A Novel Markov Model for Near-Term Railway Delay Prediction

Jin Xu¹, Weiqi Wang², Zheming Gao *,³, Haochen Luo⁴, and Qian Wu⁴

¹School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen, Guangdong 518172, China
²Department of Mathematics and Statistics, Concordia University, Montreal, QC, H3G 1M8, Canada
³College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China
⁴Department of Industrial & Systems Engineering, Texas A&M University, College Station, TX 77843, USA

Abstract

Predicting the near-future delay with accuracy for trains is momentous for railway operations and passengers’ traveling experience. This work aims to design prediction models for train delays based on Netherlands Railway data. We first develop a chi-square test to show that the delay evolution over stations follows a first-order Markov chain. We then propose a delay prediction model based on non-homogeneous Markov chains. To deal with the sparsity of the transition matrices of the Markov chains, we propose a novel matrix recovery approach that relies on Gaussian kernel density estimation. Our numerical tests show that this recovery approach outperforms other heuristic approaches in prediction accuracy. The Markov chain model we propose also shows to be better than other widely-used time series models with respect to both interpretability and prediction accuracy. Moreover, our proposed model does

*Corresponding author: gaozheming@ise.neu.edu.cn
†The first two authors contributed to the work equally and should be regarded as co-first authors.
not require a complicated training process, which is capable of handling large-scale forecasting problems.

Keywords: Railway delay prediction, Markov property test, non-homogeneous Markov chain, Gaussian kernel density estimate

1 Introduction

As one of the primary means of transport, railways provide freight shipments and passenger services that enable a huge number of goods and people to travel. Every year, about 10,000 billion freight tonne-kilometers and 3,000 billion passenger-kilometers are traveled via railways around the world (Railway Statistics 2015 Report, 2015). Punctuation is one of the most crucial measures to quantify the quality of railway operations and passengers’ traveling experience. A train is punctual if it arrives or departs at the planned time specified by the timetable. In the ideal scenario, the trains are operated punctually as the timetable. However, the railway operations would inevitably encounter disturbances, and the punctuation of the railway system could be determined by various factors, including the severe weather conditions, unexpected mechanical failure, drivers’ and travelers’ behavior, and temporary speed restrictions (Z. Li et al., 2021; Nabian et al., 2019; Olsson and Haugland, 2004). As shown in (Harris et al., 2013), the delay variations in Norway by different railway lines are significant, with the best-performing route achieving 94.4% and the worst routes achieving only near 80%, against the target of 90%. An accurate real-time delay predicting model thus would help the railway operators better coordinate and reschedule the trains, thereby improving the railway system reliability and reducing system operating costs. Moreover, announcing accurate delay information to passengers can better assist them in making travel plans, which improves customers’ traveling experience.

In this paper, we aim to develop a data-driven model that can predict the delay of a train in the near future. To accurately predict the delay is challenging for the following reasons. First, training a large number of customized models could be challenging. For the same train that travels through multiple stations, its delay at one station may be distinct from another. For different trains operating at different stations and regions, their delay processes could also be distinct. The notable heterogeneity of delay’s evolution across trains and stations makes it impractical to train a single unified model that predicts the delay for all the trains and stations. Therefore, building
a customized model for each train at a particular station is necessary. However, training a unique model for each train at each station could be challenging for a large area where many trains are operated. For instance, the Netherlands Railways data (The Institute for Operations Research and the Management Sciences, 2018) contains more than 6000 trains and 750000 stations. Even in the optimistic case where each station has only one train to travel through, more than 750000 models still need to be trained if we consider a unique model for each station.

Second, there might be no adequate records of delays in the historical data. For example, the worst route in (Harris et al., 2013) achieves an 80% punctuation rate, which implies that most of the historical data are still punctual. Moreover, the variety of delay causes makes each type of delay even less recorded. It is challenging to train an accurate prediction model with such small historical data of delays.

Third, some machine learning models are recently applied to delay prediction, such as the artificial neural networks (ANN) (P. Huang et al., 2020). Although these models may do well in predicting the delay, they may lack interpretability and cannot provide the insights of delays to the railway operators. In addition, these models usually require multiple data streams as input and many data to train. They would be less accurate when certain streams of data are not recorded, or when the training data are small. Moreover, these machine learning models usually require much manual intervention in hyper-parameter tuning. The long training and tuning processes make these models hard to implement in scenarios where a large number of models need to be built.

To overcome the challenges above, we propose a non-homogeneous Markov chain model to predict the delay over stations. The main contributions of our work are summarized as follows.

- **Markov chain modeling**: We build our prediction model by assuming the delay evolution over stations follows a non-homogeneous Markov chain. Each transition matrix in our model can characterize the unique pattern of delay evolution between two adjacent stations that a train travels through. The delay evolution over multiple stations can be decently captured by the Chapman-Kolmogorov equations of the Markov chain model.

- **Transition matrix recovery method**: We propose a Gaussian-kernel-based method to recover the transition matrices for the Markov chain when the training data are limited. This recovery method captures the delay transition probabilities from the existing data. We show that this recovery method achieves a higher prediction accuracy than the other heuristic approaches.
• Markov property test: We propose a chi-square Markov property test for the non-homogeneous Markov chain model when the transition matrix is sparse due to a lack of training data. The Markov property test verifies our assumption that the delay evolution over stations has a first-order Markov property. It strengthens the interpretability of our prediction model as well.

• Accuracy and lightweight: We conduct numerical experiments to verify the effectiveness and efficiency of the proposed model in delay prediction. The proposed model outperforms other commonly used time-series-based prediction models with respect to accuracy. Besides, only a series of transition matrices are necessary to be stored for each train in prediction. Recovering these matrices needs less computational power than other methods. Our model is thus suitable for forecasting the delays in a railway system with a large number of trains and stations. Moreover, our Markov chain model only relies on the delay data at each station for training. So it can be applied in scenarios where other factors related to train operation are not recorded.

We organize the rest of the paper as follows. We review the related work in Section 2 and describe the railway delay forecasting problem in Section 3. In Section 4, we provide a detailed description of the proposed Markov chain model. We test our model and compare it with other benchmarks in Section 5. We provide the conclusion and discuss our future research in Section 6.

2 Related Work

The models for traffic delay prediction have been investigated from different perspectives. Some research focuses on the relationship between railway delay and various factors in the railway systems. For instance, Olsson and Haugland (2004) analyze the correlation between train departure delay, number of passengers, and occupancy ratio using Norwegian railway data. Goverde (2005) adopts linear regression to explain the dependencies between train services with a transfer connection and the impact of the bottleneck in a particular station at Eindhoven. Flier et al. (2009) employ linear regression to analyze the delay dependencies on resource conflicts and maintained connections using the Swiss Railways data. Marković et al. (2015) investigate support vector regression (SVR) models that capture the relation between passenger train arrival delays and various characteristics of a railway system, such as the passenger train category, the scheduled time of arrival at the
station, the infrastructure influence, the percentage of the journey completed distance-wise, and the traveling distance. All these research studies mainly focus on the dependency between delay and existing system characteristics. They aim to understand the factors influencing railway delays so that to provide guidance to system operators and planners. Performing real-time delay forecasting is not the main focus of these studies.

Recently, machine learning approaches have been used for delay prediction. Lessan et al. (2019) propose a Bayesian-network-based train delay prediction model to characterize the complexity and dependency nature of different train operations. Yaghini et al. (2013) use ANN models to predict monthly averaged passenger train delays for Iranian railways. Real-time delay prediction is not the main focus of this paper. Oneto et al. (2018) propose shallow and deep extreme learning algorithms incorporating the types of the running day (whether it is a weekday or holiday), dwell times, and the running times for all the other trains running over the same section of the railway network during the day, to predict the delays in Italian railways. Nabian et al. (2019) propose a random forest model by incorporating many features of the railway system such as distance, number of stops, composite change, and driver change. These models rely on features other than the train’s historical delay information for training, and may not be applicable in scenarios where these additional features are not recorded. Moreover, training these models could be time-consuming, especially when the number of trains and stations is large.

Some statistical time series models that only rely on historical delay information have been applied to traffic forecasting. Suwardo et al. (2010) employ the Autoregressive Integrated Moving Average (ARIMA) model to predict bus travel time on the Ipoh-Lumut corridor in Malaysia. Lippi et al. (2013) compare the statistical time series models with the learning-based models on short-term traffic flow forecasting based on data collected from the California Freeway Performance Measurement System. The seasonal ARIMA (SARIMA) model with Kalman filter turns out to be the most accurate one. Although the ARIMA models are applicable in many scenarios, they may not perform well in railway delay forecasting where the delay evolution over stations is significantly heterogeneous and complex.

Recently, hybrid statistical time series and machine learning models have been proposed for traffic forecasting. Zhang (2003) proposes a hybrid ARIMA and neural network model for time series forecasting. Ma et al. (2020) concatenate a fundamental neural network model with the ARIMA model for network-wide traffic forecasting. Ge et al. (2021) propose a hybrid ARIMA
model and fuzzy SVR for high-speed railway passenger traffic forecasting. However, these learning-based models may lack interpretability, and some statistical assumptions made in these models are hard to justify in practice.

As a model of a time-dependent system, the Markov chain describes a sequence of random events where each event depends only on the states attained in the previous several events. The order of a Markov chain specifies how many previous events that the current state depends on. Many real-world processes can be illustrated by first-order Markov chain models where the current state only depends on the state in the previous event, such as the wildlife migration (C.-C. Huang, 1977), the relative risk of dementia (Yu et al., 2010), and the flight booking process (van der Walt & Bean, 2022). The analytical results and the corresponding inferences of Markov chains are well studied in the literature (Aalen & Johansen, 1978; Anily & Federgruen, 1987; Dobrushin, 1956a, 1956b). However, it is still unknown if the delay evolution over stations for railway systems follows a Markov chain of a certain order.

In summary, there are very few railway delay forecasting models with accuracy, lightweight, and interpretability altogether. Although the Markov chain is used for prediction in multiple scenarios, it is still unclear if the Markov chain model can capture the delay evolution of railway systems. Therefore, in our study, we plan to investigate the Markov property of delays in railway systems and design prediction models that rely on Markov chains without training burdens.

3 Problem Description

Delay Prediction

Passenger train transportation is a very important mode of transport in the Netherlands. Over a million passengers in Netherlands travel by train every day. As the biggest passenger train operator, the Netherlands Railways operates almost 6,000 trains daily. Over the recent years, data regarding delays in the entire railway network have been collected and made available to the operators. These data can be potentially used to train delay prediction models useful in passenger information systems and dispatching centers. An accurate prediction for the delays of trains in the near future can be announced to passengers through broadcasting or smartphone apps, which will be beneficial for passengers in adjusting their travel plans, hence improving their traveling experience. Moreover, an accurate estimate of the delay in the near future will enable the railway operation to make timely
dispatching adjustments and reschedules, especially if the delay increases or decreases dramatically.

The objective of this research work is to develop a data-driven prediction model that predicts the train’s delay trend (decrease, equal, or increase), the delay’s jump property (delay increases or decreases for more than two minutes), and the actual delay (in minutes) in the near future (e.g., 20 minutes). Moreover, we aim to develop a general real-time prediction model with low computational costs, so that it is capable to predict the delay for a railway system with a large number of trains operated.

Data Description

The data used for building our model is provided by ProRail, an infrastructure manager responsible for track maintenance and coordinating train operations (The Institute for Operations Research and the Management Sciences, 2018). The raw dataset contains railway operation history from September 4, 2017 to December 9, 2017, and it consists of a planned timetable and the realization data.

The planned timetable contains the planned arrival and departure times for each train at each station it travels through. The timetable is the same for Mondays, Tuesdays, Thursdays, and Fridays. On Wednesdays, extra trains operate on a busy part of the network between Eindhoven and Amsterdam. The timetables for weekends are not included due to altered operations on weekends.

The realization data contains the actual delay for each operated train during the recorded period. It includes the realized departure and arrival times (in seconds). Canceled trains are missing from the realization data. We will mainly rely on the realization data to train our prediction model.

Additional information, including the crew schedules, rolling stock circulation, infrastructure data, and weather conditions, are also provided in the dataset. However, our proposed model does not rely on these additional features. Thus our model is robust and can be deployed in many other systems when these additional variables are not recorded in history.

4 The Markov Chain Prediction Model

Before introducing our Markov chain prediction model, we first answer the questions of whether the delay evolution over stations follows a Markov chain and whether the Markov chain has order one. Answering these questions will help us understand how the delay at the earlier stations influences
that at later ones. Moreover, it will provide theoretical support in explaining why the Markov chain we propose works. Therefore, in this section, we first discuss the Markov property of the railway delays in Section 4.1. We then introduce the proposed non-homogeneous Markov chain prediction model in Section 4.2.1. In Section 4.2.2, we introduce the Gaussian kernel matrix recovery method in the circumstances where transition matrices for the Markov chain are sparse.

4.1 Markov Property for Railway Delays

In this subsection, we develop a chi-square test to verify that the delay evolution over stations for each train follows a first-order Markov chain. We first introduce the chi-square test for sparse transition matrices and then use the Netherlands Railways data to show that the Markov chain has order one.

4.1.1 Markov Property Test

We first introduce the concept of Markov property as follows. To facilitate our discussion, we now focus on a particular train that travels through multiple stations. We denote the delay at station $t$ by $D(t)$, where $D(t) \in [-N, N]$ is a bounded integer random variable. We denote the originated station as station 1, and the destination station as station $M$. Assume $d(1), d(2), \ldots, d(t)$ is a delay series (rounded in minutes) from station 1 to station $t$ ($t \leq M$). The delays evolution over stations satisfies a $k^{th}$ order Markov property if it satisfies

$$P(D(t) = d(t) | D(t-j) = d(t-j), \forall j = 1, \ldots, t-1) = P(D(t) = d(t) | D(t-j) = d(t-j), \forall j = 1, \ldots, k),$$

where $k = 1, \ldots, t-1$. In particular, the delays satisfy a zero order Markov property if

$$P(D(t) = d(t) | D(t-j) = d(t-j), \forall j = 1, \ldots, t-1) = P(D(t) = d(t)),$$

When the zero-order Markov property holds, $D(t)$ is a random variable independent of the random variables $D(t-1), \ldots, D(1)$, which means that the current delay is independent of the historical delays.

We aim to use a chi-square test to verify that the delay evolution over stations follows a first-
order Markov chain. The chi-square test will be performed on the historical data. Therefore, before
describing how the chi-square test works, we first introduce some necessary notations to describe
the values obtained from the historical data:

- \(n_j(t)\): the number of observations in the historical data that the train delays for \(j\) minutes at
station \(t\);
- \(n_{i,j}(t)\): the number of observations that the train delays for \(j\) minutes at station \(t\), and delays
for \(i\) minutes at station \(t - 1\);
- \(n_{h,i,j}(t)\): the number of observations that the train delays for \(j\) minutes at station \(t\), delays
for \(i\) minutes at station \(t - 1\), and delays for \(h\) minutes at station \(t - 2\).

Based on the number of observations defined above, we derive the following relations, which would
be useful for future analysis:

\[
\begin{align*}
n_j(t) &= \sum_{i=-N}^{N} n_{i,j}(t) = \sum_{h=-N}^{N} \sum_{i=-N}^{N} n_{h,i,j}(t), \\
n_{i}(t-1) &= \sum_{j=-N}^{N} n_{i,j}(t) = \sum_{h=-N}^{N} \sum_{j=-N}^{N} n_{h,i,j}(t), \\
n_{h}(t-2) &= \sum_{i=-N}^{N} \sum_{j=-N}^{N} n_{h,i,j}(t).
\end{align*}
\]

(3)

Using the defined terms in Equations (3), we now define the maximum likelihood estimate of the
state transition probabilities that will be used in chi-square test as follows:

- \(\hat{p}_j(t) = \frac{n_j(t)}{\sum_{i} n_{i,t}(t)}\): the frequency that the train delays for \(j\) minutes at station \(t\). The value
\(\hat{p}_j(t)\) is the \textit{maximal likelihood estimate} (MLE) of the probability \(p_j(t) = \mathbf{P}(D(t) = j)\).
- \(\hat{p}_{i,j}(t) = \frac{n_{i,j}(t)}{\sum_{i} n_{i,t}(t)}\): the frequency that the train delays for \(j\) minutes at station \(t\), given that
the train delays for \(i\) minutes at station \(t - 1\). The value \(\hat{p}_{i,j}(t)\) is the MLE of the probability
\(p_{i,j}(t) = \mathbf{P}(D(t) = j | D(t - 1) = i)\).
- \(\hat{p}_{h,i,j}(t) = \frac{n_{h,i,j}(t)}{\sum_{i} n_{h,i,t}(t)}\): the frequency that the train delays for \(j\) minutes at station \(t\), given
that the train delays for \(i\) minutes at station \(t - 1\) and \(h\) minutes at station \(t - 2\). The value
\(\hat{p}_{h,i,j}(t)\) is the MLE of the probability \(p_{h,i,j}(t) = \mathbf{P}(D(t) = j | D(t - 1) = i, D(t - 2) = h)\).

After introducing the necessary notations above, we now perform the Markov property test following
a similar procedure as (Bickenbach & Bode, 2003; Tan & Yilmaz, 2002). The general idea of
the testing procedure is that we test the Markov property from order zero, until a certain order of Markov property is accepted. However, it is important to note that the tests introduced in (Bickenbach & Bode, 2003; Tan & Yilmaz, 2002) are for homogeneous Markov chains, i.e., the values of $p_j(t)$, $p_{i,j}(t)$, and $p_{h,i,j}(t)$ do not vary for different stations $t$. We relax this assumption in our work by modeling the delay evolution as non-homogeneous Markov chains since the delay may have distinct forms of evolution over stations. There are multiple reasons for the delay to be non-homogeneous over stations. For instance, due to the delay, the train can be overtaken by a slower train at a certain track. As a result, this train cannot go faster than the slower train that is now in front, which causes the delay to accumulate (Lee et al., 2016). Moreover, the traveling distance between stations could be distinct. Two stations with a long distance in between may also have a large slack time in the timetable, and the slack time can be used to reduce the delay. The third reason could be the change of the rolling stock composition at certain stations (The Institute for Operations Research and the Management Sciences, 2018), which may result in delay distributions different from those stations without rolling stock change.

A challenge for testing the Markov property for non-homogeneous Markov chains is that the frequency is an approximation of the actual probability. If the probability is small, for example, if $p_{i,j} \ll 1$, it is quite likely to observe $n_{i,j} = 0$ from historical data. Then the obtained transition matrix of the Markov chain that indicates the probabilities of transiting from historical delays to the current delay may be sparse. To overcome this challenge, we propose a Markov property testing approach for the non-homogeneous Markov chain based on the likelihood ratio and chi-square statistic by removing the zero rows and columns of the transition matrices. The likelihood ratio and chi-square statistic are tested against a chi-square distribution whose degree of freedom relies only on the non-zero empirical probabilities calculated from the historical data.

We now test the null hypothesis that the Markov chain has zero order for a specific station $t$, i.e., $H_0^{(0)} : \{\forall i, j : p_{i,j}(t) = p_j(t)\}$. We define $A(t) = \{j : n_j(t) > 0\}$ as the index set of the delay minutes observed from the historical data. Similarly, we define $B_i(t) = \{j : n_{i,j}(t) > 0\}$ and $C_j(t) = \{i : n_{i,j}(t) > 0\}$. Considering all $\hat{p}_{i,j}(t)$ as parameters testing the null hypothesis, we then obtain the zero-order likelihood ratio (Koch, 1988) $LR^{(0)}(t)$ and chi-square statistic (Pearson, 1900) $Q^{(0)}(t)$ for hypothesis tests as follows:
\[ LR^{(0)}(t) = -2 \ln \prod_{j} \left( \hat{p}_j(t) \right)^{n_j(t)} = 2 \sum_{i,j: \hat{p}_{i,j}(t) \neq 0} n_{i,j}(t) \ln \frac{\hat{p}_{i,j}(t)}{\hat{p}_j(t)}, \quad (4) \]

\[ Q^{(0)}(t) = \sum_{i,j: \hat{p}_{i,j}(t) \neq 0} n_i(t-1) \frac{(\hat{p}_{i,j}(t) - \hat{p}_j(t))^2}{\hat{p}_j(t)}. \quad (5) \]

Then we analyze the degree of freedom for likelihood ratio and chi-square tests in the following lemma.

**Lemma 1.** Both \( LR^{(0)}(t) \) and \( Q^{(0)}(t) \) follow an asymptotic chi-square distribution with degree of freedom

\[ (|A(t-1)| - 1) (|A(t)| - 1). \quad (6) \]

**Proof.** It has been proven in Anderson and Goodman (1957) and Bickenbach and Bode (2003) that \( LR^{(0)}(t) \) and \( Q^{(0)}(t) \) follow the asymptotic chi-square distributions with identical degree of freedom. We thus consider the degree of freedom for the chi-square distribution focusing on \( Q^{(0)}(t) \).

Using a similar argument in Anderson and Goodman (1957), we can show that \( \sum_{j: \hat{p}_{i,j}(t) \neq 0} n_i(t-1) \frac{(\hat{p}_{i,j}(t) - \hat{p}_j(t))^2}{\hat{p}_j(t)} \) has an asymptotic chi-square distribution with a maximal degree of freedom \( |A(t)| - 1 \). To further derive the degree of freedom of \( Q^{(0)}(t) \), we only need to compute how many rows of \( i \) such that \( n_i(t) \neq 0 \). Here, we define the indicator function as

\[ 1_A = \begin{cases} 1 & \text{if condition A is true,} \\ 0 & \text{if condition A is false.} \end{cases} \]

Since \( |A(t)| = \sum_{j=-N}^{N} 1_{n_j(t) \neq 0} \), based on Equations (3), we have \( |A(t)| \geq \sum_{j=-N}^{N} 1_{n_i,j(t) \neq 0} \) for any \( i \in [-N, N] \). Now we compute the number of non-zero \( B_i \)s as follows:

\[
\sum_{i=-N}^{N} 1_{|B_i(t)| \neq 0} = \sum_{i=-N}^{N} 1_{\sum_{j=-N}^{N} n_{i,j}(t) \neq 0} \\
= \sum_{i=-N}^{N} 1_{n_i(t-1) \neq 0} \\
\leq |A(t-1)|. 
\]
Therefore, there is no more than $|\mathcal{A}(t - 1)|$ number of $B_i$. Moreover, since $\sum_{j=-N}^{N} p_j(t) = 1$, we should also subtract $(|\mathcal{A}(t)| - 1)$ number of degree of freedom from the summation. Therefore, the degree of freedom under $H_{0}^{(0)}$ is given by

$$|\mathcal{A}(t - 1)| (|\mathcal{A}(t)| - 1) - (|\mathcal{A}(t)| - 1) = (|\mathcal{A}(t - 1)| - 1) (|\mathcal{A}(t)| - 1).$$

Remark 2. In the scenarios where we do not have enough training data, it is likely that $n_{i,j}(t)$ has rows and columns with all the elements being zero. These zero rows and columns do not provide additional information in the test, thus are removed when we calculate the degree of freedom in Lemma 1. The degree of freedom given in Lemma 1 is thus the one we compute based on the truncated matrix $n_{i,j}(t)$. It is the actual degree of freedom for the matrix after removing all the zero rows and columns. Figure 1 provides a demonstrative graph of the truncated matrix.

Figure 1: Matrix truncation. The left matrix has zero rows and columns, and the right one is the truncated matrix after removing zero rows and columns.

If the zero-order hypothesis is rejected, we further test the hypothesis that the Markov chain has order one for a specific station $t$, i.e., $H_{0}^{(1)} : \{\forall h,i,j : p_{h,i,j}(t) = p_{i,j}(t)\}$. Similarly, we obtain the first-order likelihood ratio $LR^{(1)}(t)$ and chi-square statistic $Q^{(1)}(t)$ as follows:

$$LR^{(1)}(t) = 2 \sum_{h,i,j : \hat{p}_{h,i,j}(t) \neq 0} n_{h,i,j}(t) \ln \frac{\hat{p}_{h,i,j}(t)}{\hat{p}_{i,j}(t)}, \quad (7)$$

$$Q^{(1)}(t) = \sum_{h,i,j : \hat{p}_{h,i,j}(t) \neq 0} n_{h,i}(t-1) \left(\frac{\hat{p}_{h,i,j}(t) - \hat{p}_{i,j}(t)}{\hat{p}_{i,j}(t)}\right)^2. \quad (8)$$

Then we have the following lemma for the chi-square distribution that these two statistics test against.

**Lemma 3.** Both $LR^{(1)}(t)$ and $Q^{(1)}(t)$ follow the asymptotic chi-square distribution with degree of freedom

$$\left( |\mathcal{A}(t-2)| - 1 \right) \left( |\mathcal{A}(t-1)| - 1 \right).$$

(9)

**Proof.** We prove this lemma following a similar argument to the proof for Lemma 1 by focusing on the degree of freedom for $Q^{(1)}(t)$. For each fixed $h$ and $i$, we have $\sum_{j=-N}^{N} 1_{n_{h,i,j}(t) \neq 0} \leq |A(t)|$, the maximum degree of freedom is then $|A(t)| - 1$. We now consider the number of $h$ and $i$ such that $n_{h,i}(t-1) \neq 0$, then $n_{h,i,j}(t) \leq n_{h,i}(t-1)$. Summing over these $h$ and $i$, we have

$$\sum_{h=-N}^{N} \sum_{i=-N}^{N} 1_{n_{h,i,j}(t) \neq 0} \leq \sum_{h=-N}^{N} \sum_{i=-N}^{N} 1_{n_{h,i}(t-1) \neq 0}$$

$$\leq \sum_{h=-N}^{N} 1_{\sum_{i=-N}^{N} n_{h,i}(t-1) \neq 0} |A(t-1)|$$

$$\leq \sum_{h=-N}^{N} 1_{n_{h}(t-2) \neq 0} |A(t-1)|$$

$$\leq |A(t-2)||A(t-1)|.$$

Now the total degree of freedom is $|A(t-2)||A(t-1)|(|A(t)| - 1)$. We subtract it by the number of degree of freedom $|A(t-1)||A(t) - 1|$ we lose by imposing $\sum_{j} p_{i,j}(t-1) = 1$. Therefore, the degree of freedom under $H_0$ is given by

$$|A(t-2)||A(t-1)|(|A(t)| - 1) - |A(t-1)||A(t) - 1| = (|A(t-2)| - 1)|A(t-1)|(|A(t)| - 1).$$

Another way to prove the lemma is to follow a similar argument in Anderson and Goodman (1957): We consider the statistic

$$\chi^2_i = \sum_{h,j; \hat{p}_{h,i,j}(t) \neq 0} n_{h,i}(t-1) \frac{(\hat{p}_{h,i,j}(t) - \hat{p}_{i,j}(t))^2}{\hat{p}_{i,j}(t)}.$$

The statistic $\chi^2_i$ has the degree of freedom $(|A(t-2)| - 1)(|A(t)| - 1)$ due to $\sum_{j} \hat{p}_{h,i,j}(t) = 1$ and $\sum_{j} \hat{p}_{i,j}(t) = 1$. Therefore, $Q^{(1)}(t) = \sum_{i} \sum_{j; \hat{p}_{h,i,j}(t) \neq 0} \chi^2_i$ has the degree of freedom $(|A(t-2)| - 1)(|A(t)| - 1)$. [10]
2)|1 − 1)|A(t − 1)|(|A(t)| − 1). Hence proved.

If the first-order hypothesis $H_0^{(1)}$ is rejected, we can then test the next null hypothesis that the Markov chain has order two. For conciseness, we do not provide the details here since the railway delay can be captured by a first-order Markov chain, as we will show in the numerical study later. Algorithm 1 summarizes the detailed procedure for the Markov property test, where we use the chi-square statistics as the testing statistics.

**Algorithm 1** Markov Property Test

1: Obtain the number of observations $n_j(t)$, $n_{i,j}(t)$, and $n_{h,i,j}(t)$ from historical data
2: Obtain the frequencies $\hat{p}_j(t) = \frac{n_j(t)}{\sum_l n_l(t)}$, $\hat{p}_{i,j}(t) = \frac{n_{i,j}(t)}{\sum_l n_{i,l}(t)}$, and $\hat{p}_{h,i,j}(t) = \frac{n_{h,i,j}(t)}{\sum_l n_{h,i,l}(t)}$.
3: Compute $Q^{(0)}(t) = \sum_{i,j: \hat{p}_{i,j}(t) \neq 0} n_i(t − 1)(\hat{p}_{i,j}(t) − \hat{p}_j(t))^2$.
4: Let $\alpha_1$ be the error probability for the zero order hypothesis $H_0^{(0)}$
5: if $Q^{(0)}(t) < \chi^2_{\alpha_1, df_0}$, where degree of freedom $df_0 = (|A(t−1)|−1)(|A(t)|−1)$ then
6: Do not reject $H_0^{(0)}$
7: else
8: Reject $H_0^{(0)}$
9: Compute $Q^{(1)}(t) = \sum_{h,i,j: \hat{p}_{h,i,j}(t) \neq 0} n_{h,i}(t − 1)(\hat{p}_{h,i,j}(t) − \hat{p}_{i,j}(t))^2$.
10: Let $\alpha_2$ be the error probability for the first order hypothesis $H_0^{(1)}$
11: if $Q^{(1)}(t) < \chi^2_{\alpha_2, df_1}$, where degree of freedom $df_1 = (|A(t−2)|−1)|A(t−1)|(|A(t)|−1).
    then
12: Do not reject $H_0^{(1)}$
13: else
14: Reject $H_0^{(1)}$
15: end if
16: end if

### 4.1.2 Numerical Results

We now conduct the numerical test to show that the delay evolution follows a first order Markov chain, based on all the trains that are scheduled to operate during 8:00-8:20 and 12:00-12:20 in the historical data. In our numerical test, we let $N = 15$ since only very few delays (0.58%) in the historical data are out of the range of $[-N,N]$ with $N = 15$.

Note that a train may have different activities in the same station in the historical data. For example, in the historical data, “V” denotes the departure, “D” denotes passage without stop, “A” denotes arrival, “KV” denotes the departure at a short stop, and “KA” denotes the arrival at a short stop. A train will have “V” and “A” (or “KV” and “KA”) at the same station. Since a train may
stop at a station longer than scheduled, train delays can be different at the arrival and departure epochs at the same station. Thus, in our analysis, we regard the delays at the arrival and departure at the same station as at two different stations.

We calculate the total number of stations that all the trains travel through during these two periods and the number of stations that reject the zero-order and first-order Markov test. We provide the test results for both likelihood ratio $LR$ and chi-square statistics $Q$ in Table 1. From Table 1 we find that both statistics reject the $H_0^{(0)}$ with a high frequency and reject $H_0^{(1)}$ with a low frequency. This result shows that the delay evolution over stations follows a first-order Markov chain.

### Table 1: Markov Property Test

| Time period | Total Stations | Statistics | Stations Reject $H_0^{(0)}$ | Stations Reject $H_0^{(1)}$ |
|-------------|----------------|------------|----------------------------|----------------------------|
| 8:00-8:20   | 1632           | $LR^{(1)}$ | 1468                       | 2                          |
|             |                | $Q^{(1)}$  | 1571                       | 3                          |
| 12:00-12:20 | 1861           | $LR^{(1)}$ | 1630                       | 1                          |
|             |                | $Q^{(1)}$  | 1784                       | 2                          |

4.2 Prediction Model

We now introduce how to use the Markov chain framework to predict delay distributions, and then discuss how to recover the transition matrices of the Markov chain from historical data.

4.2.1 The Non-homogeneous Markov Chain Framework

Since we have shown the delay evolution over stations follows a first-order Markov chain in Section 4.1, it is reasonable to utilize the first-order Markov chain model to predict the delay in the near future. We aim to predict the delay for trains in a future station, given the delay at the current station. Without loss of generality, we suppose that the train is currently located at station $S$ with delay $d(S)$ minutes. We assume that from the timetable, the train is supposed to arrive at station $T$ in the future. We aim to predict the delay $d(T)$ at station $T$.

We predict the delays $d(t)$ using a $1 \times (2N + 1)$ dimension probability vector $v(t)$, where the $i^{th}$ element of $v(t)$ (denoted as $v_i(t)$) represents the probability that the train is delayed for $i$ minutes at station $t$ in our prediction model. We then have $\sum_{i=-N}^{N} v_i(t) = 1$ since $v_i(t) = P(D(t) = i)$.  

15
The delay at the current station $d(0)$ is a given number, as we already know the current delay. So we have $v_{d(S)}(S) = 1$ and $v_i(S) = S$ for $i \neq d(S)$. The transition matrix from station $t - 1$ to station $t$ that incorporates all the transition probabilities is given by

$$P(t) = \begin{pmatrix}
    p_{N-N}(t) & p_{-N,-N+1}(t) & \ldots & p_{-N,N}(t) \\
    p_{N+1,-N}(t) & p_{N+1,-N+1}(t) & \ldots & p_{N+1,N}(t) \\
    \vdots & \vdots & \ddots & \vdots \\
    p_{N,-N}(t) & p_{N,-N+1}(t) & \ldots & p_{N,N}(t)
\end{pmatrix}.$$ 

The delay distribution $v(S+1)$ at station $S+1$ is then given as $v(S+1) = v(S)P(S+1)$. By induction, we have the Chapman-Kolmogorov equation

$$v(T) = v(S) \prod_{t=S+1}^{T} P(t).$$ 

We then obtain the probability distribution $v(T)$ for the delay at station $T$.

### 4.2.2 The Gaussian-Kernel Method for Transition Matrix Recovery

We now need to recover the transition matrix $P(t)$ from the historical data. As described in Section 4.1, the transition probability is recovered by the empirical probability $\hat{p}_{i,j}(t) = \frac{n_{i,j}(t)}{\sum_{l=-N}^{N} n_{i,l}(t)}$. However, when the historical data are limited, it is possible that for some $i$, we have $\sum_{l=-N}^{N} n_{i,l}(t) = n_i(t-1) = 0$. This scenario means that there is no observation from the historical data that the train is delayed for $i$ minutes at station $t-1$. In that way, if the delay at the current station $S$ is $i$ minutes and $n_i(S-1) = 0$, then it is impossible to predict the delay at station $S+1$. A heuristic approach to revolve this issue is to let $\hat{p}_{i,i}(t) = 1$ and $\hat{p}_{i,j}(t) = 0$ for $j \neq i$. This recovery approach is to assume that the delay does not change over stations, if we do not observe the delay value from historical data. This approach is robust in some cases. However, it does not utilize the statistical information of the observed data. We will show in Section 5 that this approach can have a poor performance in practice. In this subsection, we propose a matrix recovery method that utilizes the existing observations. The general idea of the proposed matrix recovery approach is to utilize the Gaussian kernel density estimate to recover a two-dimensional distribution matrix and then normalize this matrix into a transition matrix.
To recover the two-dimensional matrix, we first consider the joint distribution for \( D(t-1) \) and \( D(t) \). We suppose \( f_{D(t-1), D(t)}(x, y) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \) is a two-dimensional probability density function for \( D(t) \) and \( D(t-1) \) that satisfies \( \int_{\mathbb{R}^2} f_{D(t-1), D(t)}(x, y) \, dx \, dy = 1 \). To estimate \( f_{D(t-1), D(t)}(x, y) \), we suppose \( \{x_1(t), x_2(t), ..., x_m(t)\} \) are \( m \) observations for \( (D(t-1), D(t)) \) from the historical data with mean \( \bar{x}(t) \). We then estimate the probability density function using the Gaussian kernel density estimate provided in (Silverman, 2017) as follows:

\[
\hat{f}_{t,h}(x) = \frac{1}{m h^2 |\Sigma|^\frac{1}{2}} \sum_{i=1}^{m} \Phi\left( h^{-2}(x - x_i(t))^T \Sigma^{-1}(x - x_i(t)) \right),
\]

where \( \Phi(u) = \frac{1}{2\pi} e^{-\frac{u^2}{2}} \), \( \Sigma = \frac{1}{m-1} \sum_{i=1}^{m} (x_i(t) - \bar{x}(t))(x_i(t) - \bar{x}(t))^T \) is the covariance matrix, and \( h \) is a hyper-parameter called window width that determines the fitting smoothness. Since we are fitting a two-dimensional distribution, the optimal window width is chosen as \( h = m^{-1/6} \) (Silverman, 2017).

However, if \( D(t-1) \) and \( D(t) \) are in perfect correlation (the covariance between two random variables equals 1), the covariance matrix \( \Sigma \) is singular. \( D(t-1) \) and \( D(t) \) are shown to be perfectly correlated when delay change is zero or constant. This phenomenon frequently occurs when the size of the historical data set is small. To resolve this issue, we may add an i.i.d. disturbance to the original data, e.g., we let \( \{\hat{x}_1(t), \hat{x}_2(t), ..., \hat{x}_m(t)\} = \{x_1(t) + \epsilon_1, x_2(t) + \epsilon_2, ..., x_m(t) + \epsilon_m\} \) to be the modified data, where each \( \epsilon_i \) is an independent two-dimensional disturbance randomly chosen within \([-\epsilon, \epsilon] \times [-\epsilon, \epsilon]\) for a small \( \epsilon \). We can thus obtain a non-singular covariance matrix using the modified data.

We now recover the transition matrix using the fitted Gaussian kernel density estimate. Specifically, we denote \( \hat{P}(t) \) as the recovered transition matrix for station \( t \), and each element \( \hat{p}_{i,j}(t) \) within \( \hat{P}(t) \) is given as follows:

\[
\hat{p}_{i,j}(t) = \frac{\hat{f}_{t,h}((i, j))}{\sum_{k=-N}^{N} \hat{f}_{t,h}((i, k))}.
\]

We can then predict the delay at station \( T \) by substituting \( P(t) \) with \( \hat{P}(t) \) in Equation (10). We summarize our prediction model in Algorithm 2.
Algorithm 2 The Markov Chain Prediction Model with Gaussian Kernel

1: Initialization: Given historical data \((x_1(t), x_2(t), ..., x_m(t))\) with \(t = S + 1, ..., T\) and delay \(d(S)\) at station \(S\)

2: Let \(v(S) = (0, ..., 1, ..., 0)\).

3: for \(t\) from \(S + 1\) to \(T\) do

4: function \(\hat{f}_{t,h}(x)\)

5: Randomly generate \(m\) two-dimensional small perturbations \((\varepsilon_1, \varepsilon_2, ..., \varepsilon_m)\) from uniform distribution on \([-\epsilon, \epsilon] \times [-\epsilon, \epsilon]\).

6: Modified data \((\tilde{x}_1(t), \tilde{x}_2(t), ..., \tilde{x}_m(t)) = (x_1(t) + \varepsilon_1, x_2(t) + \varepsilon_2, ..., x_m(t) + \varepsilon_m)\).

7: Data average \(\bar{x}(t) = \frac{1}{m} \sum_{i=1}^{m} \tilde{x}_i(t)\)

8: Let \(\Phi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}\), \(\Sigma = \frac{1}{m-1} \sum_{i=1}^{m} (\tilde{x}_i(t) - \bar{x}(t))(\tilde{x}_i(t) - \bar{x}(t))^T\), and \(h = m^{-1/6}\).

9: return \(\hat{f}_{t,h}(x) = \frac{1}{mh^2|\Sigma|^2} \sum_{i=1}^{m} \Phi\left(h^{-2}(x - \bar{x}_i(t))^T \Sigma^{-1}(x - \bar{x}_i(t))\right)\).

10: end function

11: Let \(\tilde{P}(t)\) be the transition matrix with \(\tilde{p}_{i,j}(t) = \frac{\hat{f}_{t,h}((i,j))}{\sum_{k \neq j} \hat{f}_{t,h}((i,k))}\).

12: Obtain the delay distribution \(v(t)\) at station \(t\) by \(v(t) = v(t-1)\tilde{P}(t)\)

13: end for

14: return \(v(T)\)

5 Prediction Results and Discussions

This section presents the numerical tests and discussions for the proposed model. We first introduce the performance measures in Section 5.1, and then discuss the metrics for prediction in Section 5.2. In Section 5.3, we investigate different methods for recovering the transition matrix. We then compare the proposed Markov chain model with other time series models in Section 5.4.

5.1 Performance Measures

We evaluate the performance of the prediction models by considering their capability to predict delays at the predicted station:

1. the train’s delay trend (decrease, equal, or increase, compared with the current delay);

2. whether there is a delay jump (i.e., the predicted delay increases or decreases for more than two minutes, compared with the current delay);

3. the minutes of delay.

We evaluate our model on all the \(M\) trains that operate during a randomly selected time window. The model will forecast delay trends, delay jumps, and minutes of delay for each train (we will
discuss in detail how these metrics are extracted from the delay distribution in Section 5.2). The model’s forecasting scores are calculated based on the prediction results of all the selected trains, as we shall show in the following.

**Delay Trend Prediction Score**

Before introducing the delay trend predictions score, we first define the following terms:

- **True Positive** ($TP_{IN}$): The number of trains whose delay is predicted to increase and the delay actually increases.
- **True Negative** ($TN_{IN}$): The number of trains whose delay is not predicted to increase, but the actual delay increases.
- **False Positive** ($FP_{IN}$): The number of trains whose delay is predicted to increase and the delay does not occur in reality.
- **False Negative** ($FN_{IN}$): The number of trains whose delay is not predicted to increase, but the actual delay increases.
- **Total number of trains**: $M = TP_{IN} + TN_{IN} + FP_{IN} + FN_{IN}$

We then define the positive predictive value for the increasing trend prediction ($PPV_{IN}$) as

$$PPV_{IN} = \frac{TP_{IN}}{TP_{IN} + FP_{IN}},$$

and the true positive rate for the increasing trend prediction ($TPR_{IN}$) as

$$TPR_{IN} = \frac{TP_{IN}}{TP_{IN} + FN_{IN}}.$$

We next evaluate the model’s prediction performance in increasing trend by considering the F1 score defined as

$$F_{IN} = \frac{2 \cdot PPV_{IN} \times TPR_{IN}}{PPV_{IN} + TPR_{IN}}.$$
The F1 score is ranged from 0 and 1, and a high F1 score indicates high classification performance (Tharwat, 2021).

Similarly, we can define the model’s F1 score for predicting delay decreasing $F_{DE}$ and predicting delay remaining equal $F_{EQ}$. We thus use the F1 score

$$F_{TR} = \frac{1}{3} (F_{IN} + F_{DE} + F_{EQ})$$

as the metric to measure the model’s performance in predicting delay trends.

**Delay Jump Prediction Score**

Like the F1 score defined for delay trend prediction, we use the F1 score to evaluate the prediction score for delay jump.

**Minutes of Delay Prediction Score**

We evaluate the prediction accuracy for minutes of delay using the root weighted mean square error (RWMSE). Denote $\hat{d}_i$ as the predicted delay and $d_i$ is the actual (realized) delay for train $i$. We let the weight for the absolute delays $|d_i|$ of 0 and 1 minute be 0.2, and weight for all the other delays be 0.8. Therefore, the RWMSE has a greater weight on large delays, which penalizes more if the large delays are not accurately predicted. We then define the root weighted mean square error as

$$RWMSE = \sqrt{\sum_{i=1}^{M} (w_1 1_{|d_i| \leq 1} + w_2 1_{|d_i| > 1}) |\hat{d}_i - d_i|},$$

where $w_1 = \sum_{i=1}^{M} \frac{0.2}{1_{|d_i| \leq 1}}$ and $w_2 = \sum_{i=1}^{M} \frac{0.8}{1_{|d_i| > 1}}$. It is easy to verify that

$$\sum_{i=1}^{M} (w_1 1_{|d_i| \leq 1} + w_2 1_{|d_i| > 1}) = \sum_{i=1}^{M} \left( \frac{0.2}{\sum_{i=1}^{M} 1_{|d_i| \leq 1}} 1_{|d_i| \leq 1} + \frac{0.8}{\sum_{i=1}^{M} 1_{|d_i| > 1}} 1_{|d_i| > 1} \right) = 1.$$

So that Equation (11) is a valid RWMSE.
**Total Prediction Score**

We use the total prediction score provided in The Institute for Operations Research and the Management Sciences (2018) to evaluate the model’s general prediction performance. The total prediction score is a linear combination of $F_{TR}$, $F_{JP}$, and RWMSE, which is given as

$$\text{Score} = 10F_{JP} + 5F_{TR} - \text{RWMSE}. \quad (12)$$

The total prediction score in Equation (12) values the delay jump prediction more than the trend prediction and RWMSE. The reason is that in reality, being unable to predict the drastic delay change may cause severe damage to both the railway system schedulers and passengers. Note that the performance of our proposed model is insensitive to the weights provided in Equation (12). As we will show later, our proposed model outperforms other models in each of the $F_{TR}$, $F_{JP}$, and RWMSE scores.

**Testing Data**

We evaluate the model performance based on two data sets from the historical data. *Test Set 1* contains 174 trains operated during 8:00-8:20, November 7, 2017. The mean of the actual delay at the predicted station in *Test Set 1* is 1.82183, and the variance is 8.30911. *Test Set 2* contains 222 trains operated during 12:00-12:20, November 9, 2017, with the mean of the actual delay at the predicted station in *Test Set 2* being 0.35211 and the variance being 0.96505. The delay in *Test Set 1* is more divergent than that in *Test Set 2*. All of our models are trained based on the historical data from September 4, 2017 to December 9, 2017, excluding these the data on these two testing dates.

**5.2 Delay Prediction Metrics**

Algorithm 2 introduced in Section 4.2 returns a prediction of distribution $v(T)$ for the delay value $D(T)$. We then compare the approaches to obtain the predicted delay trend, delay jump, and minutes of delay from the distribution $v(T)$.

We first define the mean, mode, and median for $D(T)$ as follows:

- **Mean:** $\text{Mean}(D(T)) = \sum_{i=-N}^{N} v_i(T) \cdot i$. 


• Mode: \( \text{Mode}(D(T)) = \arg \max_i \{ v_i(T) \} \).

• Median: \( \text{Median}(D(T)) = \min \{ i : \sum_{j=-N}^{i} v_i(T) \geq \frac{1}{2} \} \).

We also define the probability of delay increasing, decreasing, remaining equal, and delay jump as follows:

• Probability of delay increasing: \( P(D(T) > d(S) | D(S) = d(S)) = \sum_{i=d(S)+1}^{N} v_i(T) \).

• Probability of delay decreasing: \( P(D(T) < d(S) | D(S) = d(S)) = \sum_{j=-N}^{d(S)-1} v_i(T) \).

• Probability of delay remaining equal: \( P(D(T) = d(S) | D(S) = d(S)) = v_d(S)(T) \).

Moreover, we define the probability of delay jump as

\( P(|D(T) - d(S)| \geq 2 | D(S) = d(S)) = 1 - \sum_{i=d(S)-1}^{d(S)+1} v_i(t) \).

We now select the best one to predict the delay trend, delay jump, and minutes of delay based on these defined metrics.

**Delay Trend Prediction**

We now compare four approaches to predict the delay trend based on the mean value, mode, median, and probabilities. When using the mean value to predict the delay trend, we will do the following:

• Return “increase” if \( \text{Mean}(D(T)) - d(S) \geq 1 \); Return “decrease” if \( \text{Mean}(D(T)) - d(S) \leq -1 \); Return “equal” otherwise.

The way of using the mode and median to predict the delay trend is similar to that of using the mean. When using the increasing/decreasing/equal probability to predict delay trend, we do the following:

• Return “increase” if \( \sum_{i=d(S)+1}^{N} v_i(T) > \max \{ \sum_{i=-N}^{d(S)-1} v_i(T), v_d(S)(T) \} \); Return “decrease” if \( \sum_{i=-N}^{d(S)-1} v_i(T) > \max \{ \sum_{i=d(S)+1}^{N} v_i(T), v_d(S)(T) \} \); Return “equal” otherwise.

**Delay Jump Prediction**

Similar to the delay trend prediction, we compare four approaches to predict whether there is a delay jump based on the mean, mode, median, and probabilities. When using the mean to predict delay jump, we will
• Return “yes” if \(|\text{Mean}(D(T)) - d(S)| \geq 2\); Return “no” otherwise.

We will use the same criterion when using the mode and median to predict delay jump. When using the jump probability to predict delay jump, we will

• Return “yes” when the probability of delay jump \(1 - \sum_{i=d(S) + 1}^{d(S)+1} v_i(t) \geq 0.5\); Return “no” otherwise.

**Metrics Comparison**

For the minutes of delay prediction, we compare the performance of using mean, mode, and median at the predictor.

We present the performance score of using each metric in Table 2. For delay trend prediction, we find that using the mean, mode, median, and probability in prediction results in similar scores of \(F_{TR}\). Using the mode in prediction results in the lowest score for Test Set 1, and the highest score for Test Set 2. Using delay increasing/decreasing/equal probability to predict delay trend has the highest score for Test Set 1, but its performance for Test Set 2 ranks only the third. We thus choose the median to predict the delay trend, as its performance at both testing data sets ranks second among the four metrics.

We choose the jump probability to predict the delay jump, as its \(F_{JP}\) scores for both testing sets are much higher than the other metrics. Using the probability to predict the delay jump is also more reasonable than using the other metrics since both large delay increase and decrease are regarded as delay jumps. An example is that when the conditional delay distribution \(p_{i,j}(t)\) is Gaussian with a large tail, both large delay decrease and increase have a high probability, but the mean/mode/median value returns a delay minute in the middle, indicating no delay jump.

We choose the mean value to predict the minutes of delay since the mean value achieves the lowest RWMSE for both testing sets. The reason is that the mean value can better characterize the central tendency of the delay distribution, and the data with extremely large or small delays are rarely observed in history.

### 5.3 Comparison of Matrix Recovery Methods

In Section 4.2.2, we have proposed a Gaussian kernel method to recover the zero elements due to a lack of training data in the transition matrix of the Markov chain. We now compare the following
| Performance Measure | Test Data | Mean     | Mode     | Median   | Probability |
|---------------------|-----------|----------|----------|----------|-------------|
|                     | Test Set 1 | 0.56934  | 0.54502  | 0.57231  | **0.58312** |
|                     | Test Set 2 | 0.60353  | **0.73044** | 0.69949  | 0.66006     |
|                     | Test Set 1 | 0.48387  | 0.38596  | 0.45902  | **0.56716** |
|                     | Test Set 2 | 0.48276  | 0.59259  | 0.51852  | **0.62500** |
|                     | Test Set 1 | 2.88631  | 3.17746  | 2.96964  | N/A         |
|                     | Test Set 2 | 2.58319  | 2.81596  | 2.74579  | N/A         |

Table 2: Metrics Comparison

matrix recovery methods with the Gaussian kernel approach.

**Diagonal Filling**

Under the diagonal filling approach, we recover the transition matrix \( \tilde{P}(t) \) in the following way.

- If \( \sum_{l=-N}^{N} n_{i,l}(t) \neq 0 \), let \( \tilde{p}_{i,j}(t) = \frac{n_{i,j}(t)}{\sum_{l=-N}^{N} n_{i,l}(t)} \).
- If \( \sum_{l=-N}^{N} n_{i,l}(t) = 0 \), let \( \tilde{p}_{i,i}(t) = 1 \) and \( \tilde{p}_{i,j}(t) = 0 \) for \( j \neq i \).

The idea of the diagonal filling approach is that we use the frequency from the historical data to recover the transition probability. If there is no observation that the train is delayed for \( i \) minutes at station \( t-1 \), i.e., \( \sum_{l=-N}^{N} n_{i,l}(t) = 0 \), we assume that the delay at the station \( t \) is identical to the delay at the previous station \( t-1 \).

**Uniform Filling**

Under the uniform filling approach, we recover the transition matrix \( \tilde{P}(t) \) in the following way.

- If \( \sum_{l=-N}^{N} n_{i,l}(t) \neq 0 \), we let \( \tilde{p}_{i,j}(t) = \frac{n_{i,j}(t)}{\sum_{l=-N}^{N} n_{i,l}(t)} \).
- If \( \sum_{l=-N}^{N} n_{i,l}(t) = 0 \), we let \( \tilde{p}_{i,j}(t) = \frac{1}{2N+1} \) for \( j \in \{-N, \ldots, N\} \).

The idea of the uniform recovery is similar to that of the diagonal filling approach, as both of them will use the frequency to recover the transition probability. The difference is that when delay for \( i \) minutes at station \( t-1 \) is not observed from historical data, under the uniform filling approach, we assume that the delay at station \( t \) is uniformly distributed within \( \{-N, \ldots, N\} \).
A Gaussian Regression Recovery Approach

One phenomenon that we observe from the train delay data is that for each \( i \), the conditional probability \( p_{i,j}(t) \) is likely to be concentrated around \( p_{i,i}(t) \), i.e., \( p_{i,j}(t) \) is greater when \( j \) is close to \( i \). This is because the delay jump between two stations is quite rare, and the delay minutes are likely to be similar between two stations. So we can assume that the distribution \( p_{i,j}(t) \) follows a Gaussian distribution for each \( i \), and develop the Gaussian regression recovery approach as follows.

1. If \( \sum_{l=-N}^{N} n_{i,l}(t) \neq 0 \), we let \( \tilde{p}_{i,j}(t) = \frac{n_{i,j}(t)}{\sum_{l=-N}^{N} n_{i,l}(t)} \).

2. For \( i \) such that \( \sum_{l=-N}^{N} n_{i,l}(t) \neq 0 \), calculate the mean \( \hat{\mu}_i = \frac{\sum_{l=-N}^{N} n_{i,l}(t) \cdot l}{\sum_{l=-N}^{N} n_{i,l}(t)} \) and standard deviation \( \hat{\sigma}_i = \frac{\sum_{l=-N}^{N} n_{i,l}(t) \cdot (l - \hat{\mu}_i)^2}{\sum_{l=-N}^{N} n_{i,l}(t) - 1} \) for the fitted Gaussian distribution.

3. Perform a linear regression based on the fitted mean \( \hat{\mu}_i \) and standard deviation \( \hat{\sigma}_i \). Obtain two regressed linear functions

\[
\mu_i = \alpha + \beta \cdot i,
\]

and

\[
\sigma_i = \bar{\alpha} + \bar{\beta} \cdot i,
\]

where \( \mu_i \) and \( \sigma_i \) are the mean and standard deviation for the \( i^{th} \) row of \( \tilde{P}(t) \), and \( \alpha, \beta, \bar{\alpha}, \) and \( \bar{\beta} \) are fitted parameters.

4. For \( i \) such that \( \sum_{l=-N}^{N} n_{i,l}(t) = 0 \), let

\[
\hat{\mu}_i = \alpha + \beta \cdot i,
\]

and

\[
\hat{\sigma}_i = \bar{\alpha} + \bar{\beta} \cdot i.
\]

Then let \( \tilde{p}_{i,j}(t) = \frac{g_{\hat{\mu}_i, \hat{\sigma}_i}(j)}{\sum_{l=-N}^{N} g_{\hat{\mu}_i, \hat{\sigma}_i}(l)} \), where \( g_{\mu,\sigma}(x) \) is a one-dimension probability density function of the Gaussian distribution with mean \( \mu \) and standard deviation \( \sigma \).
Comparison and Discussion

We now compare the matrix recovery approaches discussed above with the Gaussian kernel approach provided in Section 4.2.2. We present the Gaussian kernel approach results in Figure 2, and Figure 3 further provides the results for the matrix recovery methods mentioned above. Both Figures 2 and 3 are based on the train with the number “519” and station name “Bl”.

Figure 2(a) presents the original transition matrix that we obtained by simply letting $\hat{p}_j(t) = \frac{n_j(t)}{\sum_l n_l(t)}$ if $\sum_l n_l(t) \neq 0$. We find that many rows of the transition matrix are zeros because the negative delay in this station was not observed. Figure 2(a) also shows that the recorded delays are likely to concentrate near the diagonal of the transition matrix. Only in a few cases do we see that the delay has a jump. For instance, the delay jumped from 9 minutes to 12 minutes with a probability 1 in the historical record.

Figure 2(b) plots the two-dimensional distribution density after the Gaussian kernel fitting. The delay density is also concentrated around the diagonal, showing that most historical delay values are small. Figure 2(c) provides the transition matrix recovered by the Gaussian kernel approach. We obtained Figure 2(c) by normalizing each row in Figure 2(b), so that Figure 2(c) is a valid transition matrix with all values within a row summing to 1. From Figure 2(c), we can see that the Gaussian kernel method recovers most features of the recorded values in Figure 2(a). For instance, the recovered transition matrix shows that the next delay is likely to stay unchanged when the current delay is small. Moreover, when the current delay is around 9 minutes, the recovered matrix indicates a high probability that the next delay will jump.

We plot the matrix by diagonal filling approach in Figure 3(a), and the one by uniform filling approach in Figure 3(b). These two approaches do not rely on the available data within the original transition matrix. We can expect their performance to degrade when the training data size becomes smaller.

We plot the matrix by the Gaussian regression recovery approach in Figure 3(c). We find that this approach does not retain the features of the original matrix. Using linear regression to fit the parameters $\mu_i$ and $\sigma_i$ can result in a large $\sigma$ on one side and a small one on the other, as we observe from Figure 3(c). When the current delay $i$ is small, the fitted $\sigma_i$ is large. Thus we see that the distribution is more dispersed. When the current delay is large, we even have negative fitted $\sigma_i$ values in the numerical study. For rows with negative fitted $\sigma_i$ values, we let its diagonal be 1. So eventually, this approach leads to a skewed transition matrix, as we see in Figure 3(c).
Figure 2: Gaussian Kernel Method
Figure 3: Comparison of Matrix Recovery Approaches
Table 3: Performance Comparison for Matrix Recovery Approaches

Table 3 provides the performance comparison for the matrix recovery approaches. Again, we select the median to predict the delay trend, the jump probability to predict the delay jump, and the mean to predict the minutes of delay for these matrix recovery approaches. We find from Table 3 that the Gaussian kernel method achieves the highest score for delay jump prediction, RWMSE, and total score, and a similar score in delay trend prediction as the other filling approaches. This result shows that the Gaussian kernel recovery approach outperforms the other approaches in terms of prediction accuracy. Moreover, this result shows that the good performance of the Gaussian kernel recovery approach is insensitive to the weights given in Equation (12).

5.4 Comparison with Other Time Series Models

We now compare our proposed model with other widely-used time series models.

Naive Prediction Approach

A naive forecasting approach is to assume the delay at station $T$ is identical to the current delay, i.e., $d(T) = d(S)$. This approach does not rely on historical data, and is simple to implement in reality.

Probability Distribution

We can predict the delay at station $T$ by simply using its delay distribution. Specifically, we compute $v(T) = \left( \frac{n_{-N}(T)}{\sum_{i=-N}^{N} n_i(T)}, \ldots, \frac{n_{N}(T)}{\sum_{i=-N}^{N} n_i(T)} \right)$ for station $T$. We then use the median to predict delay trend, use jump probability to predict delay jump, and delay expectation to predict the minutes of delay.
ARIMA Model

ARIMA model is a linear time series forecasting model that has been used in various forecasting fields, including economics, engineering, and geology (Cryer & Chan, 2008). Originated from the autoregressive (AR) model and the moving average (MA) model, the ARIMA model can be used when the time series is stationary and with no missing values (Ediger & Akar, 2007). An ARIMA model is usually written as $ARIMA(p, \beta, q)$, where $p$ is the order of the AR term that refers to the number of lags to be used as predictors. The parameter $q$ is the order of the MA term that refers to the number of lagged forecast errors that should go into the ARIMA model. The parameter $\beta$ is the minimum number of differencing needed to make the series stationary.

We implement a nonseasonal ARIMA model with the Python package “pmdarima” (Smith et al., 2017-) to predict the delay. We now use Test Set 1 as an example to demonstrate how we develop the ARIMA model. For each train, we use the delays recorded before 8:00 as the training set for the ARIMA model. Then we use the trained model to predict the delay at the predicted station. A detailed description of how to use ARIMA model to predict several time units into the future can be found in Cryer and Chan (2008).

We conduct a stepwise algorithm (Hyndman & Khandakar, 2008) to find the optimal model parameters, including the order of AR $p \in \{1, 2, 3\}$ and the order of MA $q \in \{1, 2, 3\}$. We rely on the Akaike’s Information Criterion (AIC) value in selecting the best orders of the ARIMA model. We further obtain the optimal value $\beta$ by conducting differencing tests.

Model Comparison

We compare the prediction scores of the time series model mentioned above in Table 4. Although the naive forecasting approach has the minimum RWMSE, its $F_{TR}$ and $F_{JP}$ scores are the worst since it does not utilize the historical data for training. Using probability distribution to predict delay can be promising when the delay is stable. For instance, its performance for Test Set 2 is close to the performance of our Markov chain model. However, for Test Set 2 whose delays are divergent and unstable, the performance of using probability distribution is much worse than our model. The reason is that the probability distribution is only obtained from historical data. This approach does not utilize the delays at the past stations on the particular predicted date.

The ARIMA model only uses the historical data on the predicted date for training without using
| Method                   | Test Data | $F_{TR}$ | $F_{JP}$ | RWMSE   | Total Score |
|--------------------------|-----------|----------|----------|---------|-------------|
| Naive Forecasting        | Test Set 1| 0.18156  | 0.00476  | 2.86511 | -1.95253    |
|                          | Test Set 2| 0.24182  | 0.00111  | 2.45632 | -1.23613    |
| Probability Distribution | Test Set 1| 0.43879  | 0.53488  | 4.00108 | 3.54168     |
|                          | Test Set 2| 0.64149  | 0.64849  | 2.58138 | 7.11094     |
| ARIMA Model              | Test Set 1| 0.39252  | 0.28000  | 3.44114 | 1.32148     |
|                          | Test Set 2| 0.67438  | 0.42424  | 2.60824 | 5.00610     |
| Our Model                | Test Set 1| 0.57231  | 0.56716  | 2.86311 | 5.64684     |
|                          | Test Set 2| 0.69949  | 0.62500  | 2.58319 | 7.16426     |

Table 4: Time Series Model Comparison

the other historical data. When the predicted train’s current station $S$ is not far from its starting station 1, the training set could be small so that not much pattern can be learned from history. We can find from Table 4 that the ARIMA model performs worse than the method of using probability distribution due to its training set being small.

The Markov chain model we proposed in this work has a better overall performance than the other models described above. The reason is that the Markov chain model is trained based on historical data and also takes the delay at the current station as the input. Our model has the largest score in $F_{TR}$. Its $F_{JP}$ score is the best for Test Set 1 where the delays are divergent. Its RWMSE scores are close to the minimal scores given by the naive forecasting approach. We can then conclude that the good performance of our model is not sensitive to the weights assigned in Equation (12).

## 6 Conclusion and Future Research

This research proposes a novel, accurate, and efficient model to predict railway delays using a Markov-chain-based framework. Theoretical properties of the proposed model are rigorously investigated, and insights are developed. Moreover, we conduct numerical experiments to verify the prediction accuracy and efficiency using the Netherlands Railways data. The major findings of this paper are summarized below:

- We propose a non-homogeneous Markov chain to characterize the delay process over stations.

  To test the order of the Markov chain, we propose and conduct a chi-square Markov property test with the Netherlands Railways data. The results show that the delays over stations of
the same train follow a first-order Markov chain.

- We develop a Gaussian-kernel-based method to recover the transition matrices for the Markov chain model when the size of training data is small. The Markov chain model equipped with the proposed recovery method achieves a higher prediction accuracy than being with other heuristic matrix recovery methods.

- We conduct numerical experiments on the real-world Netherlands Railways data and measure the prediction performance of each implemented model using a certain score calculated with the delay trend, the delay jump, and the root weighted mean square error. The proposed model provides a higher prediction score than other benchmark time series prediction models.

Our investigation of the Markov-chain-based delay prediction model leads to some potential research works. For instance, the strategies for train scheduling and passenger assignment can be improved when considering the potential railway delay (X. Li et al., 2020; Xu et al., 2019). Incorporating delay prediction into online railway scheduling and dispatching is one of our future research directions. Besides, we can migrate the proposed Markov property test, Markov chain model, and matrix recovery method to other application areas, such as health prognostics (Ghamlouch et al., 2018) and electric load prediction (Khashei & Chakhkoutahi, 2021).

References

Aalen, O. O., & Johansen, S. (1978). An empirical transition matrix for non-homogeneous Markov chains based on censored observations. Scandinavian Journal of Statistics, 5(3), 141–150. http://www.jstor.org/stable/4615704

Anderson, T. W., & Goodman, L. A. (1957). Statistical inference about Markov chains. The Annals of Mathematical Statistics, 28(1), 89–110. https://www.jstor.org/stable/2237025

Anily, S., & Federgruen, A. (1987). Ergodicity in parametric nonstationary Markov chains: An application to simulated annealing methods. Operations Research, 35(6), 867–874. https://pubsonline.informs.org/doi/abs/10.1287/opre.35.6.867

Bickenbach, F., & Bode, E. (2003). Evaluating the Markov property in studies of economic convergence. International Regional Science Review, 26(3), 363–392. https://doi.org/10.1177/0160017603253789

32
Cryer, J. D., & Chan, K.-S. (2008). *Time Series Analysis: With Applications in R*. Springer. https://link.springer.com/book/10.1007/978-0-387-75959-3#bibliographic-information

Dobrushin, R. L. (1956a). Central limit theorem for nonstationary Markov chains. I. *Theory of Probability & Its Applications*, 1(1), 65–80. http://mi.mathnet.ru/eng/tvp/v1/i1/p72

Dobrushin, R. L. (1956b). Central limit theorem for nonstationary Markov chains. II. *Theory of Probability & Its Applications*, 1(4), 329–383. http://mi.mathnet.ru/eng/tvp/v1/i4/p365

Ediger, V. Ş., & Akar, S. (2007). ARIMA forecasting of primary energy demand by fuel in Turkey. *Energy Policy*, 35(3), 1701–1708. https://www.sciencedirect.com/science/article/pii/S0301421506002291

Flier, H., Gelashvili, R., Graffagnino, T., & Nunkesser, M. (2009). Mining railway delay dependencies in large-scale real-world delay data. *Robust and online large-scale optimization* (pp. 354–368). Springer. https://link.springer.com/chapter/10.1007/978-3-642-05465-5_15

Ge, M., Junfeng, Z., Jinfei, W., Huiting, H., Xinghua, S., & Hongye, W. (2021). ARIMA-FSVR hybrid method for high-speed railway passenger traffic forecasting. *Mathematical Problems in Engineering*, 2021(9961324). https://www.hindawi.com/journals/mpe/2021/9961324/

Ghamlouch, H., Fouladirad, M., & Grall, A. (2018). Prognostics for non-monotonous health indicator data with jump diffusion process. *Computers & Industrial Engineering*, 126, 1–15. https://www.sciencedirect.com/science/article/pii/S036083521830425X

Goverde, R. M. (2005). Punctuality of railway operations and timetable stability analysis. *Doctoral thesis, The Netherlands TRAIL Research School*. http://resolver.tudelft.nl/uuid:a40ae4f1-1732-4bf3-bbf5-fdb8df635e7

Harris, N. G., Mjøsund, C. S., & Haugland, H. (2013). Improving railway performance in Norway. *Journal of Rail Transport Planning & Management*, 3(4), 172–180. https://www.sciencedirect.com/science/article/pii/S2210970614000134

Huang, C.-C. (1977). Non-homogeneous Markov chains and their applications. *Doctoral dissertation, Iowa State University*. https://dr.lib.iastate.edu/handle/20.500.12876/80511

Huang, P., Wen, C., Fu, L., Peng, Q., & Tang, Y. (2020). A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. *Information Sciences*, 516, 234–253. https://www.sciencedirect.com/science/article/pii/S0020025519311715
Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software, 27*(3), 1–22. https://www.jstatsoft.org/index.php/jss/article/view/v027i03

Khashei, M., & Chahkoutahi, F. (2021). A comprehensive low-risk and cost parallel hybrid method for electricity load forecasting. *Computers & Industrial Engineering, 155*, 107182. https://www.sciencedirect.com/science/article/pii/S0360835221000863

Koch, K.-R. (1988). Parameter estimation and hypothesis testing in linear models. Springer. https://link.springer.com/book/10.1007/978-3-662-03976-2

Lee, W.-H., Yen, L.-H., & Chou, C.-M. (2016). A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services. *Transportation Research Part C: Emerging Technologies, 73*, 49–64. https://www.sciencedirect.com/science/article/pii/S0968090X16302005

Lessan, J., Fu, L., & Wen, C. (2019). A hybrid Bayesian network model for predicting delays in train operations. *Computers & Industrial Engineering, 127*, 1214–1222.

Li, X., Li, D., Hu, X., Yan, Z., & Wang, Y. (2020). Optimizing train frequencies and train routing with simultaneous passenger assignment in high-speed railway network. *Computers & Industrial Engineering, 148*, 106650. https://www.sciencedirect.com/science/article/pii/S0360835220303843

Li, Z., Wen, C., Hu, R., Xu, C., Huang, P., & Jiang, X. (2021). Near-term train delay prediction in the Dutch railways network. *International Journal of Rail Transportation, 9*(6), 520–539. https://doi.org/10.1080/23248378.2020.1843194

Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems, 14*(2), 871–882. https://ieeexplore.ieee.org/document/6482260

Ma, T., Antoniou, C., & Toledo, T. (2020). Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. *Transportation Research Part C: Emerging Technologies, 111*, 352–372. https://www.sciencedirect.com/science/article/pii/S0968090X19303821

Marković, N., Milinković, S., Tikhonov, K. S., & Schonfeld, P. (2015). Analyzing passenger train arrival delays with support vector regression. *Transportation Research Part C: Emerg-
Nabian, M. A., Alemazkoor, N., & Meidani, H. (2019). Predicting near-term train schedule performance and delay using bi-level random forests. *Transportation Research Record, 2673*(5), 564–573. https://journals.sagepub.com/doi/abs/10.1177/0361198119840339

Olsson, N. O., & Haugland, H. (2004). Influencing factors on train punctuality—results from some Norwegian studies. *Transport Policy, 11*(4), 387–397. https://www.sciencedirect.com/science/article/pii/S0967070X04000332

Oneto, L., Fumeo, E., Clerico, G., Canepa, R., Papa, F., Dambra, C., Mazzino, N., & Anguita, D. (2018). Train delay prediction systems: a big data analytics perspective. *Big data research, 11*, 54–64. https://www.sciencedirect.com/science/article/pii/S2214579617300060

Pearson, K. (1900). X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 50*(302), 157–175. https://doi.org/10.1080/14786440009463897

Railway Statistics 2015 Report. (2015). International Union of Railways. UIC, 16 rue Jean Rey ,F 75015 Paris, France. https://uic.org/IMG/pdf/synopsis_2015_print_5_.pdf

Silverman, B. W. (2017). *Density estimation for statistics and data analysis*. Routledge. https://www.taylorfrancis.com/books/mono/10.1201/9781315140919/density-estimation-statistics-data-analysis-silverman

Smith, T. G. et al. (2017-). pmdarima: ARIMA estimators for Python [Online; accessed 2022-05-13]. http://www.alkaline-ml.com/pmdarima

Suwardo, W., Napiah, M., & Kamaruddin, I. (2010). ARIMA models for bus travel time prediction. *Journal of the Institute of Engineers Malaysia*, 49–58. https://www.semanticscholar.org/paper/ARIMA-MODELS-FOR-BUS-TRAVEL-TIME-PREDICTION-Suwardo-Madzlan/b2c16e5eb4efba0bd78ca3f93eb1a13f3f7677

Tan, B., & Yilmaz, K. (2002). Markov chain test for time dependence and homogeneity: an analytical and empirical evaluation. *European Journal of Operational Research, 137*(3), 524–543. https://ideas.repec.org/a/eee/ejores/v137y2002i3p524-543.html
Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics, 17*(1), 168–192. https://www.emerald.com/insight/content/doi/10.1016/j.aci.2018.08.003/full/html

The Institute for Operations Research and the Management Sciences. (2018). 2018 RAS Problem Solving Competition: Train Delay Forecasting. https://connect.informs.org/railway-applications/new-item3/problem-solving-competition681

van der Walt, A., & Bean, W. L. (2022). Inventory management for the in-flight catering industry: A case of uncertain demand and product substitutability. *Computers & Industrial Engineering, 165*, 107914. https://www.sciencedirect.com/science/article/pii/S0360835221008184

Xu, X., Li, K., & Lu, X. (2019). Simultaneous locomotive assignment and train scheduling on a single-track railway line: A simulation-based optimization approach. *Computers & Industrial Engineering, 127*, 1336–1351. https://www.sciencedirect.com/science/article/pii/S0360835217305223

Yaghini, M., Khoshraftar, M. M., & Seyedabadi, M. (2013). Railway passenger train delay prediction via neural network model. *Journal of Advanced Transportation, 47*(3), 355–368. https://doi.org/10.1002/atr.193

Yu, L., Griffith, W. S., Tyas, S. L., Snowdon, D. A., & Kryscio, R. J. (2010). A nonstationary Markov transition model for computing the relative risk of dementia before death. *Statistics in medicine, 29*(6), 639–648. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2830381/

Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159–175. https://www.sciencedirect.com/science/article/pii/S0925231201007020

36