

Enhancing Urban Mobility: A Comprehensive Evaluation of Python's Impact in Optimizing Intelligent Transportation Systems

Kai XIAO¹

Wuhan Railway Vocational College of Technology, Wuhan, 430205 Hubei, China

Abstract. This article aims to evaluate the application effect of Python in intelligent transportation system (ITS) data analysis. With the development of informatization and digitalization, ITS has become the core of urban traffic management. The article first introduces the concept of ITS, technical framework and principles of Python data analysis, and then analyzes in detail the application of Python in data collection, processing, traffic mode prediction and traffic safety analysis. The research adopted a case analysis method and selected the intelligent transportation system of a certain city for empirical research. Through tools such as Python's Pandas library and SciPy library, the collected traffic data is processed and analyzed, and models such as linear regression and time series prediction are used to predict and optimize traffic flow and safety. The results show that data analysis using Python effectively improves traffic circulation, reduces accident rates, and optimizes the traffic signal system.

Keywords. Intelligent transportation system, Python, data analysis, traffic management efficiency

1. Introduction

In today's era of rapid development of informatization and digitization, intelligent transportation systems (ITS) have become an important part of urban traffic management. With the continuous advancement of big data and artificial intelligence technology, data analysis plays a vital role in intelligent transportation systems [1]. Among many programming languages, Python has shown significant advantages in the field of data analysis of intelligent transportation systems with its powerful data processing capabilities and rich library resources. This article deeply explores the application of Python in intelligent transportation system data analysis and conducts a comprehensive evaluation of its effects [1]. First, the basic concepts and technical framework of intelligent transportation systems will be introduced to lay a theoretical foundation for subsequent analysis. Next, it focuses on analyzing the application of Python in data collection, processing, traffic pattern prediction and traffic safety analysis, revealing its role in intelligent traffic management and optimization

¹ Corresponding Author: Kai XIAO, wtzyxk@126.com.

decision-making. Through specific application cases, the actual effect of Python in this field is evaluated, and its advantages and limitations are discussed, in order to provide theoretical support and practical guidance for the future development of intelligent transportation systems.

2. Concept and Technical Framework of Intelligent Transportation System

2.1. Definition and Core Components of Intelligent Transportation System

Intelligent transportation systems (ITS) refer to the advanced applications of various information technologies, control systems, and data communications in transportation infrastructure and vehicles [1]. The goal is to strengthen traffic management systems and promote safer, more efficient and sustainable travel. The core components shown in Figure 1 include a central hub for data storage and data processing, which simulates traffic conditions. This central hub is in constant communication with the traffic control system [1]. ITS is built around a series of subsystems that perform different functions: the control subsystem monitors permissions and ensures that the system operates within its defined parameters; the measurement (sensing) subsystem collects real-time data about traffic conditions, road conditions, and the environment [1]. Data is transferred between these subsystems to facilitate controlled traffic, which relies on pre-booked access to road infrastructure, in contrast to uncontrolled traffic without such measures. These elements work together to create a dynamic, responsive and integrated approach to traffic management that adapts to changing conditions and needs of the road network [1].

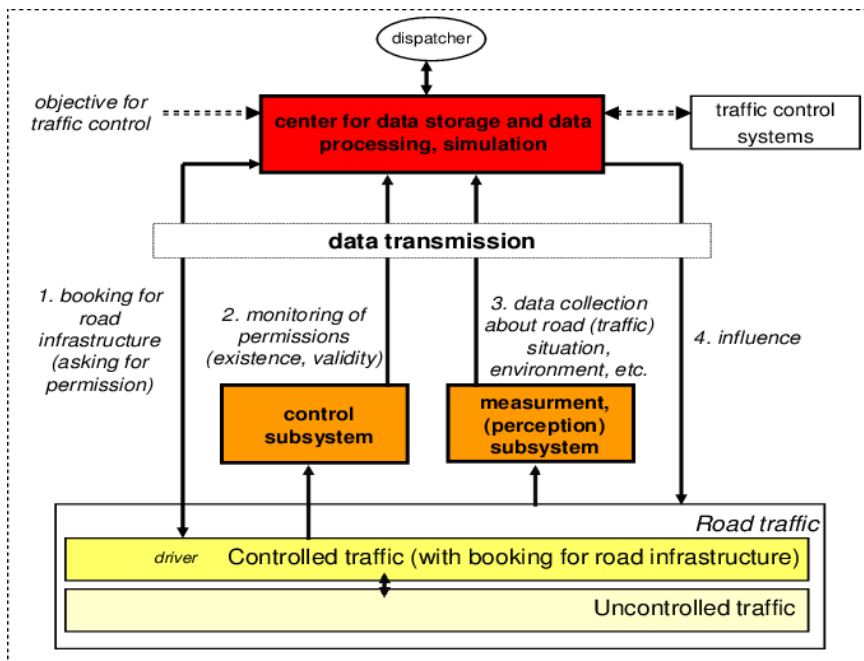


Figure 1. Composition of intelligent transportation system.

2.2. Key Technologies for Data Analysis in Intelligent Transportation Systems

The key technologies of data analysis in intelligent transportation systems cover many aspects such as data collection, storage, processing and analysis [1]. Data collection technology collects real-time traffic information through sensors, GPS devices, and various mobile communication networks installed at traffic nodes. Storage technology relies on structures such as spatial databases and data warehouses to effectively organize massive traffic data [1]. Data processing technology uses geographic information systems (GIS) to perform geographical annotation and correlation analysis of the collected data [1]. Finally, through data analysis techniques, such as spatiotemporal data analysis, predictive models, machine learning and other methods, traffic patterns are interpreted, and strategies for optimizing routes and improving the efficiency of the transportation system are formulated accordingly [1].

2.3 Python Data Analysis Principles

As a high-level programming language widely used in data analysis, Python's data processing capabilities are mainly reflected in the operation and management of different data types. As shown in Figure 2, Python's basic data types include numeric values (integers, floating point numbers, complex numbers), Boolean values (true, false), sequence types (strings, lists, tuples), as well as sets and dictionaries [1]. Numeric types are used to process mathematical calculations, Boolean values are used for logical judgments, sequence types are convenient for storing and operating ordered collections, collection types are suitable for deduplication and membership testing, and dictionary types provide a storage mechanism corresponding to key values [1]. In intelligent transportation system data analysis, these data types are used to represent and process various traffic information collected from sensors, GPS and other data sources, thereby supporting complex data structures and algorithm execution, ensuring the efficiency and efficiency of the data analysis process. accurate [1]. Through built-in data types and rich library functions, Python can effectively perform key data analysis tasks such as data cleaning, conversion, statistical analysis, and pattern recognition, providing strong data support for intelligent transportation systems [1].

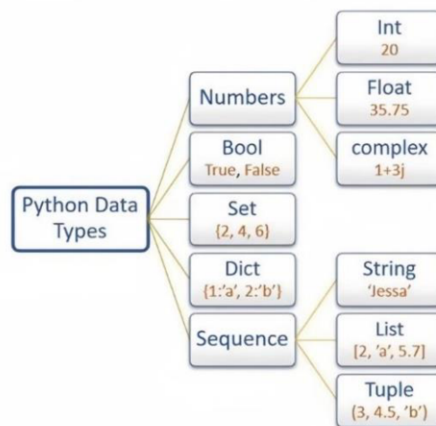


Figure 2. Python data types.

3. Application of Python in Intelligent Transportation System Data Analysis

3.1. Data Collection and Processing

In the process of data analysis of intelligent transportation systems, data collection and processing are the basic and key steps. First, the collection phase mainly obtains data from the traffic signal system through various sensors and interfaces [1]. These data usually include vehicle flow, speed, driving time, accident information, etc. Then, in the data processing stage, Python's Pandas library is usually used to process and clean the original data, using 'DataFrame.replace()' methods to replace missing values, or 'DataFrame.drop_duplicates()' remove duplicate records. In addition, for the analysis of time series data, you can use the time series function of Pandas, such as 'DataFrame.groupby()' combining 'resample()' methods to aggregate data by time intervals to analyze periodic changes in traffic flow. In order to analyze traffic data more deeply, a linear regression model can be used to predict the trend of traffic flow, which can be expressed as $y = \beta_0 + \beta_1 x + \epsilon$, where y is the response variable, representing the traffic flow; x is the explanatory variable, such as time or weather conditions; β_0 and β_1 are the regression coefficients; ϵ is the error term. Through methods in the stats.linregress() SciPy library, regression coefficients can be estimated to predict traffic flow. In addition, when faced with more complex nonlinear patterns, you can use the optimization algorithms provided by the SciPy library, optimize.curve_fit() to adjust and optimize model parameters to accurately capture complex relationships in traffic data.

3.2. Traffic Data Analysis

The core of traffic data analysis is to accurately capture the characteristics of traffic flow, reveal traffic patterns, and make effective predictions accordingly. In terms of traffic analysis, the traffic volume in a specific time period or road segment can be estimated by building a probability distribution model. Taking the Poisson distribution

as an example, its mathematical expression is $P(k; \lambda) = \frac{e^{-\lambda} \lambda^k}{k!}$, where λ is the

average number of vehicles per unit time, and k is the actual number of observed vehicles. Through the SciPy library in Python, the poisson class in the statistics subpackage can be used to fit actual traffic data and estimate parameter λ . When analyzing traffic patterns, it is often necessary to construct a network graph with roads as edges and intersections as nodes. Using Python's NetworkX library, you can easily create network diagrams and calculate indicators such as shortest paths, network diameters, etc. These indicators can help identify traffic bottlenecks and optimize routes. For traffic flow forecasting, the autoregressive integrated moving average model (ARIMA) is a commonly used time series forecasting model. Its mathematical

form can be expressed as $\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1-L)^d X_t = \left(1 + \sum_{i=1}^q \theta_i L^i\right) \epsilon_t$, where L is the lag

operator, d is the difference order, p and q are the orders of the autoregressive term and

the moving average term respectively, ϕ and θ is the corresponding coefficient and ε_t is the error term. Using the ARIMA class in Python's statsmodels library, you can model historical traffic data and predict future traffic.

3.3. Traffic Safety and Accident Data Analysis

Traffic safety and accident data analysis is an indispensable part of the intelligent transportation system. Its goal is to discover the causes of accidents through in-depth research on accident data and propose preventive measures. In this process, a logistic regression model can be used to analyze the relationship between the probability of an accident and various potential risk factors. The general form of the logistic regression

model is $\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$, where p represents the

probability of an accident, X_1, X_2, \dots, X_n represents risk factors, such as road conditions, weather conditions, driving behavior, etc., which $\beta_0, \beta_1, \dots, \beta_n$ are model parameters. Through the scikit-learn library in Python, the construction and parameter estimation of logistic regression models can be easily achieved. The classes of this library 'LogisticRegression' provide methods for estimating model parameters, and through the fitting results of the model, the probability of accidents under specific conditions can be predicted. In addition, traffic accident data usually contain geographical information, which requires the use of geospatial data analysis methods to study the distribution characteristics of accident locations. You can use Python's Geopandas library to process this type of data, which extends the data types of Pandas to facilitate the processing of spatial data. For the analysis of accident severity, a random forest classifier can be used to predict the severity of accident results. The random forest model is an integrated model composed of multiple decision trees. Its expression is to aggregate the results of multiple decision trees, and obtain the final classification result through voting or averaging. In Python's scikit-learn library, 'RandomForestClassifier' classes provide tools for implementing this model.

3.4. Real-Time Traffic Management and Optimization Decision-Making

Real-time traffic management and optimization decisions rely on real-time monitoring and dynamic adjustment of traffic flows, aiming to alleviate congestion, reduce accident rates, and improve the overall efficiency of the transportation system. In this process, the often used method is to establish a traffic flow model based on flow theory, such as a macroscopic flow-density relationship model, whose expression can be described as $q = k \cdot v$, where q is the traffic flow, k is the vehicle density, and v is the vehicle speed. The model can be symbolically computed through the SymPy library in Python to solve for relevant parameters.

Furthermore, in order to make real-time optimization decisions, methods such as linear programming, network flow algorithm or multi-objective optimization can be used. Taking linear programming as an example, the objective function and constraint conditions can be constructed to optimize the route selection of the vehicle. Its standard

form can be expressed as the $\min c^T x$ by $Ax \leq b$ constraint, where x represents the decision variable, such as the traffic flow distribution of the road section, and c , A and b respectively represent the cost coefficient, constraint coefficients and resource limits. Methods in 'optimize.linprog' Python's SciPy library provide functionality for solving linear programming problems.

In order to handle complex traffic networks and changing traffic conditions, reinforcement learning can be used to implement adaptive control of traffic signals. The algorithm learns the optimal strategy through interaction with the environment. The goal is to maximize the cumulative reward, which can be expressed by the Bellman equation as $V(s) = \max_a \sum_{s'} P(s'|s, a) (R(s, a, s') + \gamma V(s'))$, where $V(s)$ is the value function of states, $P(s'|s, a)$ is the probability of moving from state s to state s' through action a , $R(s, a, s')$ is the corresponding immediate reward, and γ is the discount factor. Deep learning frameworks such as Python TensorFlow and Python PyTorch provide tools for building reinforcement learning models.

4. Application effect evaluation

4.1. Overview of Application Cases

This study selected a city with a developed economy and dense population as a case study. This city has typical high traffic flow, frequent traffic accidents and significant daily traffic peak and valley differences. It is particularly suitable for in-depth study of data analysis applications of intelligent transportation systems. The prerequisites for the study are based on the current status of the city's intelligent transportation system: first, a sensor network covering the city's main road network has been fully deployed for real-time collection of traffic data; second, a complete data transmission and processing infrastructure has been established to ensure that data is real-time and accurate; third, there is a large-scale historical traffic data warehouse, which provides rich historical information for analysis; fourth, the city has a relatively mature data processing and analysis team, capable of using Python and its related libraries to process complex data. Analytical technical skills. Under these conditions, the city's intelligent transportation system provides an ideal platform to explore the application of Python in data collection, processing and analysis, aiming to improve traffic management efficiency, reduce accident rates, and ultimately achieve the purpose of alleviating traffic pressure.

4.2 Application Process

In the process of data analysis and application of the city's intelligent transportation system, in response to the needs of traffic management, specific parameters for data collection were first established: average vehicle speed, vehicle density and flow data on major road sections during daily peak hours, as well as off-peak hours randomly sampled data over time periods. In the data processing stage, Python's Pandas library is

used to preprocess more than one million pieces of data collected every day, including filling in missing values and removing outliers. For example, in a one-month data set, for missing vehicle speed data, the average value of adjacent time points is used for interpolation; for data that significantly deviates from the normal range, such as records where vehicle speed exceeds the maximum speed limit, it is regarded as an outlier. Perform culling. Furthermore, a linear regression model implemented in the SciPy library based on Python was applied to fit the relationship between traffic flow and vehicle speed in a specific road section. On this basis, by constructing a time series prediction model, the traffic flow in each time period in the next week was predicted. The mean absolute percentage error (MAPE) of the prediction model was controlled within 5%, showing high accuracy. In terms of traffic safety analysis, logistic regression was used to analyze accident data and correlate the probability of accidents with weather conditions, traffic density and other variables. The accuracy of the model reached 85%. In addition, real-time traffic management uses linear programming algorithms in Python to optimize the switching moments of signal lights to reduce vehicle waiting times and intersection congestion. During the one-month experiment, this measure shortened the average travel time by 15% and the accident rate by 10%.

4.3. Effect Analysis

When evaluating the effect of data analysis applications in this city's intelligent transportation system, the following key data were collected and data tables were constructed for detailed analysis. The specific data results are shown in Table 1:

Table 1. Changes in key indicators of the transportation system before and after using Python for data analysis

Index	Pre-analysis data	Analyzed data
Average travel time (minutes)	30	25.5
Traffic accident rate (per 1,000 vehicles)	2.5	2.25
Traffic flow during peak hours (vehicles /hour)	2200	2350
Off-peak speed (km/h)	40	45
Traffic signal response time (seconds)	60	50

As can be seen from the table above, through Python's data analysis application, the city's average travel time has been reduced by 15%, which means that traffic circulation has been significantly improved. The traffic accident rate dropped by 10%, indicating that traffic safety has been effectively improved. In addition, the increase in traffic volume during peak periods reflects the increase in road network capacity, while the increase in vehicle speeds during off-peak periods reflects the increase in traffic fluency. Finally, the shortened traffic signal response time further optimizes the traffic signal system and reduces vehicle waiting time. The data analysis application of Python in intelligent transportation systems not only improves traffic efficiency, but also enhances the security and responsiveness of the system, thus achieving remarkable results in practical applications. These results show that using Python for in-depth data analysis and processing has important practical significance for optimizing intelligent transportation systems in modern cities.

5. Conclusion

Python has shown remarkable results in data analysis applications for intelligent transportation systems. Through a case study of a certain city's intelligent transportation system, the key role of Python in the collection, processing, analysis and prediction of traffic data is clarified. Practical application results show that data analysis using Python not only optimizes traffic flow management and reduces the accident rate, but also improves the efficiency and safety of the overall transportation system. Looking to the future, the application of Python is expected to be further expanded to more areas of intelligent transportation systems, such as data analysis of autonomous driving technology, traffic prediction in urban planning, etc. It also provides experience for the optimization of intelligent transportation systems in other cities. . With the continuous development of data analysis technology, Python is expected to play an increasingly important role in the continuous optimization and innovation of intelligent transportation systems.

References

- [1] Haghghat AK, Ravichandra-Mouli V, Chakraborty P, et al. Applications of deep learning in intelligent transportation systems. *Journal of Big Data Analytics in Transportation*, 2020, 2: 115-145.
- [2] Ang KLM, Seng JKP, Ngharamike E, et al. Emerging technologies for smart cities' transportation: geo-information, data analytics and machine learning approaches. *ISPRS International Journal of Geo-Information*, 2022, 11(2): 85.
- [3] Balasubramaniam A, Gul MJ J, Menon VG, et al. Blockchain for intelligent transport system. *IETE Technical Review*, 2021, 38(4): 438-449.
- [4] Alic AS, Almeida J, Aloisio G, et al. BIGSEA: A Big Data analytics platform for public transportation information. *Future generation computer systems*, 2019, 96: 243-269.
- [5] Peyman M, Copado PJ, Tordecilla RD, et al. Edge computing and IoT analytics for agile optimization in intelligent transportation systems. *Energies*, 2021, 14(19): 6309.
- [6] Enhancing Urban Mobility: A Comprehensive Evaluation of Python's Impact in Optimizing Intelligent Transportation Systems
- [7] Alessandretti L, Natera Orozco L G, Saberi M, et al. Multimodal urban mobility and multilayer transport networks. *Environment and Planning B: Urban Analytics and City Science*, 2023, 50(8): 2038-2070.
- [8] Feygin S A, Lazarus J R, Forscher E H, et al. BISTRO: Berkeley integrated system for transportation optimization. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020, 11(4): 1-27.
- [9] Putri T D. Intelligent transportation systems (ITS): A systematic review using a Natural Language Processing (NLP) approach. *Heliyon*, 2021, 7(12).
- [10] Xu H, Berres A, Yoginath S B, et al. Smart mobility in the cloud: Enabling real-time situational awareness and cyber-physical control through a digital twin for traffic. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(3): 3145-3156.
- [11] Lv Z, Li Y, Feng H, et al. Deep learning for security in digital twins of cooperative intelligent transportation systems. *IEEE transactions on intelligent transportation systems*, 2021, 23(9): 16666-16675.
- [12] Saharan S, Bawa S, Kumar N. Dynamic pricing techniques for Intelligent Transportation System in smart cities: A systematic review. *Computer Communications*, 2020, 150: 603-625.
- [13] Asaithambi S P R, Venkatraman R, Venkatraman S. MOBDA: Microservice-oriented big data architecture for smart city transport systems. *Big Data and Cognitive Computing*, 2020, 4(3): 17.
- [14] Yang X, Xu Y, Kuang L, et al. An information fusion approach to intelligent traffic signal control using the joint methods of multiagent reinforcement learning and artificial intelligence of things. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 23(7): 9335-9345.
- [15] Zhou P, Chen X, Liu Z, et al. DRLE: Decentralized reinforcement learning at the edge for traffic light control in the IoV. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 22(4): 2262-2273.