QWDAP: The R Package of Quantum Walk-Based Data Analysis and Prediction

Abstract

In this paper, the R package **QWDAP**, which implements quantum walk as the fundamental infrastructure, is developed for spatio-temporal data modeling and prediction. In the **QWDAP** package, we mainly focus on the analysis of spatially correlated time series, and use the variable series generated by quantum walks to model and predict the time series. With the integration of a series of different mode selection, modeling, prediction and model evaluation methods, the **QWDAP** package realizes the coupled analysis of the significant modes, temporal and spatial correlations and evolution laws in the time series. The **QWDAP** package includes three modules: Basis Generation, Data Modeling and Prediction, and Model Evaluation. Without any priori assumptions such as stationarity, linearity and independence of the time series, the **QWDAP** package can extract significant modes from different perspectives, develop linear regression, nonlinear regression and time-based relationships between modes and the original time series, and predict the time series with or without considering the spatio-temporal correlations. A case study, which models and predicts the traffic volumes of highway traffic system, is used to demonstrate the structure and usage of the package.

Keywords: Quantum Walks, Modeling, Prediction, Mode Selection, Model Evaluation, Multi-time series.

1. Introduction

Graph-associated time series is an expansion of the traditional time series by considering the

spatial relative relationships of the time series to form a combination of time series containing graph definitions. While traditional time series analysis methods have difficulties in expressing the spatial relationships between time series. Graph-associated analysis methods, such as graph convolution (Kejani, Dornaika, and Talebi 2020), which can analyze the structural characteristics of graph, but the graph convolution is based on the calculation of the original data, whose operation results will be limited to the original data, and the method for prediction lacks the stochasticity. In this paper, we propose a feature extraction and time series prediction method based on quantum walk for spatio-temporal characteristics analysis of graph-associated time series.

Quantum walk, the quantum mechanical counterpart of classical random walk, has been recently shown great potential to constitute quantum computing models for rapid data operation and simulation. Quantum walk, which is one of the main algorithms of quantum computer technology (Childs 2010), constituting the general model of quantum computing (Venegas-Andraca 2012), has the potential to simulate and explain quantum systems (Schreiber, Gábris, Rohde, Laiho, Štefaňák, Potoček, Hamilton, Jex, and Silberhorn 2012). With the advantages of complete mathematical foundations, simple and clear physical mechanism, and efficient implementation in computation, quantum walk has been widely used in quantum simulation (Karski, Förster, Choi, Steffen, Alt, Meschede, and Widera 2009; Tang, Lin, Feng, Chen, Gao, Sun, Wang, Lai, Xu, Wang *et al.* 2018; Hatifi, Di Molfetta, Debbasch, and Brachet 2019), data prediction (Qiang, Loke, Montanaro, Aungskunsiri, Zhou, O'Brien, Wang, and Matthews 2016), metrology (Kitagawa, Broome, Fedrizzi, Rudner, Berg, Kassal, Aspuru-Guzik, Demler, and White 2012), quantum computing (Childs 2009; Qiang *et al.* 2016) and other fields. Developing the software which can inherent the advantages of quantum walk with common computers will provide large advantage on applied quantum computing, big data analysis, data prediction, metrology and common statistics, and it is also possible to expand the application of quantum walk.

Typically, quantum walk is developed with a walker moving on graphs by quantizing classical random walk. Several definitions of quantum walk, in both discrete and continuous time, are developed. In discrete time, the most popular models are coined quantum walk and Szegedy's quantum walk (Portugal 2016). Coined quantum walk is defined by a coin flip followed by a shift or hop to adjacent vertices. Szegedy's quantum walk quantizes the Markov chain process in classical random walk and realizes the evolution of states. The walker is always in the combination state of each vertex during the quantum walk, and moves in the way of the quantum walk evolution. Continuous-time quantum walk consists of a walker and an evolution (Hamiltonian) operator of the system (Farhi and Gutmann 1998). Different from the coin operator or Markov chain which is used in discrete-time quantum walk and can only change in discrete time steps, the evolution (Hamiltonian) operator can be applied with no timing restriction. For example, the walker walks at any time, and the evolution can be expressed via the Schrödinger equation (Childs 2010). Recently, more advanced quantum walks (e.g. Quantum stochastic walks) are developed to simulate open quantum systems.

Quantum walk is commonly regarded as a general calculation tool, and all quantum calculations can be formulated as a quantum walk on a graph (Childs 2009; Lovett, Cooper, Everitt, Trevers, and Kendon 2010). The graph that carries the quantum walk is composed of vertices and edges and can be expressed in the form of an adjacency matrix. The vertices on the graph represent the quantum states of the quantum walk, and the edges connecting the vertices carry the quantum states transitions between the vertices (Berry, Bourke, and Wang 2011).

During the quantum walk, based on a graph, the probabilities of the quantum walker being found on the vertices over time reflect the variation characteristics. There is an inherent spatio-temporal correlation between these probabilities. Therefore, a probability series of the quantum walker being found on a vertex is called a mode of the spatio-temporal process. With numerical algorithms like spectral decomposition of the adjacency matrix of the graph, it is possible to achieve efficient algorithmic simulation of quantum walk. Although the mathematical computation of quantum walk is clear and direct with linear algebra, the implementation of quantum walk as a productive software is still complicated.

There are several softwares that implement quantum walk. The **QWalk** software package developed by Marquezino and Portugal (2008) uses C language to realize discrete-time quantum walk simulation on one-dimensional and two-dimensional lattices. And it also realizes the visualization of two-dimensional and three-dimensional graphs of quantum walks. The **qwViz** software developed by Berry *et al.* (2011) also uses C language to realize discrete-time quantum walks. And it realizes the interactive visualization of quantum walks simulation. The **pyCTQW** software developed by Izaac and Wang (2015) based on Python and Fortran realizes distributed continuous-time quantum walks and can support continuous-time quantum walk simulation with high data volume. The **QSWalk** based on the Mathematica platform developed by Falloon, Rodriguez, and Wang (2017) realizes the further promotion of continuous-time random walks, the continuous-time quantum stochastic walks and the time evaluation of continuous-time quantum stochastic walks based on graphs. The **QSWalk.jl** software package developed by Glos, Miszczak, and Ostaszewski (2019) is built on the basis of **QSWalk** by using the Julia language, which improves the efficiency of processing large-scale matrices. On the basis of **QSWalk** and **QSWalk.jl**, the **QSW_MPI** software package, developed by Matwiejew and Wang (2021) based on Python and Fortran, realizes continuous-time quantum stochastic walk simulation, which makes it suitable for massively parallel computers and time series simulation. The algorithm simulation of quantum walks has been developed from discrete time quantum walks to continuous-time quantum walks. And then the algorithmic simulation of continuous-time quantum stochastic walks is realized.

Although there are already several applications of quantum walks in different programming languages, the existing quantum walk-related packages mainly focus on the development of quantum walk algorithms and the improvement of the data carrying capacity and algorithm simulation efficiency of quantum walks. Only a few applications of quantum walks are developed for classical statistical analysis. For example, there is no toolkit for applying time series simulated by quantum walk to data analysis. Although there are some theoretical models that try to model time series based on quantum walk (Konno 2019), none is implemented. As the classical random walks are largely used as the fundamental infrastructure for statistical analysis such as time series analysis, quantum walks may also have such advances when applied to classical time series. The bases, which are generated by quantum walks and used for time series expression, are called as modes. In quantum walk, the probability series (i.e. the mode) at each vertex are closely related and can reflect the structure of the graph, and the inherent structural characteristics of the spatio-temporal process can be represented by graph. Therefore, quantum walk is suitable for the modeling and prediction of spatially correlated time series. As different modes of quantum walks can be generated with different evolution parameters, it is possible to model the time series characteristics of different structures by efficiently using the modes under different parameters and integrating different mode selection, modeling and model evalution methods. Therefore, the coupled analysis of significant modes, temporal and spatial correlations and evolutionary laws in time series can be realized.

By implementing the above processes and integrating the implementation with easy-to-use software packages like R, the application of quantum walk-based data analysis can be widely extended.

Based on the above, the **QWDAP** package was built based on the R language. The **QWDAP** package provides a set of tools aiming to fully apply the spatio-temporal characteristics of quantum walks to data analysis. **QWDAP** is a software package for multi-time series modeling and prediction based on quantum walks and includes three modules: Basis Generation, Data Modeling and Prediction, and Model Evaluation. In the Basis Generation module, **QWDAP** extracts the modes from the evolution process of continuous-time quantum walk for time series analysis. In the Data Modeling and Prediction module, the mode selection, linear and nonlinear regression and temporal-correlated models like Vector Autoregressive (VAR) are developed to extract the significant modes from different perspectives. Develop linear, nonlinear and temporal-correlated relationships between modes and the original time series and predict the time series with or without considering the spatio-temporal correlations. In the Model Evaluation module, multiple evaluation indexes are developed to evaluate the performance of the model.

This paper is organized as following: In Section 2, we propose the statistical problem that the **QWDAP** software mainly solves. In Section 3, we introduce the main functions and classes in the **QWDAP** package. In Section 4, we take the traffic volumes of highway traffic system as an example and use two different combinations of modes for modeling and prediction respectively. In Section 5, summarizes and prospects of **QWDAP** package are presented.

2. Problem Definition and Basic Idea

2.1. Problem Definition

Multivariate time series is the description of time series with systematic correlation. In addition to the temporal characteristics, multivariate time series also have spatial interrelationships. However, the existing multi-channel network diagrams cannot accurately describe and express them.

Definition 1. Graph-associated time series is a combination of a graph and a set of time series corresponding to the vertices of the graph, which can be visualized as the GT in Figure 1. The graph-associated time series GT is derived from the graph structure G and the corresponding features TS of the graph's time series by some operation, and its mathematical mapping relation is

$$GT = G + TS \tag{1}$$

where "+" is used only for the connection. The graph structure G is composed of vertices and edges and its mathematical expression is

$$G = (V, E) \tag{2}$$

where $V = \{v_1, v_2, \dots, v_N\}$ is the set of N vertices and $E = \{e_1, e_2, \dots\}$ is the set of edges. And the vertices v_i in the graph-associated time series correspond to different time series, under some rule mapping F, with

$$F(TS) = \{v_1, v_2, \dots, v_N\} \tag{3}$$

In the graph correlation time series the edge e_i corresponds to the correlation of different vertices (different time series). And its correlation is often influenced by multiple factors that act together, such as connectivity, causality, etc. Due to the difficulty of expressing and describing multi-channel time series, this paper proposes quantum walk for simulation.

Definition 2. Feature decomposition. The original feature decomposition is decomposed to ensure that the extracted feature components correspond to the elemental graph structure while keeping the original graph feature structure unchanged when the feature extraction is simulated with quantum walk. The mathematical expressions are as follows

$$\begin{cases} G = G_1 = G_2 = \cdots = G_i = \cdots = G_m \\ TS = TS_1 + TS_2 + \cdots + TS_i + \cdots + TS_m \end{cases} . \tag{4}$$

The quantum walk of the time series of each vertex at moment t simulates the probability of the feature as $P(t)$, so that the mathematical expression of the quantum walk of the time series containing the feature, driven by the systematic events, is as follows

$$TS = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{v_i v_j}{L_{ij}} P(t) \cdot \delta \tag{5}$$

where L_{ij} denotes the distance between vertices v_i and v_j, and δ represents the correlation taking the value $\{0, 1\}$.
This completes the modal characteristics of the change in graph structure in a multivariate time series mixed signal, which is obtained as

$$TS_i(t) = \{TS_1(t), TS_2(t), \cdots, TS_N(t)\} \tag{6}$$

The time series features of the quantum walk simulation can be expressed in matrix form.

$$\begin{pmatrix} TS_1 \\ TS_2 \\ TS_3 \\ \vdots \\ TS_N \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1m} \\ S_{21} & S_{22} & \cdots & S_{2m} \\ S_{31} & S_{32} & \cdots & S_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{Nm} \end{pmatrix} \cdot \begin{pmatrix} t_1 & t_2 & \cdots & t_m \end{pmatrix}^{\top} \tag{7}$$

where $\begin{pmatrix} S_{i1} & S_{i2} & \cdots & S_{im} \end{pmatrix}$ represents the corresponding variation feature on vertex v_i.

Definition 3. Matching. By means of quantum walk, the features of different time series can be extracted and summed according to certain rules of operation, and finally compared with the original features. The mathematical expression is as follows

$$Sum\{GT\} - \{GT\}_{orig} \leq \sigma \tag{8}$$

where σ is a statistical indicator, which can indicate the mean extreme value, etc.

Definition 4. Prediction is the operation to get the data unobservated. For each time scale p according to the time point t, we can get $TS(t+p) = TS_1(t+p) + TS_2(t+p) + \cdots + TS_m(t+p)$,

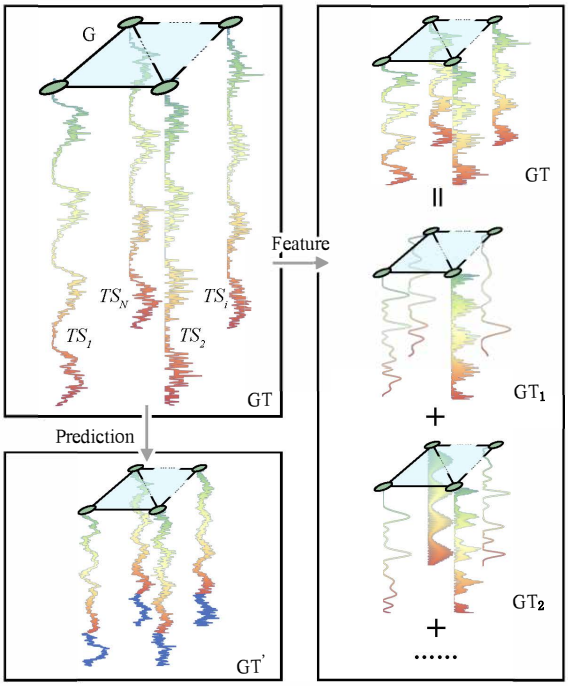

Figure 1: Graph-associated time series.

which can be express as

$$\begin{pmatrix} TS'_1 \\ TS'_2 \\ TS'_3 \\ \vdots \\ TS'_N \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1m} & \cdots & S_{1(m+p)} \\ S_{21} & S_{22} & \cdots & S_{2m} & \cdots & S_{2(m+p)} \\ S_{31} & S_{32} & \cdots & S_{3m} & \cdots & S_{3(m+p)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{Nm} & \cdots & S_{N(m+p)} \end{pmatrix} \cdot \begin{pmatrix} t_1 & t_2 & \cdots & t_m & \cdots & t_{(m+p)} \end{pmatrix}^\top \quad (9)$$

where $\begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1m} & \cdots & S_{1(m+p)} \end{pmatrix}$ represents the corresponding variation feature on vertex v_i.

The analysis method of graph-associated time series is different form the method analysis the traditional time series, which will consider the spatial relationship between the time series. However, the traditional method like Empirical Modal Decomposition (EMD) and Singular Spectrum Analysis (SSA), they analysis the time series one by one and neglect the spatial correlaction of the time series. In reality, the generation of time series is mostly accompanied by spatial interactions, such as geographic data, especially the rise of geographic spatiotemporal big data (Song, Wang, Xiang, and Zomaya 2017) is inseparable from the analysis of the spatial location of time series.

Traditional time series analysis methods, such as Fourier transform, short-time Fourier transform, wavelet transform and continuous wavelet transform, extract the characteristic components of the original series, which reveal certain periodically transformed characteristics of the time series, but Fourier transform and short-time Fourier transform cannot handle non-stationary time series (Gao and Shang 2019), wavelet transform and continuous wavelet

transform are not adaptive (Zedda and Singh 2002). Adaptive time series analysis methods, such as Empirical Modal Decomposition (EMD) and Singular Spectrum Analysis (SSA), extract the feature components from the original time series, which can reveal the periodic fluctuation characteristics of the original data on the one hand, and provide support for modeling and prediction models on the other hand. However, the above two types of time series decomposition methods depend on the original time series and are easily affected by the initial experimental data, especially the analysis of individual time series lacks the expression of the spatial connection of time series. Therefore, there is an urgent need for an analytical method that can perform a holistic analysis of graph-associated time series.

2.2. Basic idea

In order to solve the above two problems, the paper proposes a "random generation, directional filtering" feature selection method for graph-associated time series and a time series prediction method based on feature series. To obtain the features of time series, quantum walks are used for random feature generation, and model-driven or data-driven selection methods are used to get the part features that are significantly correlated with the original time series. Quantum walk, as a graph-based random data generation method, generates features that have randomness in addition to spatially interacting features, so that they are also expressive for unstable time series.
In order to predict the original time series, we establish the mapping relationship between the original time series and the features, and predict the original time series with longer time features generated by quantum walk according to the same mapping relationship. The mapping relationship between the original time series and these features is established by using regression analysis.
For graph-associated time series analysis, we use a graph-based random data generation method, quantum walk, to generate a large number of basic features, and select out the features related to specific time series. The feature model based on the quantum walk mechanism of the original time series is established by regression methods such as linear, nonlinear, and temporal-correlated methods, which can realize the prediction of graph-associated time series.

3. The Structure of QWDAP Package

From the perspective of data analysis, the **QWDAP** package can be divided into three modules: Basis Generation, Data Modeling and Prediction, and Model Evaluation. The algorithms used in each module are shown in Figure 2.
In the Basis Generation module, **QWDAP** extracts the modes from the evolution process of continuous-time quantum walk for time series analysis. The continuous-time quantum walk is performed on a given graph with an initial state, and the probability series of the walker being found at each vertex is obtained by the discretization sampling of the continuous-time quantum walk at equal intervals. The continuous-time quantum walk, which is based on the spectral decomposition of an adjacency matrix and the combination of the eigen-values and eigen-vectors, is used to simulate the random movement of a quantum walker. Starting from a certain vertex the walker is always in the combination state of each vertex without being observed during the quantum walk process. In order to extract the change characteristics

of quantum walk corresponding to each vertex, the package samples the probabilities of the walker being found at each vertex with a specific scale. Sampling at multiple scales can yield a large number of basic features with different characteristics. These probability series contain random variation features are called modes. And the modeling and prediction of the observed time series use the modes sampled on the corresponding vertices. The Data Modeling and Prediction module applies the modes generated by quantum walks to the analysis of the graph-associated time series.

In the Data Modeling and Prediction module, some regression methods are used to establish the relationship between the original observed time series and the modes through spatial or temporal correlation. And then the established relationship can be used for modeling and prediction of the original time series. Considering that there may be linear and nonlinear, time-correlated and time-non-correlated multiple relationship structures between the time series and modes, the **QWDAP** package includes three types of modeling methods: linear regression, nonlinear regression and the temporal-correlated regression. The linear regressions in the **QWDAP** package include Stepwise Regression, Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR). Nonlinear regression includes Projection Pursuit Regression (PPR) and temporal-correlated regression like Vector Autoregressive (VAR). The **QWDAP** package includes two mode selection methods, Stepwise Regression (Steyerberg, Eijkemans, and Habbema 1999) and RReliefF (Robnik-Šikonja and Kononenko 2003). Mode selection can extract some highly feature correlated modes with the original time series from the whole modes. The selected part are used to to model the original time series, and mode selection greatly reduces the useless modes for modeling and avoids overfitting in modeling.

The Model Evaluation module is used to evaluate the correlation between the modeling and prediction results and the original time series. **QWDAP** package provides the calculation of the Coefficient of Determination (R^2) , Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for two time series.

More details of the three modules of the **QWDAP** package are presented in the following sections.

3.1. Basis Generation

In order to describe the evolution process of quantum walks, an arbitrary undirected graph is used to describe the process (Izaac and Wang 2015). Suppose $G = (V, E)$ is a graph without considering its width, where V is the set of N vertices and E is the set of edges. For any vertex v, $\Gamma_v = \{u \in V, (u, v) \in E\}$ represents the vertex adjacent to v (neighbor vertex). The adjacency matrix A of graph G can be defined as

$$A_{uv} = \begin{cases} 1, & if\,(u,v) \in E \\ 0, & otherwise \end{cases} . \tag{10}$$

where $A_{uv} = A_{vu}$ and $A_{vv} = 0$.

Unlike the classical random walk, the process of quantum walks is not a Markov chain (Tsuji, Estrada, Movassagh, and Hoffmann 2018). Classically, the evolution of state vector $|\varphi(t)\rangle$ with time t can be described as the form of Schrödinger equation (Childs 2010).

$$i\frac{d}{dt}|\varphi(t)\rangle = H|\varphi(t)\rangle . \tag{11}$$

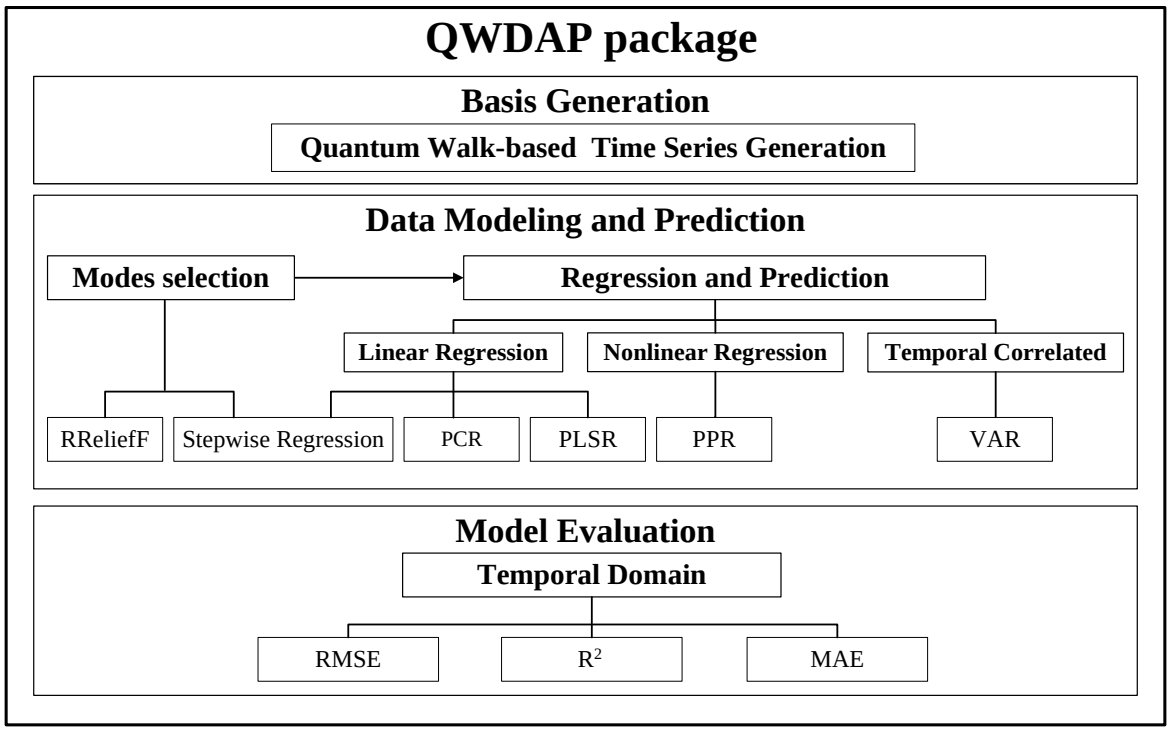

Figure 2: Structure diagram.

Where the time-independent Hamiltonian H is an $N \times N$ Hermitian matrix, such as the adjacency matrix or Laplacian of the graph (Izaac and Wang 2017). For simplicity, the Hamiltonian H is replaced by the adjacency matrix A of the graph G in this paper. And $|\varphi(t)\rangle \in \mathbb{C}^N$ is a complex-valued state vector.
The evolution equation can be solved from formula 11 with an initial state $|\varphi(0)\rangle$. The state vector $|\varphi(t)\rangle$ at time t can be expressed as

$$|\varphi(t)\rangle = e^{-iHt} |\varphi(0)\rangle , \tag{12}$$

where the e^{-iHt} is the time evolution operator (Kempe 2003), which is used to construct the dynamically evolved quantum walks (Biamonte, Faccin, and De Domenico 2019).
The state vector $|\varphi(t)\rangle$ of the quantum walk at time t is a complex linear combination of the basis states. The basis state corresponding to the walker at vertex $v \in V$ is expressed as $|v\rangle$ (Sett, Pan, Falloon, and Wang 2019). The complex-valued state vector $|\varphi(t)\rangle$ at time t can be expressed as

$$|\varphi(t)\rangle = \sum_{v \in V} a_v(t) |v\rangle , \tag{13}$$

where $a_v(t) = \langle v|\varphi(t)\rangle \in \mathbb{C}$ represents the probability amplitude of the walker being found at vertex $|v\rangle$ at time t. The probability of finding the walker at any vertex at time t is given by the squared modulus of the appropriate element of $|\varphi(t)\rangle$ (Tsuji *et al.* 2018). Therefore, the probability of the walker being found at vertex v at time t can be expressed as

$$p(|v\rangle, t) = |a_v(t)|^2 = |\langle v|\varphi(t)\rangle|^2. \tag{14}$$

Since the probability of a walker appearing at each vertex state is conservative, it satisfies $\sum_{v\in V} p\left(|v\rangle, t\right) = 1$ at time t. Formula 14 expresses the probability of the walker being found at vertex v at time t. To obtain the state vector $|\varphi\left(t\right)\rangle$, the time evolution operator e^{-iHt} with matrix and complex coefficients needs to be calculated. However, direct calculation of the time evolution operator requires a large number of exponential matrix calculations. But existing matrix calculation libraries such as **bigalgebra** (Bertrand, Kane, Emerson, and Weston 2021) are also difficult to efficiently calculate the matrix exponents. Therefore, replacing the Hamiltonian with its own eigen-values will greatly reduce the computational difficulty. The spectral decomposition of the Hamiltonian is expressed as

$$H = \Phi\Lambda\Phi^{\top}, \tag{15}$$

where Φ is the $N \times N$ matrix and can be expressed as

$$\Phi = \left(\phi_1|\phi_2|\cdots|\phi_n|\cdots|\phi_N\right). \tag{16}$$

The ordered eigen-vectors ϕ_ns of H are set as columns.

$$\Lambda = diag\left(\lambda_1, \lambda_2, \ldots, \lambda_n, \ldots, \lambda_N\right) \tag{17}$$

is the $N \times N$ diagonal matrix and the ordered eigen-values λ_n of H are taken as elements. $\lambda_1, \lambda_2, \ldots, \lambda_N$ are the eigen-values of matrix H, and the corresponding eigen-vectors are $\phi_1, \phi_2, \ldots, \phi_N$. Using the spectral decomposition of the Hamiltonian, the time evolution operator can be expressed as formula 18 (Rossi, Torsello, and Hancock 2015).

$$e^{-iHt} = \Phi e^{-i\Lambda t}\Phi^{\top}. \tag{18}$$

The formula 12 can be expressed as

$$|\varphi\left(t\right)\rangle = \Phi e^{-i\Lambda t}\Phi^{\top}\left|\varphi\left(0\right)\right\rangle. \tag{19}$$

For large matrices, an efficient spectral decomposition method is required. The QR decomposition is one of the most well-known and useful tools in numerical linear algebra (Duersch and Gu 2017). Fast adaptations of the QR algorithm are considerably simplified by transforming the matrix into an upper-Hessenberg form, and the QR decomposition is proved to be backward stable (Eidelman, Gemignani, and Gohberg 2008). The spectral decomposition with the QR decomposition is used in the **QWDAP** package to calculate the eigen-values and eigen-vectors of the Hamiltonian H. Before the QR decomposition, the matrix is converted to upper-Hessenberg matrix by matrix operations such as Householder transformation (Van Zee, Van De Geijn, Quintana-Ortí, and Elizondo 2012).
The eigen-values and eigen-vectors are used to solve the time evolution operator as formula 18. The evolution of state vector is simulated by the calculation of eigen-values, eigen-vectors and time t, which is realized by formula 19.
So far, the state vector $|\varphi\left(t\right)\rangle$ evolved from the initial state $|\varphi\left(0\right)\rangle$ at time t is obtained, the probability amplitude of each vertex can be extracted, and the probability can be calculated by the squared modulus of the probability amplitude as formula 14. The probability series corresponding to all vertices are obtained by performing formula 14 using a set of times with a same interval. This corresponds to the sampling of a set of continuous probabilities, which will be described later as the sampling of continuous time quantum walks. In order to obtain

sets of probabilities for data analysis, time series with different variances are used for multiple rounds of samplings. For ease of understanding, scaling factors $\{k_j\}_{j=1}^{J}$ are defined, where J indicates the number of sampling rounds. And the time t is replaced by $k_j t$. The t in $k_j t$ is represented by a series of natural numbers, $t = 0, 1, 2, \ldots$, and $k_j \in \mathbb{R}^+$ represents the sampling interval. Therefore, formula 19 can be expressed as

$$|\varphi(k_j t)\rangle = \Phi e^{-i\Lambda k_j t}\Phi^{\top} |\varphi(0)\rangle . \tag{20}$$

A probability series generated with a scaling factor on a vertex is called a mode which is used as a basic unit to describe the data used for analysis. A series of scaling factors $\{k_j\}_{j=1}^{J}$ are used to generate the modes for data modeling and prediction. The modes simulated with different scaling factors possess different temporal evolution characteristics. As the modes are generated based on a graph, there is an inherent spatial correlation between the modes. The probabilities generated by quantum walks are presented in the form of a percentage in the package. In order to reduce the data storage, the program converts the result of floating-point numbers into integers by default. For users with high data accuracy requirements, the original floating-point result can be obtained.

3.2. Data Modeling and Prediction

Using different scaling factors, the modes on the vertices can be produced, reflecting the law of change at different scales of the continuous time quantum walk. The modes can be used as bases to approximate complex time series. In the Basis Generation module, by adjusting parameters k_j, suffericent modes can be generated. Regression methods can then be applied to establish the relationship between the original observed time series and the generated modes. Data modeling and prediction with quantum walks are based on the orderly organization of the modes. In order to improve the effect of modeling and the accuracy of prediction, the scaling factors are increased to simulate as many modes as possible. However, the original time series may not be affected by all the modes. Therefore, to accommodate the evolutionary features and structural patterns of different time series, it is important to filter out the modes that can be used to represent the characteristics of the original time series among all the generated modes.

Mode selection

Mode selection is used to select a combination of modes for modeling among all the modes generated by quantum walks. There are mainly two types of mode selection methods: model-driven and data-driven. The representative of model-driven mode selection methods is Stepwise Regression. Stepwise Regression assumes that there is a linear correlation between the original time series and the modes. And during the Stepwise Regression process, the mode combination used for linear regression is gradually modified until getting the optimal mode combination that can be used to approximate the original time series. The data-driven mode selection method, such as RReliefF, obtains the optimal mode combination by measuring the correlation coefficient between the original time series and the modes. RReliefF calculates the weight of each mode compared with the original time series through the k nearest neighbor method. Both of the two methods can select a combination of modes that can be used to simulate the original time series and build compact and interpretable regression models. In view of the difference in the mechanism of the two selection methods, the result obtained

by Stepwise regression is the optimal mode combination, while RReliefF can get the ordered results of all modes sorted by weight compared with the original time series. The selected modes can model the original time series relatively well while reducing the amount of calculation. The two mode selection methods are introduced separately below. Stepwise Regression is also used for modeling and prediction, and we will explain it in detail in Section 3.2.2.
The RReliefF algorithm is proposed by Robnik-Šikonja and Kononenko (1997), which is an extension of ReliefF in the field of regression applications. In the RReliefF algorithm, the k proximity weight for each sample of the modes is calculated according to the original time series to sort all modes and select the modes by the weights. For each mode, all possible k (representing k nearest instances) are tested and the highest score is returned. The weight of each mode relative to the corresponding original time series can be obtained, and the amount of modes required can be selected according to the weight.

Regression and Prediction

`Linear Regression`
Linear regressions are used to model the original time series based on the linear combination of multiple modes and find the best parameter of each mode with the original time series constraints. There are three linear regression methods provided by the **QWDAP** package: Stepwise Regression, Principle Component Regression (PCR) and Partial Least Squares Regression (PLSR). In regression analysis based on modes generated by quantum walks, suitable parameters are selected and the linear combination of modes is used to approximate the original time series to be modeled. Let

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon, \tag{21}$$

where Y is the fitted time series, $X_1, X_2, \ldots, X_p$ are the modes generated by quantum walks, $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients of the modes in the model, β_0 is the constant term, and ϵ is the residual.
The three linear regression methods are essentially expressing original time series through linear combinations of the modes, but different linear regression methods have different specific algorithms for determining coefficients. The three regression methods are introduced below.
Stepwise Regression is a process of screening variables in regression analysis, which is widely used in data fitting (Burkholder and Lieber 1996) and pattern recognition (Steyerberg *et al.* 1999). The basic idea of Stepwise Regression is to gradually change the combination of modes to achieve a relatively optimal fitted effect. Whether a mode should be retained or not depends on the significance degree of the regression through the combination of modes and the original time series, and finally the selected combination of modes is used to establish a linear model. Forward selection, Backward elimination and Bidirectional elimination are three specific algorithms of Stepwise Regression. These three different algorithms constantly change the combination of the modes and judge whether to accept the change through a judgment standard. There are many judgment standards that can be used in Stepwise Regression. For example, the Akaike Information Criterion (AIC) (Yamashita, Yamashita, and Kamimura 2007) is used in **QWDAP** to judge the fitted effect of the model.

Principle Component Regression (PCR) applies principal component analysis to linear regression. It uses highly correlated modes to gather independent principal component combinations to establish a regression model, which is widely used in regression analysis and prediction

(Ieong, Lou, Ung, and Mok 2015). The PCR can overcome the interference of multicollinearity (Liu, Kuang, Gong, and Hou 2003). The principal component is the transformation of a group of potentially correlated modes into a group of linearly uncorrelated series through orthogonal transformation, and the transformed series are the constituent components. Derive a few principal components from the original time series, so that these series retain as much information about the original time series as possible. The characteristic of principal component analysis is to reveal the internal structure of multiple series through a few principal components.

The basic principle of Partial Least Squares Regression (PLSR) is to find a linear regression model by projecting the modes and the original time series into a new space through projection, which is a commonly used prediction method (Mevik and Wehrens 2007; Helland, Sæbø, Almøy, and Rimal 2018). PLSR is related to the PCR, but instead of looking for the hyperplane with the largest variance between the modes and the original time series.

`Nonlinear regression`

Projection Pursuit Regression (PPR) is a nonlinear regression analyzing method, which aims at high-dimensional data with multiple samples as well as multivariate, and is widely used in prediction (Rajeevan, Pai, Kumar, and Lal 2007). The basic idea of PPR is to project the high-dimensional data to a low-dimensional space (1-3 dimensions), find a projection that can reflect the structure or characteristics of the high-dimensional data, and perform regression analysis. The key to PPR is to find the direction of projection.

The Projection Pursuit Regression model can be expressed as

$$F(x) \sim \sum_{m=1}^{M} \beta_m G_m(Z_m) = \sum_{m=1}^{M} \beta_m G_m(\sum_{j=1}^{P} a_{mj}^{\top} X), \tag{22}$$

where $G_m(Z_m)$ is the m-th ridge function. β_m is a weight, which represents the contribution of the m-th ridge function to the output value. $Z_m = \sum_{j=1}^{P} a_{mj}^{\top} X$ is the independent variable of the ridge function, which represents the projection of the P-dimensional vector X in the a_m direction. a_{mj} is the j-th component of the m-th projection direction. P is the dimension of the input space, which is required to satisfy the formula $\sum_{j=1}^{P} a_j^2 = 1$.

`Temporal Correlated`

Vector Autoregressive (VAR) is often used to predict time series systems with inherently correlated factors and analyze the dynamic effects of random disturbances on variable systems. The VAR method constructs a model by taking each endogenous variable in the system as a function of the lag value of all endogenous variables in the system (Johansen 2002) and is often used in series correlation analysis (Goebel, Roebroeck, Kim, and Formisano 2003).

For the multivariate time series $Y \in \mathbb{R}^{N \times T}$, in the case of any t time interval, the VAR(k) model can be expressed as

$$y_t = \sum_{k=1}^{d} A_k y_{t-k} + \epsilon_t, \ t = d+1, ..., T, \tag{23}$$

where $y_t = (y_{1t}, y_{2t}, ..., y_{Nt})^{\top} \in \mathbb{R}^{N \times T}$, and $A_k \in \mathbb{R}^{N \times N}$ is the coefficient matrix of VAR, ϵ_t is the noise, and k is the lag order.

3.3. Model Evaluation

Modes generated by quantum walks are used to build a time series model and make predictions based on the model. However, the accuracy of the model, the efficiency of the modeling and the stability of the program need to have a clear indicator to reflect. T he regression methods are used to establish the correlations between the original time series and the modes, and generate a fitted series of the original time series using the modes, which will be used for model evaluation. In terms of the effects of modeling and prediction, the correlation coefficients between the fitted series and the predicted series relative to the original time series are analyzed from the perspective of the time domain. And the calculation time and physical memory occupation are analyzed to evaluate the efficiency of the entire model. The stability index mainly considers the robustness of the program, and the solution for some unreasonable or unsupported data input. From the evaluation of the accuracy of modeling, **QWDAP** can be also used to calculate some indexes of data correlation. The package can generate multi-scale, structurally heterogeneous modes based on a graph, and the modes can be used to perform regression analysis and prediction on time series. By operation of the Model Evaluation module, the evaluation indexes, like Coefficient of Determination (R^2), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), between series can be calculated.

4. The Use of QWDAP Package, a Sample of Traffic Volumes

In this section, we will discuss the specific functional implementation of the three main modules in **QWDAP** and we choose the traffic volumes of highway traffic system as an example to illustrate the usage of the package. The highway traffic system can be regarded as a relatively closed system that changes over time. The traffic volumes counted at each highway station on the same section of highway are interacting with each other. The traffic volumes counted at each station are statistical series that changes over time, and the differences in driving habits of drivers lead to differences in the characteristics of the time series. The analysis of highway multi-station time series is the analysis of the graph-associated time series. Assuming that the various driving behaviors of the driver on the expressway behaves as a mode, the **QWDAP** is used to extract the change characteristics of the continuous time quantum walk at different scales that may be the same as the local change characteristics of traffic volumes. And the modes generated by quantum walks are used to model and predict the traffic volumes.

4.1. Overview

In Table 1, we briefly show the functions included in the three main modules of the **QWDAP** package and the role of each function, and the meaning of each parameter in these functions is explained in Table 2. The experimental process is shown in Figure 3.

4.2. Data Modeling and Result Evaluation of Traffic Volumes

In this section, we establish an adjacency matrix according to the spatial characteristics of the stations distribution in the highway traffic system. **QWDAP** is used to extract the features of quantum walks and obtain the modes for data analysis. In addition, the traffic volumes are modeled based on the modes, and the data predictions are performed according to the models. The results are evaluated in the time domain.
In the **QWDAP** package, the Basis Generation module is used to generate bases for data

Modules	Tools	Function	Arguments
Basis Generation	Quantum walk-based probability generation	qwdap.qwalk	edges, startindex, lens, scals, getfloat.
Data Modeling and Prediction	Mode selection	qwdap.sws	real, ctqw, select_method, plotting.
		qwdap.rrelieff	real, ctqw, index, num, plotting.
	Linear regression	qwdap.swr	in_data, data_range, plotting.
		qwdap.pcr	in_data, data_range, plotting.
		qwdap.plsr	in_data, data_range, plotting.
	Nonlinear regression	qwdap.ppr	in_data, data_range, plotting.
	Temporal correlated	qwdap.var	in_data, data_range, plotting.
	Prediction	qwdap.predict	in_model, data_range.
Model Evaluation	Evaluation index	qwdap.eval	series1, series2.

Table 1: Modules of the QWDAP package.

Arguments	Description
edges	An adjacency matrix.
startindex	The initial position of the quantum walker.
lens	The number of records required in a round of sampling.
scals	The scaling factors for sampling.
getfloat	Choose whether to return floating point data.
real	The original time series.
ctqw	A 'CTQW' object with modes generated by function `qwdap.qwalk()`.
select_method	The stepwise regression method.
plotting	Choose whether to plot.
num	The number of modes to be selected by RReliefF.
in_data	A 'QWMS' object with selected modes.
data_range	The index range of modes used for modeling or prediction.
in_model	A 'QWMODEL' object, a built model for prediction.
series1	The first time series.
series2	The second time series.

Table 2: Parameters description.

modeling and prediction, i.e., simulate quantum walks, and the adjacency matrix needs to be input. The initial position of the quantum walker, scaling factors and other settings related to the quantum walk can be input as parameters. The Data Modeling and Prediction module has tools for multi-time series modeling and prediction, as well as mode selection operations.

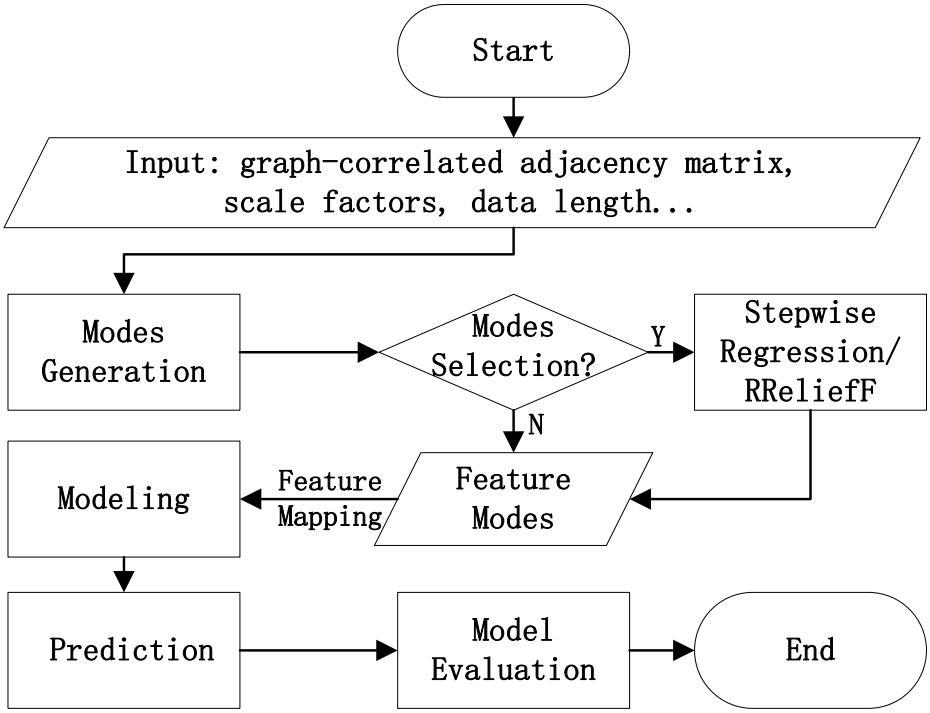

Figure 3: Experimental flow chart.

The Model Evaluation module can be used to calculate the time domain correlation indices of two time series.

The traffic volumes of the highway traffic system we used in this paper are integrated in the **QWDAP** package as the `trafficflow` dataset. This dataset is the traffic statistic of Nanjing-Changzhou section of Shanghai-Nanjing Expressway in China starting from 0:00AM Dec-01-2015 with the time interval of 10 minutes. This data set has a total of 720 records of 7 research stations, namely Tangshan (N1), Jurong (N2), Heyang (N3), Danyang (N4), Luoshuyan (N5), Xuejia (N6) and ChangzhouBei (N7). The stations are connected end to end in turn as shown in Figure 4.

Data set analysis

The `trafficflow` dataset in **QWDAP** is a collection of multi-time series, respectively representing the changes of traffic volumes at 7 stations. The Nanjing Station is the entrance, and the 7 stations data included in the dataset are all outbound traffic statistics data. The changes of the traffic volume at each station in the dataset are shown in Figure 5.

Basis Generation

The function `qwdap.qwalk()` is used to generate modes by quantum walks and the modes are used for traffic volumes modeling and series prediction. The adjacency matrix needs to be input for quantum walk simulation as parameter `edges`. The 7 research stations in the `trafficflow` data set are connected end to end, and the adjacency matrix of these 7 stations

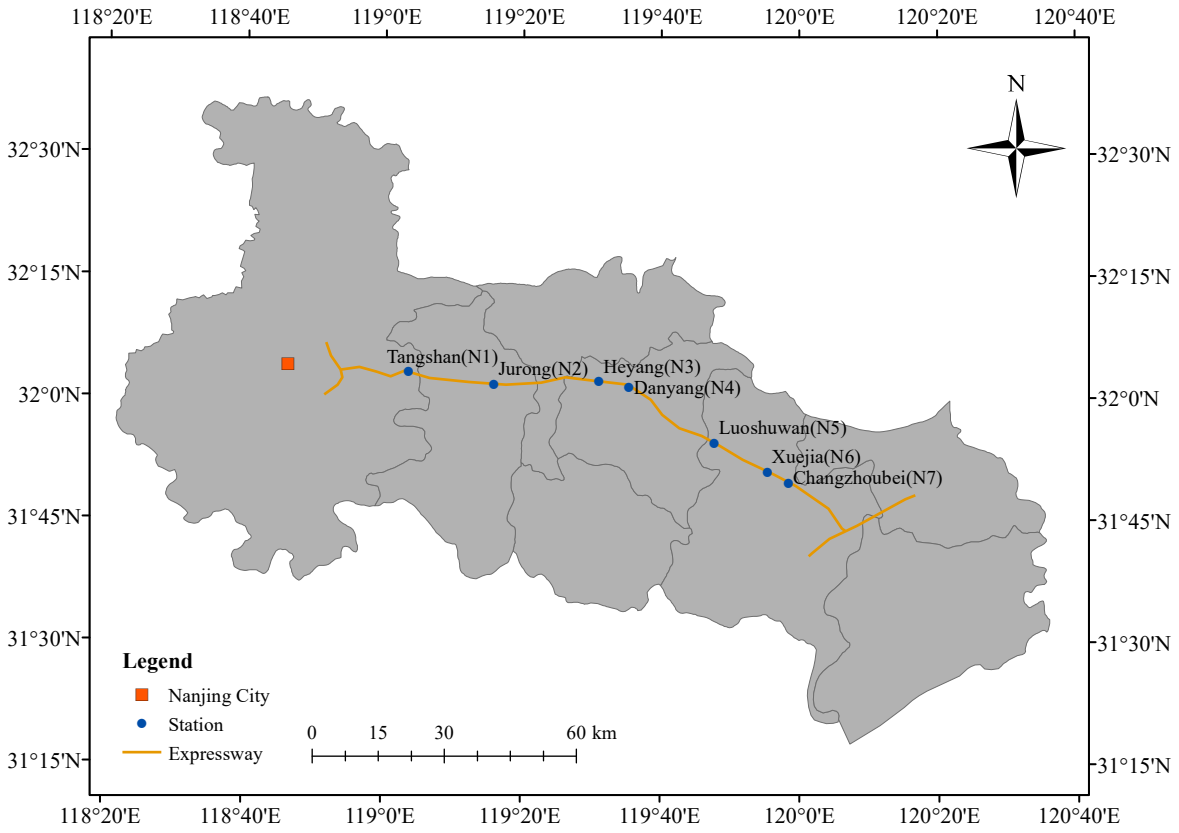

Figure 4: Research area and stations distribution.

is set as *edges*, as shown in formula 24.

$$edges = \begin{bmatrix} 0, & 1, & 0, & 0, & 0, & 0, & 0 \\ 1, & 0, & 1, & 0, & 0, & 0, & 0 \\ 0, & 1, & 0, & 1, & 0, & 0, & 0 \\ 0, & 0, & 1, & 0, & 1, & 0, & 0 \\ 0, & 0, & 0, & 1, & 0, & 1, & 0 \\ 0, & 0, & 0, & 0, & 1, & 0, & 1 \\ 0, & 0, & 0, & 0, & 0, & 1, & 0 \end{bmatrix}, \tag{24}$$

For the traffic volumes corresponding to the `trafficflow` dataset, the `N1` station is the first exit, so `N1` is set as the initial position of the quantum walker. The `trafficflow` dataset has a total of 720 records, so the record length obtained in a round of sampling is set to 720. In order to generate cases of the walker distribution as much as possible, we set the initial scaling factor to 0.01, and set 2000 scaling factors with a 0.01 increment. Then we perform the following operations to generate the modes and store them in `qw.data`, which is a CTQW object.

```
R> edges <- matrix(c(0,1,0,0,0,0,0,
+                    1,0,1,0,0,0,0,
+                    0,1,0,1,0,0,0,
```

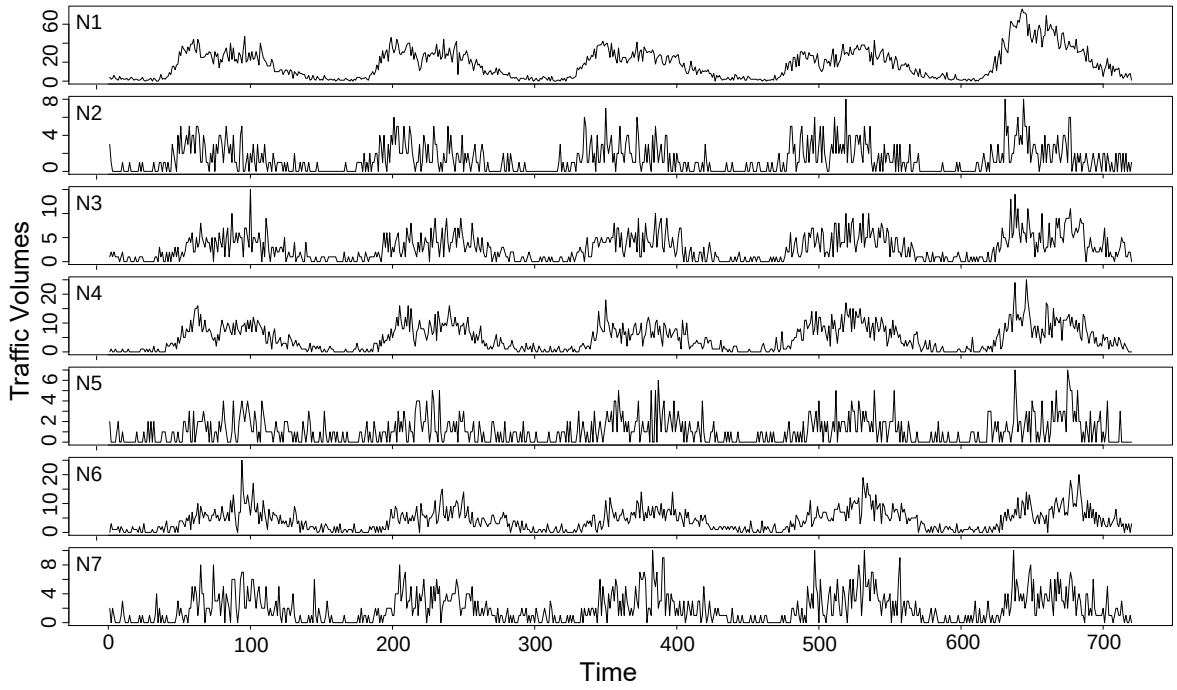

Figure 5: Traffic volumes.

```
+                     0,0,1,0,1,0,0,
+                     0,0,0,1,0,1,0,
+                     0,0,0,0,1,0,1,
+                     0,0,0,0,0,1,0), nrow = 7)
R> qw.data <- qwdap.qwalk(edges=edges, startindex=1, lens=720,
+                          scals=seq(from=0.01, by=0.01, length.out=2000))
```

The above operation will get a object of class 'CTQW' with an array whose dimension is $720 \times 7 \times 2000$. Every 720 records of `qw.data[ , , i]` with any $i \in [1, 2000]$ are a set of modes generated in a round of sampling simulated by a scaling factor. The modes generated by the first four scaling factors are shown in Figure 6. The figure contains a total of 4 groups of graphs with different time scales, where each graph represents a mode. The change characteristics of the former group are the local change characteristics of the latter group.

Modeling and prediction of the highway traffic volumes

In this part, modes generated by quantum walks are used to model the traffic volumes and predict based on the established model. Before modeling and prediction, the modes are filtered to obtain a mode combination that is highly significant with the corresponding traffic volume.

The `qwdap.qwalk()` function is used to conduct 2000 rounds of sampling with different scaling factors, so each station corresponds to 2000 modes, some of which have low correlation with the traffic volumes. In order to avoid overfitting and improve the computational efficiency of subsequent modeling, two mode selection methods are used to extract modes that are highly correlated with the original time series for modeling.

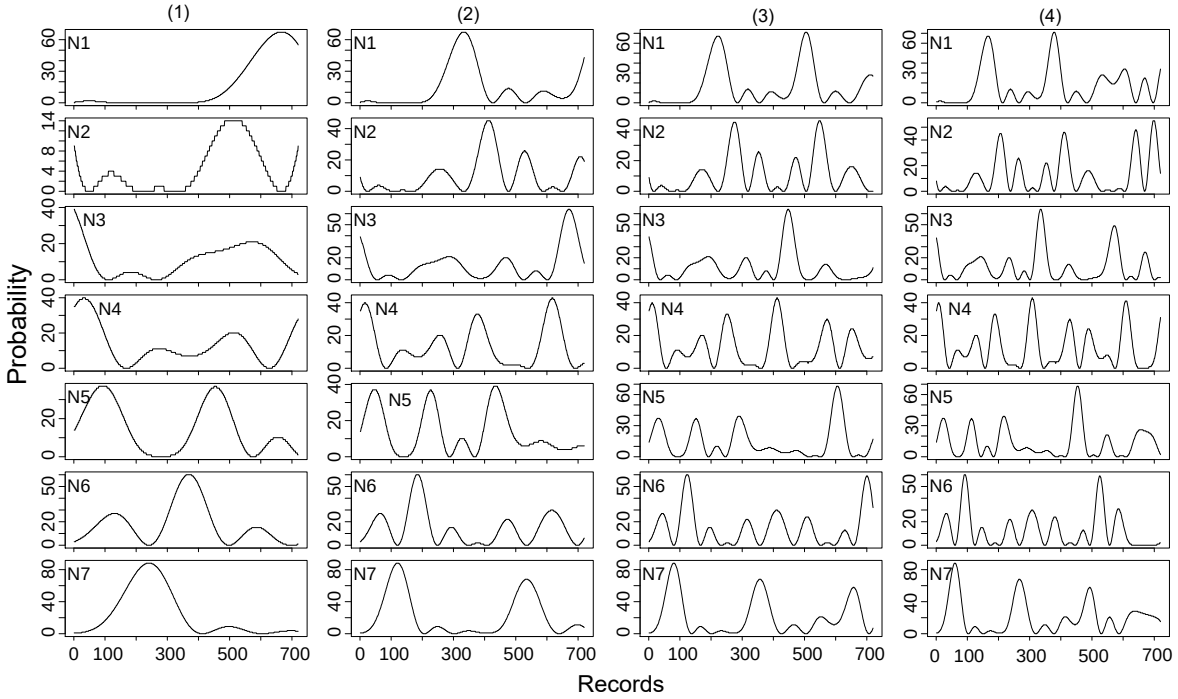

Figure 6: A broken line graph of the modes generated by quantum walks of the first four rounds of sampling.

The **QWDAP** package provides Stepwise Regression and RReliefF for mode selection. The Bidirectional elimination method of Stepwise Regression is chosen below. The parameter `select_method` is set to `bidirection`. Stepwise Regression is performed on the modes according to the original time series, and the result is stored in the list `res.sws`.

```
R> data("trafficflow")
R> res.sws <- list()
R> for(i in c(1:7)){
+   res.sws[[i]] <- qwdap.sws(real=trafficflow, ctqw=qw.data, index=i,
+                             select_method="bidirection", plotting=TRUE)
+  }
```

The list `res.sws` is a combination of 7 objects of class 'QWMS', which stores the selected modes of the 7 stations.
RReliefF is also used to select the modes of the 7 stations separately. When using RReliefF to select modes, users can specify the number of modes that needs to be obtained. We choose 50 modes for each station and set the parameter `num` to 50. When the parameter `num` is -1, it means that no filtering is performed, and the feature weight ranking of all modes is obtained. Proceed as follows:

```
R> res.rrelieff <- list()
R> for(i in c(1:7)){
+   res.rrelieff[[i]] <- qwdap.rrelieff(real=trafficflow, ctqw=qw.data[,i],
```

```
+                                          index=i, num=50, plotting=TRUE)
+  }
```

The modes selected by RReliefF are stored in `res.rrelieff` as a combination of 7 object of 'QWMS' class.
Based on the modes selected by Stepwise Regression and the modes selected by RReliefF, the regression methods are used to establish the mapping relationship between the traffic volumes and the modes, and make predictions. In addition, we will compare the performance of different modes combinations in the modeling and prediction of the traffic volumes.
There are three linear regression algorithms in the **QWDAP** package, namely Stepwise Regression, PCR, and PLSR. The selected modes of Stepwise Regression and RReliefF are used for modeling and prediction, and 720 records are divided into 570 training samples and 150 test samples.
The function `qwdap.swr()` in the **QWDAP** package is used to establish linear relationships between the modes and traffic volumes by Stepwise Regression and the function `qwdap.predict()` are used to make predictions based on the established model. Take the dataset `train.data` as an example, the following operations are performed to model and prediction.

```
R> swr_sws_models <- list()
R> swr_sws_prds <- list()
R> for(i in c(1:7)){
+    swr_sws_models[[i]] <- qwdap.swr(in_data = res.sws[[i]],
+                                     data_range = c(1, 570), plotting = TRUE)
+    swr_sws_prds[[i]] <- qwdap.predict(in_model = swr_sws_models[[i]],
+                                       data_range = c(571, 720))
+  }
```

In the parameters of `qwdap.swr()`, the first parameter is a object of class 'QWMS', and the second parameter is the index range of the modes generated by quantum walks. The function `qwdap.predict()` is used for series prediction, and the parameter `data_range` is used to pass in the index range of the modes corresponding to the part to be predicted.
The result of regression analysis includes the fitted series and other operating parameters.

```
R> summary(swr_sws_models[[1]]$model)
```

```
...
Residual standard error: 7.884 on 545 degrees of freedom
Multiple R-squared:  0.623,        Adjusted R-squared:  0.6064
F-statistic: 37.53 on 24 and 545 DF,  p-value: < 2.2e-16
```

Now use PCR for modeling and series prediction, function `qwdap.pcr()` for regression analysis, and function `qwdap.predict()` for prediction.

```
R> pcr_sws_models <- list()
R> pcr_sws_prds <- list()
R> for(i in c(1:7)){
+    pcr_sws_models[[i]] <- qwdap.pcr(in_data = res.sws[[i]],
```

```
+                                        data_range = c(1,570), plotting = TRUE)
+    pcr_sws_prds[[i]] <- qwdap.predict(in_model = pcr_sws_models[[i]],
+                                        data_range = c(571,720))
+  }
```

The model built by the function `qwdap.pcr()` includes the fitted series with different numbers of principal components.

```
R> summary(pcr_sws_models[[1]]$model)

Data:         X dimension: 570 24
Y dimension: 570 1
Fit method: svdpc
Number of components considered: 24
...
```

The operation of using PLSR for modeling and prediction is consistent with PCR. The function `qwdap.plsr()` provides PLSR regression operation, and the prediction uses the function `qwdap.predict()`.

```
R> plsr_sws_models <- list()
R> plsr_sws_prds <- list()
R> for(i in c(1:7)){
+    plsr_sws_models[[i]] <- qwdap.plsr(in_data = res.sws[[i]],
+                                        data_range = c(1,570), plotting = TRUE)
+    plsr_sws_prds[[i]] <- qwdap.predict(in_model = plsr_sws_models[[i]],
+                                         data_range = c(571,720))
+  }
```

The above three kinds of linear regression methods are used to model and predict with the selected modes of Stepwise Regression, and it is the same operation to use the modes selected by RReliefF to model and predict. Figure 7 and Figure 8 are the fitted results of modeling using modes selected by Stepwise Regression and RReliefF, respectively, as well as the predicted data based on the established models.

It is shown in Figure 7 and Figure 8 that the linear combination of the modes selected by Stepwise Regression can fit better with the traffic volumes, and the predicted results can also reflect the trends of traffic volumes. The fitted series of `N5` and `N7` based on the 50 modes selected by RReliefF do not fit well with the traffic volumes. The Stepwise Regression itself is a kind of linear regression, we guess that using the modes selected by Stepwise Regression to perform linear regression has a better result. The selection principle of RReliefF is different from that of Stepwise Regression. It is a feasible method to increase the accuracy of linear regression by increasing the number of selected modes.

The following operation uses PPR to model with the selected modes and make predictions based on the established model. The same samples are used as before. The function `qwdap.predict()` can be used to make predictions based on the model established by PPR.

```
R> ppr_sws_models <- list()
R> ppr_sws_prds <- list()
```

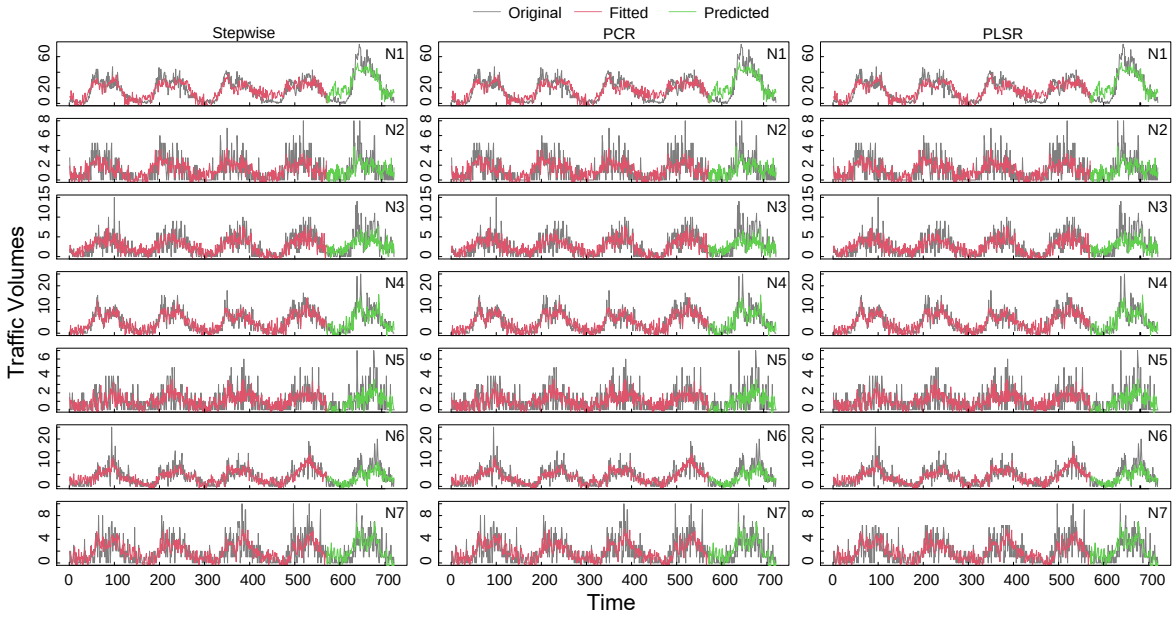

Figure 7: Linear regression and prediction results based on the modes selected by Stepwise Regression.

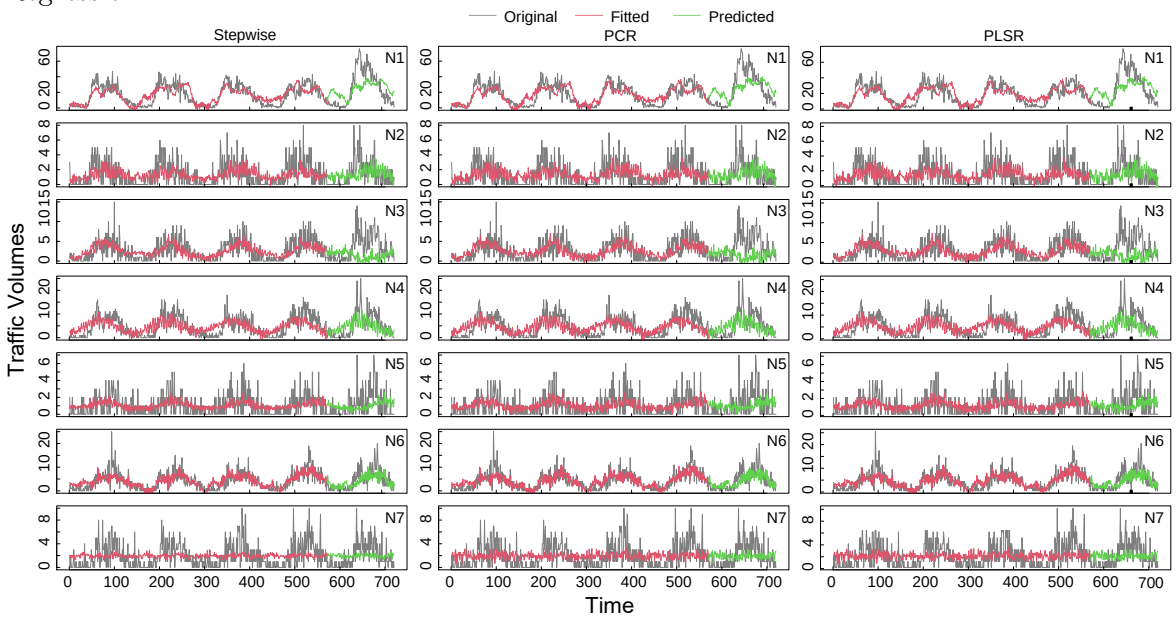

Figure 8: Linear regression and prediction results based on the modes selected by RReliefF.

```
R> for(i in c(1:7)){
+    ppr_sws_models[[i]] <- qwdap.ppr(in_data = res.sws[[i]],
+                                     data_range = c(1,570), plotting = TRUE)
+    ppr_sws_prds[[i]] <- qwdap.predict(in_model = ppr_sws_models[[i]],
+                                       data_range = c(571,720))
+  }
```

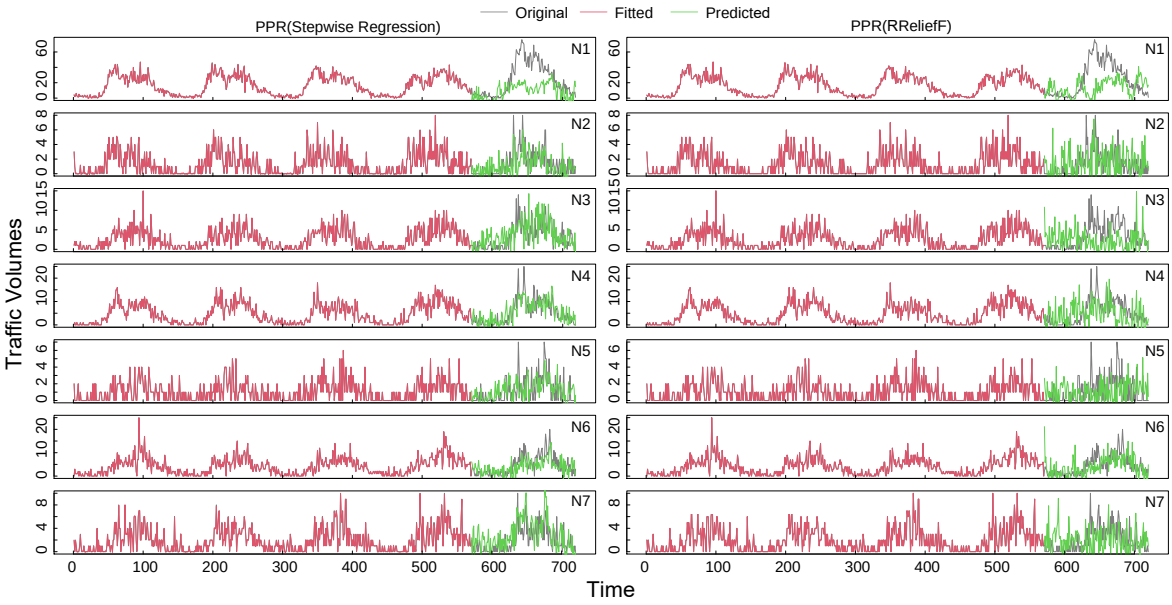

Figure 9: The results of fitting and prediction using PPR.

The results of PPR operation include the parameters of the projection direction and the parameters of the ridge function. The function `summary()` is used to get the direction parameter α of the projection and the parameter β of the ridge function.

```
R> summary(ppr_sws_models[[1]]$model)

...

Goodness of fit:
24 terms
223.7896

Projection direction vectors ('alpha'):
      term 1        term 2        term 3        term 4        term 5
V6     0.6055816697 -0.0596355952 -0.2917822469  0.1153903576  0.0344594461
...
      term 6        term 7        term 8        term 9        term 10
V6    -0.1634763049 -0.1087825134 -0.0846642900 -0.0004109335  0.3509088728
...
      term 11       term 12       term 13       term 14       term 15
V6    -0.1058751050  0.1829564111  0.1816424119  0.0757329398  0.0629799537
...
      term 16       term 17       term 18       term 19       term 20
V6    -0.2484176088 -0.1063379382  0.0754506041 -0.1777403764 -0.0248736187
...
      term 21       term 22       term 23       term 24
V6     0.1429607940  0.0300202071 -0.1547550542  0.3111212267
```

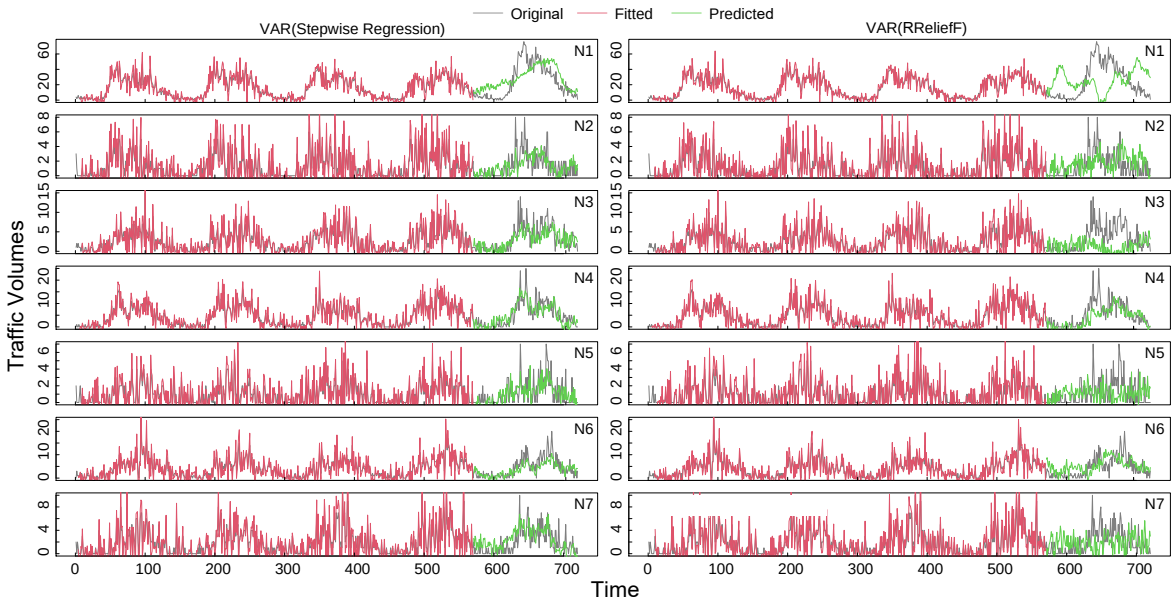

Figure 10: The results of fitting and prediction using VAR.

```
...

Coefficients of ridge terms ('beta'):
   term 1    term 2    term 3    term 4    term 5    term 6    term 7    term 8
10.622405  2.851330  2.923384  3.323855  3.247721  3.744133  1.976277  2.511969
   term 9   term 10   term 11   term 12   term 13   term 14   term 15   term 16
 3.342592  2.736404  2.623678  1.768621  3.058460  2.533658  2.517286  1.723760
  term 17   term 18   term 19   term 20   term 21   term 22   term 23   term 24
 2.160291  2.176958  2.502724  2.209129  1.743579  1.936702  1.328109  1.172130
```

The modes selected by Stepwise Regression and RReliefF are used for modeling and prediction, respectively, and the results are obtained as shown in Figure 9.
Using the selected modes of Stepwise Regression and RReliefF to model the traffic volumes and series prediction, PPR can get a better fitting effect than the linear regressions we used. In terms of series prediction, the overall effect of using Stepwise Regression for mode selection will be better. In terms of details, the fitted series based on the modes selected by RReliefF perform better.

The traffic volumes originally have temporal characteristics, which can also be used for regression and prediction of the series. VAR can model the traffic volumes with the time-domain characteristics of modes generated by quantum walks. The same experimental data as the linear and nonlinear regression are used next. The the `qwdap.var()` function is used to implement VAR operation. The function `qwdap.predict()` can be used to make predictions based on the model established by VAR.

```
R> var_sws_models <- list()
R> var_sws_prds <- list()
R> for(i in c(1:7)){
```

```
+    var_sws_models[[i]] <- qwdap.var(in_data = res.sws[[i]],
+                                     data_range = c(1,570), plotting = TRUE)
+    var_sws_prds[[i]] <- qwdap.predict(in_model = var_sws_models[[i]],
+                                       data_range = c(571,720))
+  }
```

In the VAR operation, a lag order needs to be set, and the lag order actually obtained by the AIC fitting is used by default. The fitted series is shorter than the original time series.
The modes selected by Stepwise Regression and RReliefF are used for modeling and prediction, respectively. The obtained results are shown in Figure 10.
Using the modes selected by Stepwise Regression and RReliefF to model the traffic volumes and perform series prediction, VAR can get a better effect than the linear regressions we used. Among the stations, the prediction results of the `N1` station are worse than those of the other stations, and the prediction results of other stations can obtain a relatively better fitting performance on the overall trend of the traffic volumes.

Result analysis

The Model Evaluation module can analyze the correlation between series in the time domain, and obtain some indexes that can reflect the fitted accuracy of the series. In this part, some evaluation indexes that reflect time-domain characteristics such as the R^2 and error indexes between the two series can be calculated.

Now we compare the results obtained by the two data combinations and the four regression methods from the quantitative calculation of the data. And then we use the function `qwdap.eval()` in the **QWDAP** package to directly obtain the Coefficient of Determination (R^2), the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE).

```
R> qwdap.eval(series1 = ppr_sws_models[[1]]$real[1:570],
+              series2 = ppr_sws_models[[1]]$model$fitted.values)

  Station Method        R2      RMSE       MAE
1                0.9975098 0.6265887 0.4862167
```

In general, the larger the R^2 is, the smaller the RMSE and the MAE are, which indicates greater correlation between the two series. However, because RMSE and MAE are affected by the mean of the series, RMSE and MAE cannot be used to compare results between stations, but can be used to compare results of different methods at the same station. Figure 11(1)(2)(3) are the data analyses of the fitted results of modeling using the modes selected by Stepwise Regression. Figure 11(4)(5)(6) show the accuracy of the fitted results of modeling using the modes selected by RReliefF. From R^2, the results of PPR fitted data in Figure 11(1)(4) are all 1, followed by VAR, with the highest accuracy of 0.91. There is little difference between Stepwise Regression and PLSR. Figure 12 shows the results of data analysis on the original time series and the fitted and predicted series. After adding the predicted series, the overall accuracy is reduced. As far as R^2 is concerned, PPR and VAR are generally more accurate than linear regression. For these 7 stations, the error indexes of the fitted and predicted results of the `N1` station is relatively larger because the average traffic volume at the `N1` station is higher. In terms of RMSE and MAE, when only simulation is considered, PPR has

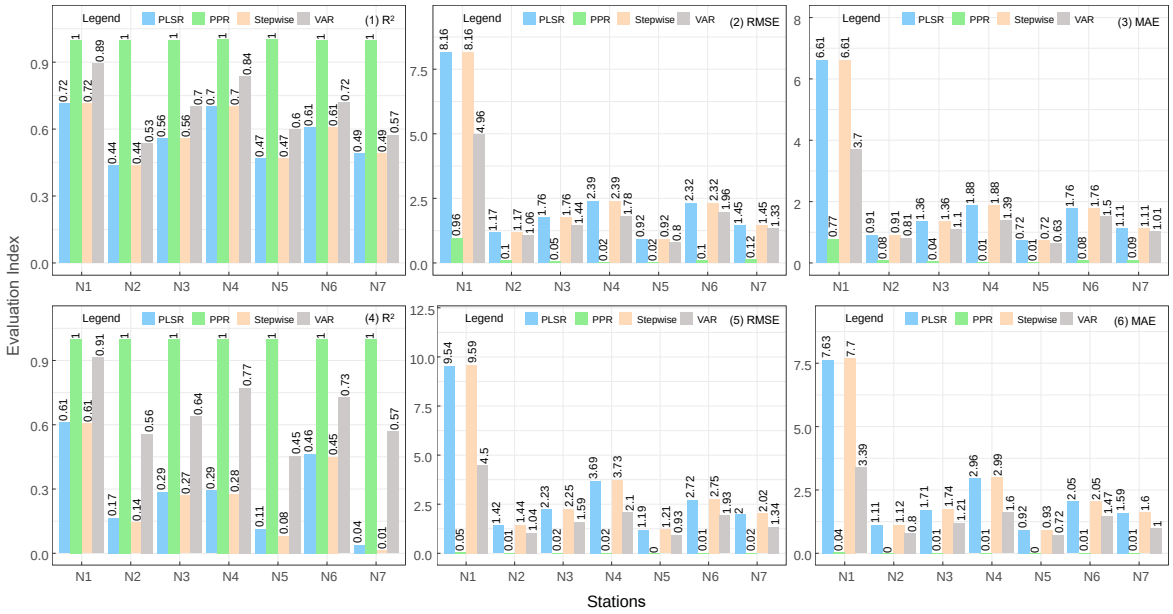

Figure 11: Statistical comparison of different regression methods.

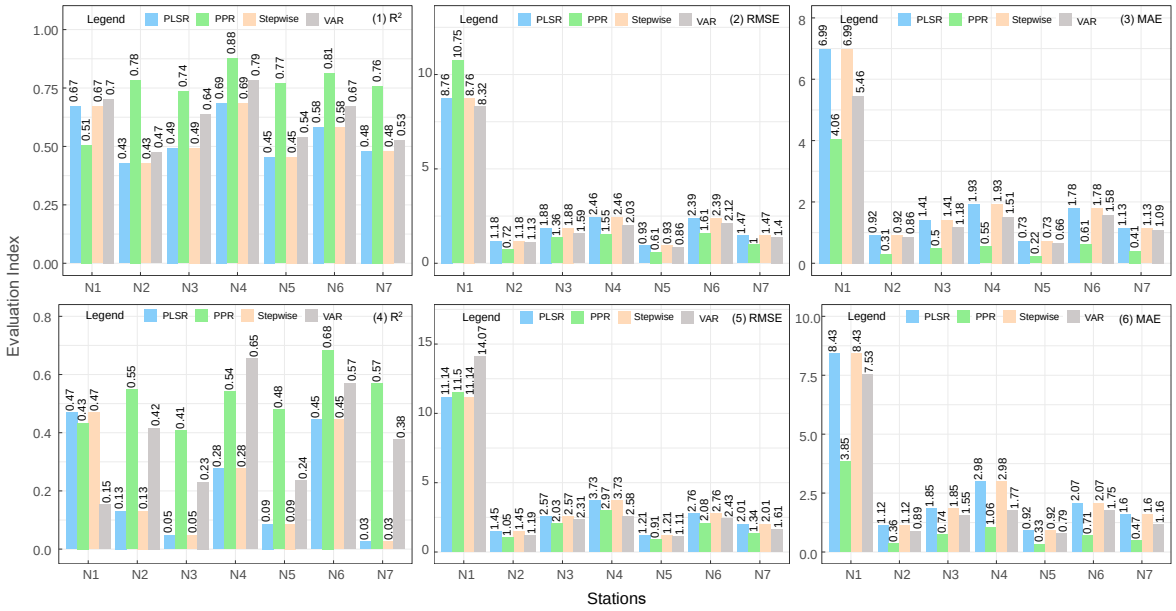

Figure 12: Statistical comparison of different regression and predict methods.

lower errors than other methods. In this paper, the results of mode selection are used for prediction and analysis. If more modes generated by quantum walks are used for analysis, higher accuracy may be achieved. However, using too many modes will cause overfitting. And from the comparison between Figure 11 and Figure 12, the modes selected by Stepwise Regression can achieve higher accuracy in fitting and prediction than the 50 modes selected by RReliefF.

5. Summary and Discussion

In this paper, we propose the **QWDAP** software package for data modeling and analysis based on quantum walk implemented by the R language. This package focuses on graph-associated time series analysis. Based on the traditional time series analysis, quantum walk provides features with graph correlation and this analysis method achieves high accuracy in the time series prediction of traffic volumes. In **QWDAP**, we implement three modules: Basis Generation, Data Modeling and Prediction, and Model Evaluation. Among these three modules: Basis Generation is used to produce multi-timescale modes, which are the probabilities generated by quantum walks on a graph from an initial state and scale factors, for data analysis. The Data Modeling and Prediction module can be used to select out the parts with characteristic correlation for a specific time series from the generated modes, and the regression methods are used to build models between the original time series and the modes, and predictions can be made based on these models. Model Evaluation evaluates the modeling effectiveness of the model in terms of the fitting accuracy of the fitted results of the model and the prediction results relative to the original time series.

As we use the quantum walk in the basis generation, the advantage of expression of multi-feature time series can be achieved. Quantum walk can be used as a graph-associated stochastic numerical simulation method that can correlate individual vertices to produce a set of spatially correlated modes. Quantum walks, which produce multi-timescale modes that can be used in the modeling and prediction of multi-time series. The modes generated by quantum walks do not require priori assumptions. Besides, the modes have irregular properties such as randomness, quasi-periodicity, and probability, and there is independence and orthogonality among the different modes, thus providing more possibilities for the expression and modeling of time series.

In **QWDAP**, two methods, model-driven Stepwise Regression and data-driven RReliefF are provided for mode selection. These two methods are able to extract the mode combinations with similar features from all the modes for a specific time series. As shown in the traffic volume simulation example, the modes selected by stepwise regression, with less modes numbers than 50, has better performances than using 50 modes selected by RRelifF. This is because the model-driven mode selection builds a linear model and adjusts the combination of modes by judging the evaluation index of the established model, so as to achieve the purpose of selecting the optimal linear combination of modes. Yet, the data-driven mode selection calculates the weight value of each mode relative to the original time series by methods such as nearest neighbor. A specified number of modes can be selected according to the weights. Compared to model-driven mode selection, data-driven mode selection focuses more on the correlation of a single mode with the original time series, while model-driven mode selection focuses on the representation of the mode combinations to the original time series. Model-driven mode selection is more suitable for modeling-oriented needs, while data-driven mode selection is more suitable for analyzing the correlation between a single mode and the original time series. For complex data modeling such as traffic volumes, the use of model-driven mode selection is preferred over data-driven mode selection.

In **QWDAP**, the relations between the original time series and the selected modes can be modeled from multiple perspectives. These perspectives can reveal the linear relationships, nonlinear relationships, and temporal-correlated relationships between the original time series and the selected modes. These three types of methods can be used to model the relationship between the original time series and the modes from different perspectives. In the example of

traffic volume simulation, linear regressions show large differences in the results of modeling using the two mode combinations. We supposed that data-driven mode selection methods can calculate the weights of individual modes relative to the original time series, but the modes with high weights do not necessarily contain all features of the original time series. In the case of modeling and prediction using the modes selected by Stepwise Regression, the prediction results using linear regression models are more stable than PPR or VAR. However, PPR and VAR can extract more features from the same mode combinations, for example, the fitted results using PPR or VAR contain more details compared to the linear regression results. From the results, nonlinear regression methods or methods based on temporal correlation can extract more features in the mode combinations, but models with more features may show greater bias in prediction. Since the modes generated by quantum walks are random data with some irregular properties, quantum walk-based modeling may yield unexpected results for irregular time series.

For the current **QWDAP** package, there is still great potential for development. In the algorithm simulation of quantum walks, a variety of mathematical simulation methods can be considered, and the structure of existing programs can be optimized to improve operating efficiency. In the application, two mode selection methods are selected from model-driven and data-driven. And some regression methods are used to apply the modes generated by quantum walks for data modeling and prediction, and it is found that the fitted results of PPR and VAR are better than that of linear regression. In the future work, more mode selection methods can be tried, such as selection based on the amount of information, hierarchical or segmented selection methods. And we will continue to study nonlinear regression and other regression ideas, add more regression and prediction methods, and apply quantum walks to more application fields. In terms of model evaluation, the current analysis mainly focuses on the correlation between series in the time domain, and the subsequent work can start from the expansion of the method and consideration of analyzing the correlation of the data from other aspects. In order to deal with the ever-increasing amount of analysis data, it is necessary to enhance the processing efficiency and processing stability of big data.

Computational details

The algorithms of the **QWDAP** package are all implemented by R language code, which can be obtained from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=QWDAP`. Some packages are used in the package, all of which can be obtained from CRAN. The implementation of the quantum walk simulation program refers to the **QuantumWalk** repository in github (`https://github.com/Evelios/QuantumWalk.git`). The CRAN **progress** (Csárdi and FitzJohn 2019) library is used in the quantum walk simulation program. The function `qwdap.sws()` of Stepwise Regression for mode selection uses the CRAN **StepReg** (Li, Lu, Cheng, and Liu 2021) library. The function `qwdap.rrelieff()` of RReliefF for mode selection uses the CRAN **COREelearn** (Robnik-Sikonja and Savicky 2021) library. The PCR-based function `qwdap.pcr()` and the PLSR-based function `qwdap.plsr()` use the CRAN **pls** (Mevik, Wehrens, and Liland 2020) library. The VAR-based function `qwdap.var()` and function `qwdap.predict()` for data prediction use the CRAN **MTS** (Tsay and Wood 2021) library.

The graphs shown in this article are made based on the result data of the **QWDAP** package with `plot()` function and **ggplot2** (Wickham 2016) package. **QWDAP** does not include the

code for making pictures in this article, and some operations in **QWDAP** have the option of making simple pictures.

References

Berry SD, Bourke P, Wang JB (2011). "qwviz: Visualisation of quantum walks on graphs." *Computer Physics Communications*, **182**(10), 2295–2302. ISSN 0010-4655. doi:10.1016/j.cpc.2011.06.002.

Bertrand F, Kane MJ, Emerson J, Weston S (2021). *'BLAS' and 'LAPACK' Routines for Native R Matrices and 'big.matrix' Objects*. R package version 1.0.1, URL https://fbertran.github.io/bigalgebra/.

Biamonte J, Faccin M, De Domenico M (2019). "Complex networks from classical to quantum." *Communications Physics*, **2**(1), 1–10. ISSN 2399-3650. doi:10.1038/s42005-019-0152-6.

Burkholder TJ, Lieber RL (1996). "Stepwise regression is an alternative to splines for fitting noisy data." *Journal of biomechanics*, **29**(2), 235–238. ISSN 0021-9290. doi:10.1016/0021-9290(95)00044-5.

Childs AM (2009). "Universal computation by quantum walk." *Physical review letters*, **102**(18), 180501. ISSN 0031-9007. doi:10.1103/PhysRevLett.102.180501.

Childs AM (2010). "On the relationship between continuous-and discrete-time quantum walk." *Communications in Mathematical Physics*, **294**(2), 581–603. ISSN 0010-3616. doi:10.1007/s00220-009-0930-1.

Csárdi G, FitzJohn R (2019). *progress: Terminal Progress Bars*. R package version 1.2.2, URL https://CRAN.R-project.org/package=progress.

Duersch JA, Gu M (2017). "Randomized QR with column pivoting." *SIAM Journal on Scientific Computing*, **39**(4), C263–C291. ISSN 1064-8275. doi:10.1137/15m1044680.

Eidelman Y, Gemignani L, Gohberg I (2008). "Efficient eigenvalue computation for quasiseparable Hermitian matrices under low rank perturbations." *Numerical Algorithms*, **47**(3), 253–273. ISSN 1017-1398. doi:10.1007/s11075-008-9172-0.

Falloon PE, Rodriguez J, Wang JB (2017). "QSWalk: a Mathematica package for quantum stochastic walks on arbitrary graphs." *Computer Physics Communications*, **217**, 162–170. ISSN 0010-4655. doi:10.1016/j.cpc.2017.03.014.

Farhi E, Gutmann S (1998). "Quantum computation and decision trees." *Physical Review A*, **58**(2), 915. ISSN 1050-2947. doi:10.1103/PhysRevA.58.915.

Gao J, Shang P (2019). "Analysis of complex time series based on EMD energy entropy plane." *Nonlinear Dynamics*, **96**(1), 465–482. ISSN 0924-090X. doi:10.1007/s11071-019-04800-5.

Glos A, Miszczak JA, Ostaszewski M (2019). "QSWalk. jl: Julia package for quantum stochastic walks analysis." *Computer Physics Communications*, **235**, 414–421. ISSN 0010-4655. doi:10.1016/j.cpc.2018.09.001.

Goebel R, Roebroeck A, Kim DS, Formisano E (2003). "Investigating directed cortical interactions in time-resolved fMRI data using vector autoregressive modeling and Granger causality mapping." *Magnetic resonance imaging*, **21**(10), 1251–1261. ISSN 0730-725X. doi:10.1016/j.mri.2003.08.026.

Hatifi M, Di Molfetta G, Debbasch F, Brachet M (2019). "Quantum walk hydrodynamics." *Scientific reports*, **9**(1), 1–7. ISSN 2045-2322. doi:10.1038/s41598-019-40059-x.

Helland IS, Sæbø S, Almøy T, Rimal R (2018). "Model and estimators for partial least squares regression." *Journal of Chemometrics*, **32**(9), e3044. ISSN 0886-9383. doi:10.1002/cem.3044.

Ieong II, Lou I, Ung WK, Mok KM (2015). "Using principle component regression, artificial neural network, and hybrid models for predicting phytoplankton abundance in Macau storage reservoir." *Environmental Modeling & Assessment*, **20**(4), 355–365. ISSN 1420-2026. doi:10.1007/s10666-014-9433-3.

Izaac J, Wang J (2017). "Systematic dimensionality reduction for continuous-time quantum walks of interacting fermions." *Physical Review E*, **96**(3), 032136. ISSN 2470-0045. doi:10.1103/PhysRevE.96.032136.

Izaac JA, Wang JB (2015). "pyCTQW: A continuous-time quantum walk simulator on distributed memory computers." *Computer Physics Communications*, **186**, 81–92. ISSN 0010-4655. doi:10.1016/j.cpc.2014.09.011.

Johansen S (2002). "A small sample correction for the test of cointegrating rank in the vector autoregressive model." *Econometrica*, **70**(5), 1929–1961. ISSN 0012-9682. doi:10.1111/1468-0262.00358.

Karski M, Förster L, Choi JM, Steffen A, Alt W, Meschede D, Widera A (2009). "Quantum walk in position space with single optically trapped atoms." *Science*, **325**(5937), 174–177. ISSN 0036-8075. doi:10.1126/science.1174436.

Kejani MT, Dornaika F, Talebi H (2020). "Graph Convolution Networks with manifold regularization for semi-supervised learning." *Neural Networks*, **127**, 160–167. ISSN 0893-6080. doi:10.1016/j.neunet.2020.04.016.

Kempe J (2003). "Quantum random walks: an introductory overview." *Contemporary Physics*, **44**(4), 307–327. ISSN 0010-7514. doi:10.1080/0010715103100011077 6.

Kitagawa T, Broome MA, Fedrizzi A, Rudner MS, Berg E, Kassal I, Aspuru-Guzik A, Demler E, White AG (2012). "Observation of topologically protected bound states in photonic quantum walks." *Nature communications*, **3**(1), 1–7. ISSN 2041-1723. doi:10.1038/ncomms1872.

Konno N (2019). "A new time-series model based on quantum walk." *Quantum Studies: Mathematics and Foundations*, **6**(1), 61–72. ISSN 2196-5609. doi:10.1007/s40509-018-0162-1.

Li J, Lu X, Cheng K, Liu W (2021). *StepReg: Stepwise Regression Analysis*. R package version 1.4.2, URL https://CRAN.R-project.org/package=StepReg.

Liu R, Kuang J, Gong Q, Hou X (2003). "Principal component regression analysis with SPSS." *Computer methods and programs in biomedicine*, **71**(2), 141–147. ISSN 0169-2607. doi:10.1016/s0169-2607(02)00058-5.

Lovett NB, Cooper S, Everitt M, Trevers M, Kendon V (2010). "Universal quantum computation using the discrete-time quantum walk." *Physical Review A*, **81**(4), 042330. ISSN 1050-2947. doi:10.1103/PhysRevA.81.042330.

Marquezino FL, Portugal R (2008). "The QWalk simulator of quantum walks." *Computer Physics Communications*, **179**(5), 359–369. ISSN 0010-4655. doi:10.1016/j.cpc.2008.02.019.

Matwiejew E, Wang J (2021). "QSW_MPI: A framework for parallel simulation of quantum stochastic walks." *Computer Physics Communications*, **260**, 107724. ISSN 0010-4655. doi:10.1016/j.cpc.2020.107724.

Mevik BH, Wehrens R (2007). "The pls package: principal component and partial least squares regression in R." *Journal of Statistical Software*, **18**(2), 1–23. ISSN 1548-7660. doi:10.18637/jss.v018.i02.

Mevik BH, Wehrens R, Liland KH (2020). *pls: Partial Least Squares and Principal Component Regression*. R package version 2.7-3, URL https://CRAN.R-project.org/package=pls.

Portugal R (2016). "Establishing the equivalence between Szegedy's and coined quantum walks using the staggered model." *Quantum Information Processing*, **15**(4), 1387–1409. ISSN 1570-0755. doi:10.1007/s11128-015-1230-7.

Qiang X, Loke T, Montanaro A, Aungskunsiri K, Zhou X, O'Brien JL, Wang JB, Matthews JC (2016). "Efficient quantum walk on a quantum processor." *Nature communications*, **7**(1), 1–6. ISSN 2041-1723. doi:10.1038/ncomms11511.

Rajeevan M, Pai D, Kumar RA, Lal B (2007). "New statistical models for long-range forecasting of southwest monsoon rainfall over India." *Climate Dynamics*, **28**(7-8), 813–828. ISSN 0930-7575. doi:10.1007/s00382-006-0197-6.

Robnik-Šikonja M, Kononenko I (1997). "An adaptation of Relief for attribute estimation in regression." In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, volume 5, pp. 296–304.

Robnik-Šikonja M, Kononenko I (2003). "Theoretical and empirical analysis of ReliefF and RReliefF." *Machine learning*, **53**(1), 23–69. ISSN 0885-6125. doi:10.1023/a:1025667309714.

Robnik-Sikonja M, Savicky P (2021). *CORElearn: Classification, Regression and Feature Evaluation*. R package version 1.56.0, URL https://CRAN.R-project.org/package=CORElearn.

Rossi L, Torsello A, Hancock ER (2015). "Measuring graph similarity through continuous-time quantum walks and the quantum Jensen-Shannon divergence." *Physical Review E*, **91**(2), 022815. ISSN 1539-3755. doi:10.1103/PhysRevE.91.022815.

Schreiber A, Gábris A, Rohde PP, Laiho K, Štefaňák M, Potoček V, Hamilton C, Jex I, Silberhorn C (2012). "A 2D quantum walk simulation of two-particle dynamics." *Science*, **336**(6077), 55–58. ISSN 0036-8075. doi:10.1126/science.1218448.

Sett A, Pan H, Falloon PE, Wang J (2019). "Zero transfer in continuous-time quantum walks." *Quantum Information Processing*, **18**(5), 1–18. ISSN 1570-0755. doi:10.1007/s11128-019-2267-9.

Song W, Wang L, Xiang Y, Zomaya AY (2017). "Geographic spatiotemporal big data correlation analysis via the Hilbert–Huang transformation." *Journal of Computer and System Sciences*, **89**, 130–141. ISSN 0022-0000. doi:10.1016/j.jcss.2017.05.010.

Steyerberg EW, Eijkemans MJ, Habbema JDF (1999). "Stepwise selection in small data sets: a simulation study of bias in logistic regression analysis." *Journal of clinical epidemiology*, **52**(10), 935–942. ISSN 0895-4356. doi:10.1016/s0895-4356(99)00103-1.

Tang H, Lin XF, Feng Z, Chen JY, Gao J, Sun K, Wang CY, Lai PC, Xu XY, Wang Y, *et al.* (2018). "Experimental two-dimensional quantum walk on a photonic chip." *Science advances*, **4**(5), eaat3174. ISSN 2375-2548. doi:10.1126/sciadv.aat3174.

Tsay RS, Wood D (2021). *MTS: All-Purpose Toolkit for Analyzing Multivariate Time Series (MTS) and Estimating Multivariate Volatility Models*. R package version 1.0.3, URL https://CRAN.R-project.org/package=MTS.

Tsuji Y, Estrada E, Movassagh R, Hoffmann R (2018). "Quantum interference, graphs, walks, and polynomials." *Chemical reviews*, **118**(10), 4887–4911. ISSN 0009-2665. doi:10.1021/acs.chemrev.7b00733.

Van Zee FG, Van De Geijn RA, Quintana-Ortí G, Elizondo GJ (2012). "Families of algorithms for reducing a matrix to condensed form." *ACM Transactions on Mathematical Software (TOMS)*, **39**(1), 1–32. ISSN 0098-3500. doi:10.1145/2382585.2382587.

Venegas-Andraca SE (2012). "Quantum walks: a comprehensive review." *Quantum Information Processing*, **11**(5), 1015–1106. ISSN 1570-0755. doi:10.1007/s11128-012-0432-5.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Yamashita T, Yamashita K, Kamimura R (2007). "A stepwise AIC method for variable selection in linear regression." *Communications in Statistics—Theory and Methods*, **36**(13), 2395–2403. ISSN 0361-0926. doi:10.1080/03610920701215639.

Zedda M, Singh R (2002). "Gas turbine engine and sensor fault diagnosis using optimization techniques." *Journal of propulsion and power*, **18**(5), 1019–1025. ISSN 0748-4658. doi:10.2514/2.6050.