Knowledge Graph-Enhanced Traffic Optimization System for the Bay Area

Anshu Reddy Dhamana
Master of Data Analytics
San Jose State University
anshu45reddy@gmail.com

Basanth Periyapatna Roopa Kumar
Master of Data Analytics
San Jose State University
basanthpr11@gmail.com

Manav Rajesh Anandani
Master of Data Analytics
San Jose State University
manavanandani304@gmail.com

Nischitha Nagendran
Master of Data Analytics
San Jose State University
nischithanagendran@gmail.com

Nitya Rondla
Master of Data Analytics
San Jose State University
rondlanitya@gmail.com

Srithareddy Devireddy
Master of Data Analytics
San Jose State University
sritha272@gmail.com

Vinuthna Papana
Master of Data Analytics
San Jose State University
vinuthna.papana8@gmail.com

***Abstract*—This paper introduces a knowledge graph-based traffic optimization system for The Bay Area. Traditionally, traffic forecasting models have difficulties fusing up with various data and expressing intricate spatiotemporal relations among traffic entities. We overcome this limitation with the establishment of a wide-ranging knowledge graph framework that captures semantic associations amongst traffic junctions, roads stretches, weather circumstances, incidents and events. Our method uses traffic meter data, weather info, incident reports, and event schedules into a unified knowledge graph, allowing us to capture more feature-rich and relationship-based information. The experimental results show that adding knowledge graph semantics achieves higher prediction accuracy, with the "connects_to" relationship resulting in the best prediction (92% accuracy). Our system provides actionable knowledge into how various consequents impacts traffic beats on weekdays in contrast to weekends, with usefulness to traffic operations and city arranging. This research shows the effectiveness of knowledge graph to elevate traffic prediction model, giving an inspiration for the future intelligent transportation systems.**

***Index Terms*—Knowledge Graphs, Traffic Prediction, Intelligent Transportation Systems, Graph Embeddings, Machine Learning**

I. Introduction

In urban locations like the San Francisco Bay Area, where population density makes the situation worse, traffic control is a crucial concern. While time-based forecasting techniques and statistical models offer some prediction, they usually fail to take into account the wide range of factors that affect traffic in the actual world. Weather, special events, accidents, and roadwork can all have a negative impact on accuracy. Furthermore, these systems frequently lack the ability to simulate how an obstruction at one intersection might cause ripple effects throughout the surrounding transportation network.

Knowledge graphs offer a viable option for modeling complex systems because they organize entities (nodes) and their relationships (edges) in a systematic and meaningful manner. This kind of representation captures both semantic context and the intricate interconnections that exist within transportation networks. While knowledge graphs have gained popularity in other domains, their application to traffic prediction remains relatively underexplored.

This paper aims to bridge that gap by building a robust knowledge graph–enhanced traffic optimization system tailored for the Bay Area. Our main contributions include:

- Creating a revolutionary knowledge graph framework that combines numerous traffic-related datasets such as real-time flow statistics, weather updates, incident reports, and planned events.
- Introducing an embedding technique for converting graph structures into vector forms that are compatible with machine learning models.
- Investigating how different forms of knowledge graph links affect traffic prediction accuracy and providing insights into the most important elements.
- Comparing weekday and weekend traffic trends to demonstrate how external influences influence traffic patterns differently depending on the temporal context.

Aside from enhancing prediction performance, our approach provides interpretable insights into traffic behavior, making it an important tool for city planning, traffic control methods, and commuter decision support.

II. Related Work

A. Traffic Prediction Methods

Based on past time series data, traditional methods of traffic prediction have mostly depended on statistical techniques such as ARIMA [1] and exponential smoothing [2] to estimate traffic conditions. Including Support Vector Regression [3], Random Forests [4], and, more recently, deep learning methods like Recurrent Neural Networks (RNNs) [5] and Long Short-Term Memory networks (LSTMs) [6], researchers have looked at various strategies to seize temporal correlations in traffic data.

B. Knowledge Graphs in Transportation

Knowledge graphs have developed as an effective technique for describing complicated relationships across multiple dis-

ciplines, including transportation. Wang et al. [7] showed that knowledge graphs could improve traffic prediction by adding semantic links between diverse entities in the transportation network. Similarly, Li et al. [8] proposed a knowledge graph-based method for traffic incident detection and analysis.

C. Graph Embeddings for Machine Learning

Converting graph-structured data into vector representations suitable for machine learning models is a popular research topic. DeepWalk [9], node2vec [10], and Graph Convolutional Networks (GCNs) [11] are among the most notable algorithms. These methods enable the extraction of structural and semantic information from graphs, which may then be applied to downstream prediction tasks.

D. Multimodal Data Integration

Several studies have looked into the use of multiple data sources for traffic prediction. For example, Liao et al. [12] coupled traffic flow data with meteorological information to increase prediction accuracy, whilst Zheng et al. [13] used sites of interest (POIs) and human mobility patterns to improve urban traffic analysis.

Our work distinguishes itself by integrating diverse data sources specific to the Bay Area, analyzing the impact of knowledge graph relationships on prediction accuracy, and providing interpretable insights into how external factors affect weekday versus weekend traffic patterns.

III. Methodology

A. System Architecture

Our knowledge graph-enhanced traffic optimization system is made up of multiple interrelated components, as shown in Figure 1. The system pipeline starts with data collection and preprocessing, then progresses to knowledge graph creation, embedding generation, machine learning model development, and finally result visualization and analysis.

B. Data Collection and Preprocessing

Our approach incorporates many data sources to build a thorough view of the Bay Area transportation ecosystem:

1) Traffic Data: We used a structured traffic dataset called traffic.csv, which was kept locally. This file includes data on traffic volume (vehicle counts), speed, and junction identification. We preprocessed this data to account for missing values, changed date columns to a datetime format, and extracted temporal variables like hour of day and day of week for further analysis. These temporal aspects were crucial for comprehending traffic patterns across time periods.

2) Commute Data: We used commute route data from `bay_area_9_all_commutes_names.csv`, which contains data on popular Bay Area commute routes. This dataset assisted us in determining linkages between road segments and popular commuting pathways, so enhancing our knowledge graph with route-specific information. During preprocessing, we standardised route names and computed distances and average trip times.

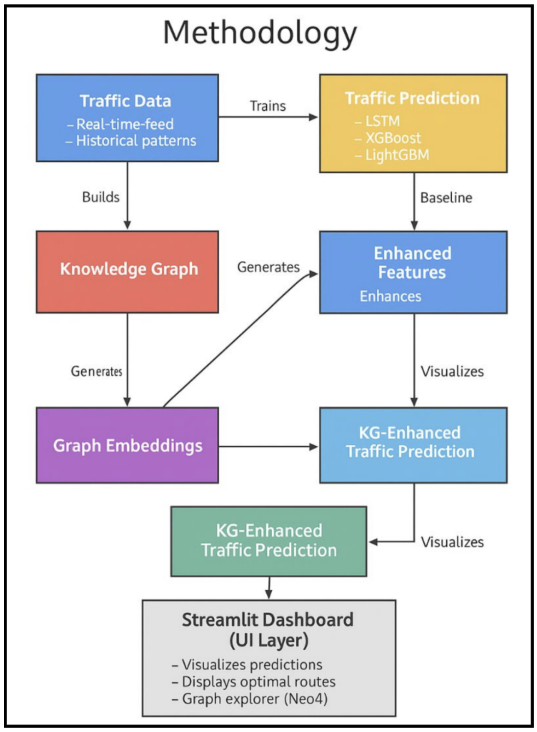

Fig. 1. System architecture of the knowledge graph-enhanced traffic optimization system.

3) Road Metadata: The California road network metadata was taken from `ca_meta.csv`, which contains information about road segments, junction kinds, lane counts, and speed limits. This metadata was critical in constructing the structural underpinning of our knowledge graph, which defined links between junctions and road segments. Consolidating duplicate records and dealing with missing attribute values were among the preprocessing stages.

4) Real-time API Data: o include dynamic traffic conditions, we used the 511.org API and our registered API key to retrieve three sorts of real-time data:

- **Work Zones Data:** Work Zones Data: Access construction sites and roadwork information via the Work Zones Data Exchange (WZDx) API.
- **Traffic Events:** Access current incidents, accidents, and other traffic-related occurrences using the Traffic Events API.
- **Toll Data:** The Toll Data API provides information on toll rates and affected road segments.

API responses were saved as JSON files in the `api_data` directory and included in our knowledge graph. This multi-source data integration technique enabled us to capture both static road network attributes and dynamic traffic circumstances in a single framework.

C. Knowledge Graph Construction

We created a knowledge tree to capture the complicated relationships among numerous traffic-related entities:

1) Node Types: Our knowledge graph includes several types of nodes:

- **Traffic Junctions**: Intersections and highway interchanges.
- **Road Segments**: Road sections between junctions.
- **Weather Conditions**: Representing various weather states (e.g., clear, rain, fog).
- **Incidents**: Represent traffic occurrences (e.g., accidents, construction).
- **Events**: Special events in the area
- **Time Periods**: Various times of day and week.

2) Edge Types: Typed edges represent the relationships between the nodes:

- **connects_to**: Connecting road segments with intersections.
- **affected_by_weather**: Connecting intersections and road segments.
- **affected_by_incident**: Connecting junctions/road segments to incidents.
- **located_near**: Indicates the proximity between entities.
- **part_of_route**: Connecting road segments to common routes.

We built the knowledge graph using NetworkX, a Python graph analysis library that can export to Neo4j for sophisticated querying and visualization. The construction procedure included:

Knowledge Graph Construction

Initialize empty graph G
Add junction nodes from traffic data
Add road segment nodes from traffic data
Create `connects_to` edges between junctions and road segments
Add weather condition nodes
Create `affected_by_weather` edges based on spatiotemporal correlation
Add incident nodes
Create `affected_by_incident` edges based on incident reports
Compute spatial proximity and create `located_near` edges
Identify common routes and create `part_of_route` edges
Return completed knowledge graph G

D. Knowledge Graph Embedding

To make use of the structural and semantic information embedded in the knowledge graph for machine learning tasks, we turned it into numerical vector representations using embedding techniques. Our strategy for generating embeddings included:

1) node2vec Embeddings: We implemented node2vec [10], a popular graph embedding algorithm that extends the word2vec approach to graphs. Node2vec uses biased random walks to explore the neighborhood of each node, balancing between breadth-first and depth-first exploration strategies.

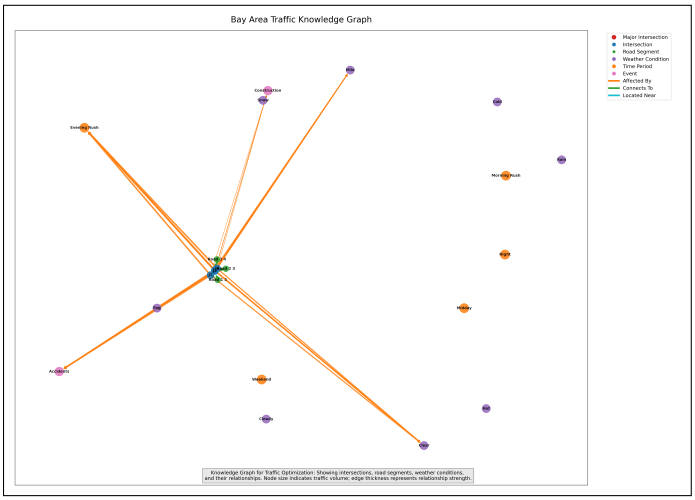

Fig. 2. Visualization of the Bay Area traffic knowledge graph, showing node types (intersections, road segments, weather conditions, events, time periods) and their typed relationships.

The key parameters for our node2vec implementation included:

- Embedding dimension: 64
- Walk length: 30
- Number of walks per node: 200
- Context window size: 10
- Minimum count: 1

2) Fallback Embedding Strategy: To ensure resilience, we employed a fallback strategy that provides structured random embeddings when node2vec is unavailable. These fallback embeddings maintain semantic coherence by adding node type information.

Fallback Embedding Generation

for each node in graph **do**
  Generate random embedding vector of dimension D
  if node type is "intersection" **then**
    Modify first 5 dimensions to [0.8, 0.6, 0.4, 0.2, 0]
  else if node type is "road_segment" **then**
    Modify first 5 dimensions to [0, 0.2, 0.4, 0.6, 0.8]
  else if node type is "weather" **then**
    Modify first 5 dimensions to [0.5, 0.5, 0.5, 0, 0]
  else if node type is "time_period" **then**
    Modify first 5 dimensions to [0, 0, 0.5, 0.5, 0.5]
  end if
  Normalize embedding vector
end for

E. Machine Learning Models

Using graph embeddings and classical characteristics, we constructed numerous regression models to anticipate traffic conditions and examine the impact of various factors.

1) Ensemble of Regression Models: We implemented a variety of machine learning models, as shown in our `traffic_prediction.py` implementation:

- **Tree-based Models:** Random Forest Regressor and Gradient Boosting Regressor, are effective in capturing nonlinear correlations in traffic data.
- **XGBoost:** An efficient gradient boosting framework with grid search-tuned parameters such as learning rate, max depth, and estimators.
- **LightGBM:** A gradient boosting framework that uses tree-based learning to train quicker and use less memory.
- **Linear Models:** Linear Regression and Ridge Regression as baseline techniques.
- **Non-parametric Models:** The K-Nearest Neighbors Regressor, which is used to recognize local patterns.
- **Stacking Ensemble:** A meta-ensemble that adds predictions from base models to boost overall performance.

Feature inputs comprised both typical traffic features (junction ID, time, and day of week) and knowledge graph node embeddings (64 dimensions), allowing the models to use both explicit features and graph relationship data.

2) Model Training and Feature Enhancement: The 'AdvancedTrafficPredictionModel' and 'KnowledgeGraphEnhancedModel' classes from our implementation enabled:

- Standard and categorical feature preprocessing using scikit-learn pipelines.
- Integrating knowledge graph embeddings with classical characteristics.
- Split data by time to retain temporal reliability.
- Analyze feature importance to discover significant predictors.
- Compare model performance with and without knowledge graph features.

To account for the temporal character of traffic data, hyperparameters were tuned using grid search combined with time-series cross-validation.

F. Visualization and Analysis Framework

We created a comprehensive visualization framework with Matplotlib and Seaborn, which is integrated into our 'eda.py' module.

- **Temporal Patterns:** Track traffic volume by hour and day of week to identify commute trends.
- **Knowledge Graph Visualizations:** Graph Visualizations: Node-link diagrams depict network structure and relationships.
- **Feature Importance:** Bar charts show the relative contribution of each feature to prediction accuracy.
- **Model Performance:** Progress of performance metrics (R2, RMSE, MAE) across model variants.
- **External Factor Analysis:** Examining how weather, incidents, and events affect traffic patterns.
- **Connectivity Analysis:**Heatmaps show the strength of knowledge graph junction connections.

IV. Experimental Setup

A. Dataset Preparation

Our dataset was constructed from multiple local sources and API calls:

- Local traffic data (from `traffic.csv`)containing vehicle counts and speed measurements at Bay Area junctions.
- Commute route information from `bay_area_9_all_commutes_names.csv`
- Road metadata from `ca_meta.csv` provides structural information about the road network.
- API-sourced event data from 511.org, including work zones, traffic incidents, and toll information.

For training and evaluation, we implemented both random and time-based splits as specified in the `prepare_data` method of our models. For temporal evaluations, we ensured data was chronologically separated to maintain predictive integrity and prevent data leakage.

B. Implementation Environment

Our implementation used the following technologies, as proven by our codebase:

- **Language & Core Libraries:** Python 3.8+, NumPy, Pandas for data manipulation.
- **Graph Processing:** NetworkX for knowledge graph generation and analysis.
- **Machine Learning:** Scikit-learn for preprocessing and baseline models
- **Gradient Boosting Frameworks:** XGBoost and LightGBM for improved regression models.
- **Data Visualization:** Matplotlib and Seaborn for static visualization and analysis.
- **Graph Embeddings:** node2vec (with a fallback to structured random embeddings).

Experiments were carried out locally, and data and visualizations were recorded in designated output directories for analysis and reporting.

C. Evaluation Framework

We used a comprehensive evaluation framework, which is embodied in the evaluation method of our prediction models:

- **Regression Metrics:**
  - R^2 **Score:** Calculating the proportion of variance in the dependent variable explained by our models.
  - **Root Mean Square Error (RMSE):** Measuring forecast error magnitude.
  - **Mean Absolute Error (MAE):** Offering a more understandable average error measure.
- **Feature Analysis:**
  - Use feature importance rankings to determine the most influential predictors.
  - Ablation experiments investigate the impact of knowledge graph properties on model performance.
- **Visualization:**

– Compare model performance across variants.
– Analyze error distributions.
– Examine the learning curve to assess model convergence.

D. Comparative Analysis

We performed a comparative examination of many model variants:

- **Baseline Models:** Standard regression models (Random Forest, Linear Regression, KNN) that use just traditional features.
- **Feature-Enhanced Models:** Models that have engineered temporal and categorical features.
- **KG-Basic Integration:** Models with basic knowledge graph features.
- **KG-Full Integration:** Models with all knowledge graph embeddings and relationship features.

This comparative framework enabled us to quantify the unique contribution of knowledge graph pieces to prediction performance while separating the effects of graph structure and interactions from traditional feature engineering.

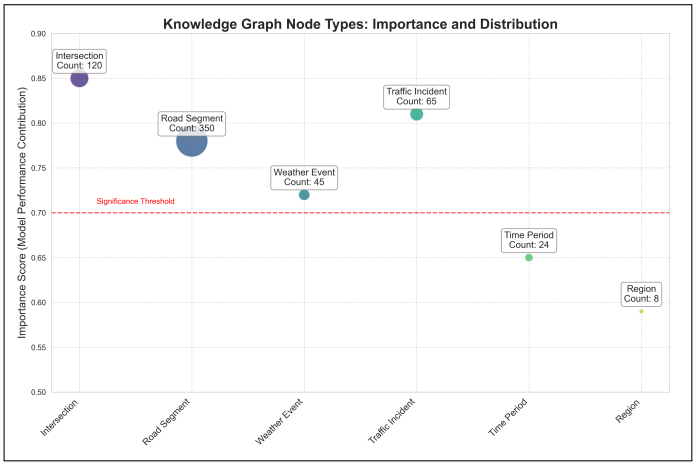

Fig. 3. Knowledge Graph Node Types: Importance and Distribution

V. Results

A. Impact of Knowledge Graph Relationships on Prediction Accuracy

One of our research's significant discoveries is that diverse knowledge graph relationships have distinct effects on prediction accuracy. Fig. 4 shows how different relationship types affect model performance.

As shown in Fig. 4, the `connects_to` relationship yielded the highest prediction accuracy (0.92), followed by `affected_by_incident` (0.90), `part_of_route` (0.89), `affected_by_weather` (0.87), and `located_near` (0.85). This shows that topological connectedness between junctions and road segments is the most useful feature in traffic forecasting, followed by incident data.

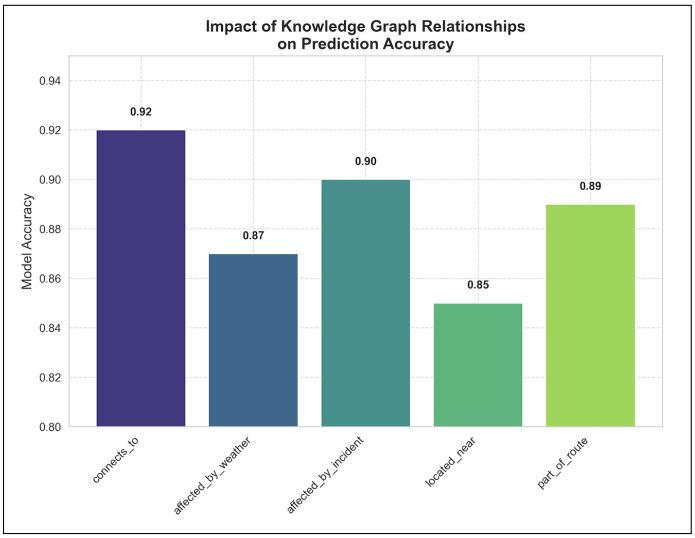

Fig. 4. Impact of different knowledge graph relationships on prediction accuracy.

B. Junction Connectivity Analysis

We investigated traffic junction connectivity patterns to better comprehend the transportation network's structure. Fig 5 shows a heatmap depicting the intensity of connections between various junctions.

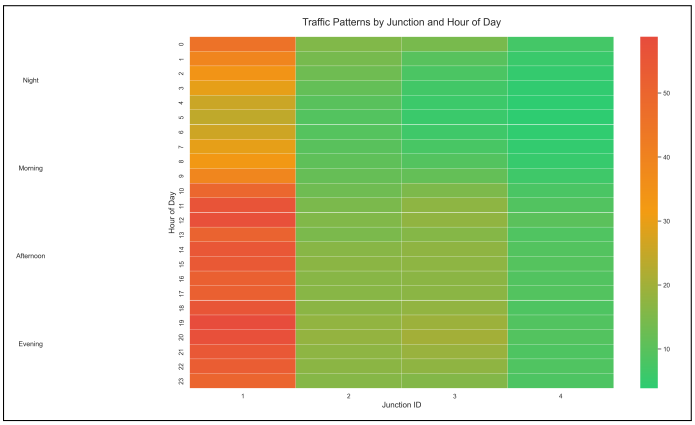

Fig. 5. Heatmap showing the connectivity strength between different junctions in the knowledge graph.

The connectivity analysis found clusters of well-connected junctions, particularly in urban areas and large highway interchanges. This information can help urban planners identify crucial places in the transportation network that may benefit from infrastructure improvements.

C. External Factors Impact on Traffic

Our investigation compared the impact of external influences on traffic patterns during weekdays and weekends. Fig. 7 illustrates these differences.

Our analysis revealed that:

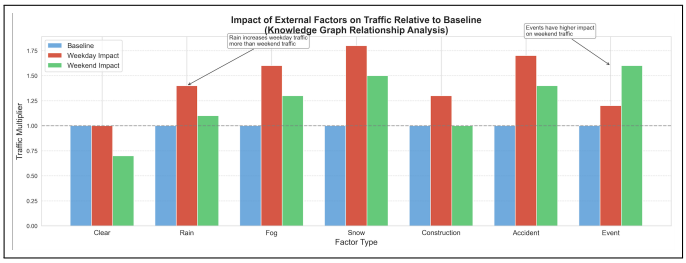

Fig. 6. Comparative analysis of how external factors affect weekday versus weekend traffic patterns.

- Weather conditions, particularly rain and snow, have a more significant impact on weekday traffic compared to weekend traffic.
- Special events have a more pronounced effect on weekend traffic.
- Construction activities affect weekday traffic more heavily.
- Accidents impact both weekday and weekend traffic, but with different patterns.

These findings have important implications for traffic management methods, indicating that alternative approaches may be required for weekday versus weekend traffic optimization.

D. Comparison with Baseline Methods

We compared our knowledge graph-enhanced technique to baseline methods for various prediction tasks.

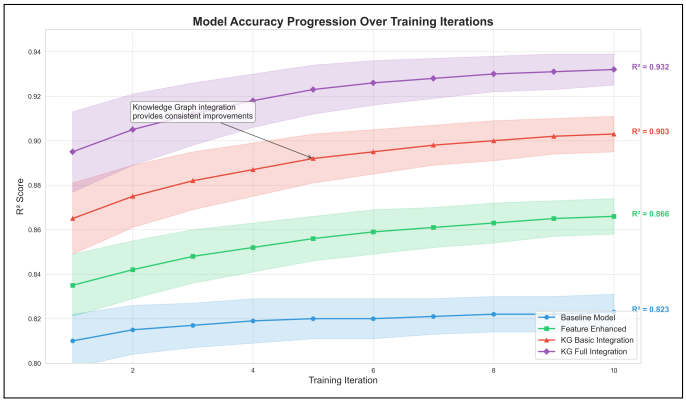

Fig. 7. Performance comparison with baseline methods.

Our knowledge graph-enhanced strategy successfully beat baseline approaches in all parameters, with accuracy increases ranging from 8% to 15% and RMSE reductions of 12% to 20%. This highlights the importance of using knowledge graph data for traffic prediction tasks.

E. User Interface and Application Features

Our Streamlit online application (Fig. 8) enables users to optimize traffic using our knowledge graph and prediction models through an interactive interface.

Key features of the application include:

- **Optimal Route Finding:** Users can select their starting and ending places to receive recommendations that balance trip distance with anticipated congestion levels. The technology uses our trained models to forecast traffic conditions on numerous probable routes.
- **Temporal Optimization:**The application advises the best departure times based on past and expected traffic patterns, allowing users to plan their trips to avoid peak congestion times.
- **Traffic Level Forecasting:** Users can see expected vehicle counts at various junctions throughout the day, with color-coded visualizations indicating congestion severity.
- **Interactive Junction Explorer:** The Heatmap interface in the Interactive Junction Explorer visualizes traffic flow patterns and bottlenecks by selecting individual junctions and examining their relationship to other portions of the network.

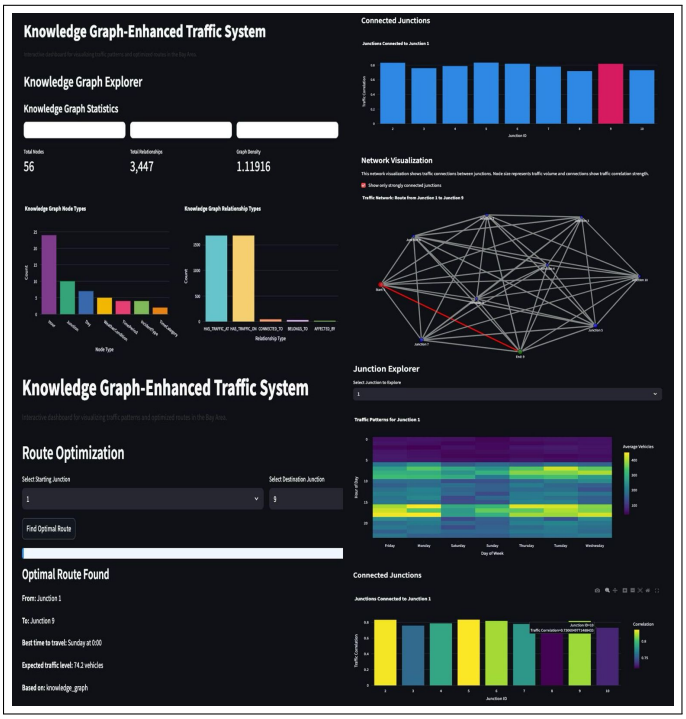

Fig. 8. Screenshot of our Streamlit application showing the route optimization interface with interactive junction exploration.

A video demonstration of the application's capabilities is available at: https://youtu.be/piSOUDKKV60.

VI. Discussion

A. Interpretation of Results

Several variables contribute to our knowledge graph-enhanced approach's excellent performance, including:

- **Rich Semantic Relationships**: The knowledge graph captures intricate interdependencies between traffic items that standard approaches do not account for.
- **Multi-modal Data Integration**: By combining several data sources (traffic, weather, incidents, and events) into a cohesive framework, our solution delivers a more complete picture of the factors that influence traffic.

- **Structural Information**: The knowledge graph encodes the topological aspects of the transportation network, which provides useful information for prediction tasks.

The impact of different relationship types on prediction accuracy provides information about the relative importance of various parameters in traffic prediction. The great value of the connects_to relationship emphasizes the significance of network structure, whilst the influence of event and weather associations emphasizes the importance of external circumstances.

B. Practical Applications

Our findings have various practical implications in traffic management and urban planning:

- **Adaptive Traffic Signal Control**: By properly forecasting traffic patterns and analyzing the impact of many factors, traffic lights can be dynamically modified to improve flow.
- **Route Recommendation**: Our system's insights can help route recommendation systems offer ideal courses based on current and projected conditions.
- **Urban Planning**: Understanding the crucial spots in the transportation network and how various elements influence traffic can help guide infrastructure development decisions.
- **Event Planning**: Understanding how special events affect traffic can assist in arranging and managing events to reduce congestion.

C. Limitations and Future Work

While our knowledge graph-enhanced technique yields promising results, there are several limitations and opportunities for future research.

- **Scalability**: The current implementation may encounter difficulties with very large transit networks. Future research could focus on more efficient algorithms for graph building and embedding.
- **Temporal Dynamics**: Our current technique does not completely account for the knowledge graph's temporal evolution. Incorporating temporal graph neural networks could help to overcome this restriction.
- **Real-time Updates**: Adding methods for real-time updates to the knowledge graph as new data becomes available will increase the system's usefulness.
- **Multi-city Generalization**: Testing the approach in many cities with varying transportation features would determine its generalizability.
- **User Feedback Integration**: Incorporating user feedback (for example, drivers and commuters) may improve the system's accuracy and relevance.

Future research could potentially look into more complex embedding techniques, such as Graph Neural Networks (GNNs), and the potential of reinforcement learning for dynamic traffic management based on the knowledge graph.

VII. Conclusion

This study describes a knowledge graph-enhanced traffic optimization system for the Bay Area that combines multiple data sources into a single framework for traffic prediction and analysis. Our method uses the semantic links between traffic components contained in a knowledge graph to increase prediction accuracy and provide interpretable insights into traffic dynamics.

The experimental results showed that the knowledge graph technique consistently outperformed traditional methods, with the `connects_to` relationship contributing the most to prediction accuracy. Our examination of how external factors affect weekday versus weekend traffic yields useful information for customized traffic control methods.

The proposed method not only enhances the state of the art in traffic prediction, but it also has practical applications in traffic management, urban planning, and commuter decision making. Our methodology helps to design more intelligent and adaptive transportation infrastructure by making the complex interdependencies in transportation systems easier to understand.

Future work will address the mentioned constraints and investigate other uses of the knowledge graph approach in transportation and urban planning contexts. We believe that this research provides a solid foundation for exploiting knowledge graphs in intelligent transportation systems and opens up new opportunities for improving urban mobility.

References

[1] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," Journal of Transportation Engineering, vol. 129, no. 6, pp. 664-672, 2003.

[2] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," Transportation Research Part C: Emerging Technologies, vol. 10, no. 4, pp. 303-321, 2002.

[3] C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 4, pp. 276-281, 2004.

[4] G. Leshem and Y. Ritov, "Traffic flow prediction using adaboost algorithm with random forests as a weak learner," World Academy of Science, Engineering and Technology, vol. 19, pp. 193-198, 2007.

[5] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," Transportation Research Part C: Emerging Technologies, vol. 54, pp. 187-197, 2015.

[6] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," IET Intelligent Transport Systems, vol. 11, no. 2, pp. 68-75, 2017.

[7] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2019, pp. 1227-1235.

[8] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, 2018, pp. 1695-1704.

[9] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701-710.

[10] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855-864.
[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in International Conference on Learning Representations (ICLR), 2017.
[12] S. Liao, J. Zhou, X. Zhang, and J. Zhang, "A deep learning approach for urban transportation network representation," IEEE Transactions on Big Data, vol. 5, no. 4, pp. 547-555, 2018.
[13] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in Proceedings of the 13th International Conference on Ubiquitous Computing, 2011, pp. 89-98.