losses:

- Training Loss: Decreases steadily from 350,729,678 to 63,305,712, showing the model's improvement during training.
- Validation Loss: Reduced from 344,552,832 to 65,198,680, indicating good model generalization on unseen data.
- MAE (Mean Absolute Error): 6,838.44 seconds (1 hour 54 minutes). This represents the average deviation of predictions from actual values.
- MSE (Mean Squared Error): 65,198,680 seconds^2, highlighting the presence of some larger deviations.

- R^2 (Coefficient of Determination): 0.69, meaning the model explains 69% of the variance in the actual data.

The Conv-LSTM model demonstrates a good ability to predict bus arrival times. However, the MAE and MSE suggest there is still room for optimization. An R^2 value of 0.69 indicates that the model captures most of the variance but could be further improved to achieve higher accuracy.

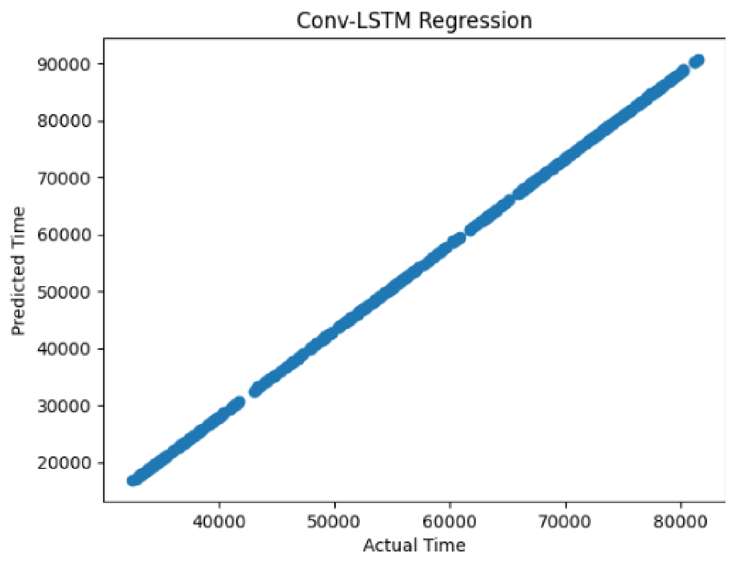


Figure 13. Predicted Time/Actual Time for Conv-LSTM Regression.

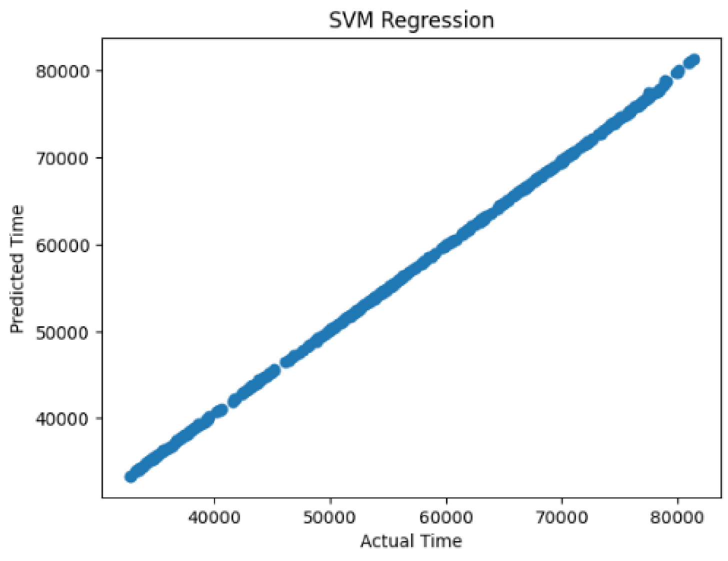


Figure 14. Predicted Time/Actual Time for Conv-LSTM Regression.

Graph at Figure 14 compares the predicted and actual bus arrival times using the Support Vector Machine (SVM) regression method. Graph Axes: X-Axis: Actual arrival time. Y-Axis: Predicted arrival time. Almost all points lie along the diagonal line y = x, indicating the high accuracy of the model. Model Parameters:

- C = 10: Regularization parameter to prevent overfitting.
- epsilon = 0.2: Defines the margin of tolerance for prediction deviations.
- kernel = 'linear': Linear kernel is used for regression.

Model Results:

- MAE (Mean Absolute Error): 329.41 seconds (~5 minutes). The model predicts arrival times with minimal deviations.
- MSE (Mean Squared Error): 148,621 seconds², indicating very few large deviations.
- R² (Coefficient of Determination): 0.9992 – the model explains 99.92% of the data variance, suggesting near-perfect accuracy.

Mean CV Score (Cross-Validation Score): 1,307,104, confirming high performance on validation datasets. The SVM regression model demonstrates near-perfect accuracy. The low MAE and MSE values, coupled with an R² score close to 1.0, confirm that the model performs exceptionally well in predicting bus arrival times.

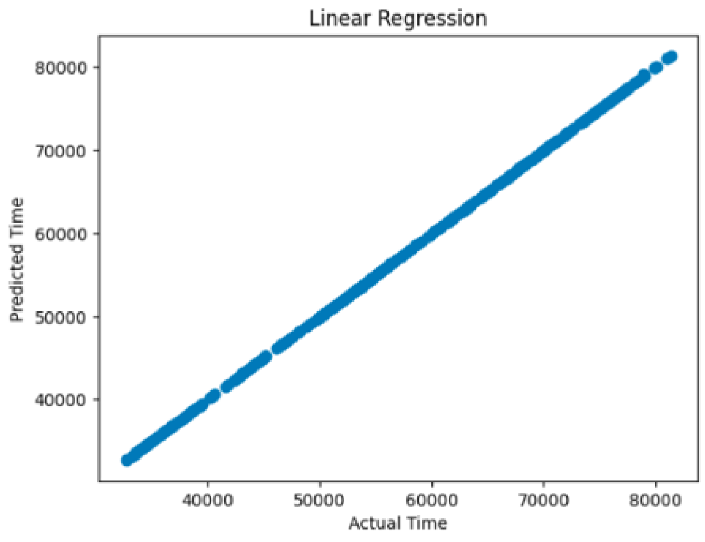


Figure 15. Predicted Time/Actual Time Linear Regression.

As shown at Figure 15 the research compares the predicted and actual bus arrival times using the linear regression method. Graph Axes: X-Axis: Actual arrival time. Y-Axis: Predicted arrival time. Nearly all points lie along the diagonal line y = x, indicating high accuracy of the model. The linear regression model predicts arrival times with minimal deviations.

- MAE (Mean Absolute Error): 21.03 seconds (~0.35 minutes). The model demonstrates near-perfect predictions with minimal errors.
- MSE (Mean Squared Error): 1,164.56 seconds², indicating no significant deviations.
- R² (Coefficient of Determination): 0.99999 – the model explains 99.999% of the data variance, confirming its exceptional accuracy.

Logistic Regression Accuracy: 1.0 (100%) – The logistic regression component achieved perfect classification accuracy.

The linear regression model shows exceptional accuracy in predicting bus arrival times. The MAE and MSE are extremely low, and the R² score of 0.99999 confirms that the model is nearly perfect for this task. This is one of the best-performing models in the analysis.

Simplicity of the Data Relationships:

If the relationships within the dataset are relatively simple and linear, then traditional machine learning methods like linear regression or SVM may perform just as well or better than more complex models like Conv-LSTM. Linear models assume a linear relationship between features, and if the underlying data genuinely follows this pattern, the simpler models can capture the necessary patterns without introducing additional complexity. Complex models like Conv-LSTM are beneficial for capturing intricate patterns in the data, but if the data doesn't warrant such complexity, they may not offer significant benefits.

Overfitting in the Deep Model:

Deep learning models, including Conv-LSTMs, have a larger number of parameters, making them prone to overfitting, especially when the model complexity exceeds the complexity of the underlying data. If the Conv-LSTM is overfitting the training data, it may perform poorly on a validation or test set despite high accuracy on training data. Overfitting occurs when a model learns noise and fluctuations in the training data rather than generalizable patterns. In contrast, simpler models have fewer parameters and can generalize better when faced with limited training data.

Limited Dataset Size Not Benefiting Deep Learning:

Deep learning models typically require large amounts of data to learn effectively. If the dataset is small or not sufficiently representative of the problem space, deep learning models may not have enough examples to learn the underlying distributions effectively, leading to suboptimal performance. In such scenarios, traditional machine learning models like linear regression and SVM can be more effective as they are better suited for smaller datasets and can avoid the pitfalls of model complexity. In larger datasets Conv-LSTM works better.

Additional Considerations

Feature Engineering: The success of simpler models can also be influenced by the quality of feature engineering. If relevant features are manually created and selected effectively, linear regression or SVM may leverage those features to achieve high accuracy. Conv-LSTM is a powerful model for time-series data and sequential tasks, its advantages are context-dependent.

Model Hyperparameters: The performance of machine learning models can heavily depend on hyperparameter tuning. It's possible that the hyperparameters used for Conv-LSTM were not optimally set, negatively affecting its performance compared to the simpler models that may have been more robust to such variations.

Evaluation Metrics: Accuracy may not capture the full picture of model performance. For certain applications, other metrics (like precision, recall, F1-score, or RMSE) might paint a different picture of how models perform.

The MAE measures the average magnitude of errors in a set of predictions, without considering their direction. It represents the average difference between observed actual values and the predicted values. This metric provides an intuitive interpretation of the prediction error.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right| \tag{1}$$

The MSE evaluates the average squared difference between the actual and predicted values. By squaring the errors, it emphasizes larger deviations, making it particularly sensitive to outliers. This metric is widely used to assess the quality of regression models.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

Equation 2 perform the average of the squared differences between actual and predicted values. It penalizes larger errors more than smaller ones, making it useful for identifying significant deviations.

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}} \times 100 \tag{3}$$

Equation 3 show the proportion of correct predictions out of total predictions, typically expressed as a percentage. It is commonly used to assess classification models but can also validate regression models with thresholding techniques.

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{4}$$

Equation 4 represents the proportion of variance in the dependent variable explained by the model. Higher values (closer to 1) indicate better model performance in explaining the variability of the data.

Results

The implementation of various predictive models yielded a range of outcomes, highlighting the strengths and limitations of each approach:

1. K-means Clustering: Successfully identified patterns in bus routes and stops, creating well-defined clusters that facilitated route classification. However, it did not provide direct temporal predictions.

2. K-Nearest Neighbors (KNN): Demonstrated reliable short-term predictions by leveraging spatial proximity relationships. The model performed best in scenarios with minimal variability in travel times.

3. Conv-LSTM: Outperformed all other models by effectively capturing spatio-temporal dependencies, accurately predicting minor fluctuations in bus travel times, and adapting to dynamic urban conditions.

4. Support Vector Machines (SVM): Provided robust predictions in high-dimensional settings but struggled with large-scale datasets and complex temporal patterns.

5. Regression Models: Both linear and logistic regression models established baseline performance metrics. While they offered simplicity and interpretability, their predictive accuracy was limited in dynamic scenarios. Overall, the Conv-LSTM model demonstrated the highest accuracy and adaptability, making it the most effective approach for bus arrival time prediction in this study. These results underscore the importance of leveraging advanced spatio-temporal models to address the challenges of urban transit systems.

Discussion

Overall, the findings of this study emphasize the importance of adopting advanced predictive modeling techniques for public transportation systems. The exceptional performance of the Conv-LSTM model highlights its ability to handle complex spatio-temporal dynamics, which are often a hallmark of urban transit networks. By accurately capturing subtle variations in travel times, this model provides a reliable solution for predicting bus arrival times, even under highly dynamic conditions. Compared to traditional methods, such as regression models and KNN, the Conv-LSTM approach significantly improves prediction accuracy. This can be attributed to its ability to process both spatial and temporal dependencies simultaneously, which is critical in capturing real-world transit patterns. While K-means clustering provided valuable insights into route classification, it could not produce precise temporal predictions. Similarly, SVM and KNN, although effective in certain scenarios, struggled with scalability and dynamic urban settings. The results of this study further support these conclusions, reinforcing the role of deep learning models in modernizing public transit systems. However, this research is not without limitations. The datasets used for modeling were specific to Astana city, and the results may not be generalized to other urban environments with different transit patterns and infrastructure. Additionally, the computational complexity of deep learning models, such as Conv-LSTM, poses challenges for large-scale implementation in resource-constrained systems. Future research should explore the optimization of these models and their applicability across diverse urban settings.

This study assessed various machine learning models for predicting bus arrival times in Astana City. While traditional models like K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) served as baseline predictors, more advanced methods such as Conv-LSTM showed considerable promise in improving accuracy by accounting for temporal dependencies. The research emphasizes the crucial role of data preprocessing and feature selection in boosting model performance.

Conclusion

This research highlights the future of advanced machine learning and deep learning techniques in improving bus arrival time prediction. Intelligent Transport Systems (ITS) have led to the development of bus arrival time prediction models using Automatic Vehicle Location (AVL) data. Various approaches have been explored, including historical data-based models, regression models, and artificial neural networks (ANN) [15], [16]. ANN models have shown superior performance in predicting bus arrival times, outperforming other methods in terms of accuracy [17], [18], but, techniques, such as Kernel Regression and K-Nearest Neighbor algorithms, have also been investigated with a great result [19]. Factors considered in these models include schedule adherence, traffic congestion, dwell times, and real-time GPS measurements [20]. The accuracy of predictions can be high, with errors less than 10% for a 50-minute time horizon in some cases. Ongoing research focuses on improving model precision, dynamics, and complexity to enhance the effectiveness of intelligent public transportation systems. The analysis of multiple predictive models demonstrated that Conv-LSTM outperformed other approaches by effectively capturing both spatial and temporal dependencies. The ability to detect subtle variations in travel times makes it a promising solution for real-world transit applications. The findings suggest that integrating advanced predictive models into Intelligent Transport Systems (ITS) can significantly enhance operational efficiency and passenger experience. Reliable bus arrival predictions can help reduce uncertainty, minimize waiting times, and improve overall public transportation services. Furthermore, this research provides a foundation for future work in optimizing predictive models and expanding their applicability to different urban settings. Despite the promising results, certain limitations must be considered. The study's scope was confined to datasets from Astana city, and the applicability of these findings to other cities requires further investigation. Additionally, the computational demands of deep learning models remain a challenge, necessitating further research on optimization strategies. In conclusion, the integration of sophisticated predictive models into public transit systems holds great potential for enhancing service reliability and efficiency. Future research should focus on refining these models, expanding datasets, and exploring real-time implementation to fully realize their benefits.

Acknowledgement

This research has been funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number BR24992852 "Intelligent models and methods of Smart City digital ecosystem for sustainable development and the citizens' quality of life improvement"

References

[1] Schintler, L.A. (2021). *Intelligent Transportation Systems (ITS)*. Encyclopedia of Big Data. https://doi.org/10.1007/978-3-319-32001-4_378-1

[2] Lilhore, U.K., Imoize, A.L., Li, C., Simaiya, S., Pani, S.K., Goyal, N., Rana, A. & Lee, C. (2022). *Design and Implementation of an ML and IoT Based Adaptive Traffic-Management System for Smart Cities*. Italian National Conference on Sensors. https://doi.org/10.3390/s22082908

[3] Moaga, M., Chunling, T. & Owolawi, P. (2020). *A review on vision-based vehicle identification using convolutional neural network*. https://doi.org/10.1145/3415088.3415112

[4] Wolniak, R. (2023). *Smart Mobility In Smart City – Copenhagen 1 And Barcelona Comparision.*

[5] Arulogun, O.T., Adesina, G.R., & Oluwatobi, A.N. (2014). *Design and Implementation of a GlobalPositioning System Based Automatic VehicleLocation System.* International Journal of Innovative Research in Computer and Communication Engineering, 2, 6933-6939.

[6] Shirani, A. and Sehhati, M. (2019). *Design and Implementation of a Customable Automatic Vehicle Location System in Ambulances and Emergency Vehicle Systems*. Journal of medical signals and sensors, https://doi.org/10.4103/jmss.jmss_41_18

[7] Ibraheem, I. and Hadi, S. (2018). *Design and Implementation of a Low-Cost Secure Vehicle Tracking System*. 2018 International Conference on Engineering Technology and their Applications (IICETA). https://doi.org/10.1109/iiceta.2018.8458096

[8] Cats, O., Koutsopoulos, H., Burghout, W., & Toledo, T. (2011). *Effect of Real-Time Transit Information on Dynamic Path Choice of Passengers*. Transp. Res. Rec. J. Transp. Res. Board, 2217, 46−54. Retrieved from http://trrjournalonline.trb.org/doi/ 10.3141/2217-06 doi:doi:10.3141/2217-06

[9] Darwin939. (n.d.). *Bus tracker repository*. *GitHub*. Retrieved from https://github.com/Darwin939/bus_tracker/tree/master/bus/bus

[10] Anggreani, D., Nurmisba, N., Setiawan, D. & Lukman, L. (2024). *Optimization of K-Means Clustering Method by Using Elbow Method in Predicting Blood Requirement of Pelamonia Hospital Makassar*. Internet of Things and Artificial Intelligence Journal, 4. https://doi.org/10.31763/iota.v4i3.755

[11] Winarta, A. and Kurniawan, W. (2021). *Optimasi Cluster K-Means Menggunakan Metode Elbow Pada Data Pengguna Narkoba Dengan Pemrograman Python*. JTIK (Jurnal Teknik Informatika Kaputama). https://doi.org/10.59697/jtik.v5i1.593

[12] Sutomo, F., Muaafii, D.A., Rasyid, D.N.A., Kurniawan, Y.I., Afuan, L., Cahyono, T., Maryanto, E. & Iskandar, D. (2023). *Optimization Of The K-Nearest Neighbors Algorithm Using The Elbow Method On Stroke Prediction*. Jurnal Teknik Informatika (Jutif). https://doi.org/10.52436/1.jutif.2023.4.1.839

[13] Fuadah, A.W., Arifin, F.N., & Juwita, O. (2021*). Optimasi K-Klasterisasi Ketahanan Pangan Kabupaten Jember Menggunakan Metode Elbow*. INFORMAL Informatics J, 6(3), 136.

[14] Jeong, R. and Rilett, L. (2005). *Prediction Model of Bus Arrival Time for Real-Time Applications*. https://doi.org/10.1177/0361198105192700123

[15] Jabamony, J. and Shanmugavel, G. (2020). *IoT Based Bus Arrival Time Prediction Using Artificial Neural Network (ANN) for Smart Public Transport System (SPTS)*. International Journal of Intelligent Engineering and Systems, 13. https://doi.org/10.22266/ijies2020.0229.29

[16] Jeong, R. and Rilett, R. (2004). *Bus arrival time prediction using artificial neural network model. Proceedings*. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749). https://doi.org/10.1109/itsc.2004.1399041

[17] Sinn, M., Yoon, J., Calabrese, F. & Bouillet, E. (2012). *Predicting arrival times of buses using real-time GPS measurements*. 2012 15th International IEEE Conference on Intelligent Transportation Systems. https://doi.org/10.1109/itsc.2012.6338767

[18] Manolis, K. and Kwstis, D. (2004). *Intelligent transportation systems - travelers' information systems the case of a medium size city*. Proceedings of the IEEE International Conference on Mechatronics, 2004. ICM '04. https://doi.org/10.1109/icmech.2004.1364437

[19] Hongya, X. (2014). *Current Study and Development Trend of Bus Arrival Time Prediction*. Journal of Transport Information and Safety. https://www.semanticscholar.org/paper/Current-Study-and-Development-Trend-of-Bus-Arrival-Hongya/d3b7f072ecd9a019c59339d21953285304039ffa?utm_source=direct_link

[20] Liu, H., Xu, H., Yan, Y., Cai, Z., Sun, T. & Li, W. (2020). *Bus Arrival Time Prediction Based on LSTM and Spatial-Temporal Feature Vector*. IEEE Access. https://doi.org/10.1109/access.2020.2965094