Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Controlling highway toll stations using deep learning, queuing theory, and differential evolution

Andrija Petrović [a,*], Mladen Nikolić [b], Uglješa Bugarić [c], Boris Delibašić [a], Pietro Lio [d]

[a] University of Belgrade - Faculty of Organizational Sciences, Jove Ilica 154, Belgrade, Serbia
[b] University of Belgrade - Faculty of Mathematics, Studentski Trg 16, Belgrade, Serbia
[c] University of Belgrade - Faculty of Mechanical Engineering, Kraljice Marije 16, Belgrade, Serbia
[d] University of Cambridge, The Old Schools, Trinity Ln, Cambridge CB2 1TN, United Kingdom

ARTICLE INFO

Keywords:
Traffic congestion
Deep learning
Queuing theory
Inhomogeneous markov processes
Meta-heuristics

ABSTRACT

Traffic congestion is, nowadays, one of the most important highway problems. Highway tolls with booth operators are one of the causes of traffic congestion on highways, especially in rush hour periods, or during seasonal holiday travels. The value of driver waiting time (needed to stop and pay the toll) and the cost of the toll booth operators can reach up to about one-third of the revenue. In this paper we propose a novel methodology for continuous-time optimal control of highway tolls by predicting the optimal number of active modules (booths) in toll stations. The proposed methodology is based on a combination of recurrent neural networks, queuing theory, and metaheuristics. We utilized several recurrent neural network architectures for predicting the average intensity of vehicle arrivals. Moreover, the prediction error of the first recurrent neural network was modelled by another one in order to provide confidence estimates, additional regularization, and robustness. The predicted intensity of vehicle arrival rates was used as an input of the queuing model, whereas differential evolution was applied to minimize the total cost (waiting and service costs) by determining the optimal number of active modules on a highway toll in continuous time. The developed methodology was experimentally tested on real data from highway E70 in the Republic of Serbia. The obtained results showed significantly better performance compared to the currently used toll station opening pattern. The solutions obtained by solving a system of differential equations of the queuing model were also validated by a simulation procedure.

1. Introduction

One of the biggest challenges which occurs in modern-day traffic engineering is related to the reduction of traffic congestion in urban areas and highways (Ahmed, 2018). This problem is closely related to the loss of productivity due to the time spent in traffic jams (De and Rajbongshi, 2020), the pollution of the environment (Chowdhury et al., 2017), high fuel consumption (Bharadwaj et al., 2017), risks to human health (Requia et al., 2018), etc.

A wide variety of ideas have been proposed in order to reduce traffic congestion: optimization of traffic light management systems (Malim et al., 2019), usage of CCTV to monitor road conditions (Kurniawan et al., 2018), enforcing existing road traffic laws (Bump et al., 2019), improving perception of buses (AlRukaibi and AlKheder, 2019), and others. Nowadays, in the era of big data, a wide range of traffic optimization techniques based on machine learning (Ata et al., 2019; Devi and Neetha, 2017; Elfar et al., 2018) and different mathematical programming methods (Jamal et al., 2020; Kumar et al., 2020; Guo et al., 2020) have become very popular research directions in this field. Since a wide variety of problems related to congestion reduction are combinatorial optimization problems, numerous different meta-heuristic algorithms are naturally employed for solving them (Chaimatanan et al., 2018; Abu-Shawish et al., 2020).

In tackling the problem of alleviating highway congestion, we start from the observation that severe highway congestion is often caused by the presence of human-operated highway toll stations and that the congestion severity depends on the number of active toll modules (operator booths). The greater the number of the active modules, the lower the congestion. However, opening of toll modules incurs service costs. Therefore, there is a trade-off between traffic congestion (meaning longer waiting times for the drivers), and the costs of toll station operation. Therefore, determining the optimal number of open modules is an important practical problem of operating a highway.

In this paper, we propose a novel mathematical methodology for predicting the optimal number of active modules on highway tolls in continuous time over a considered future time interval. The methodology is based on a combination of deep learning methods, queuing

[*] Corresponding author.
E-mail address: andrija.petrovic@fon.bg.ac.rs (A. Petrović).

https://doi.org/10.1016/j.engappai.2022.105683
Received 3 October 2021; Received in revised form 23 October 2022; Accepted 25 November 2022
Available online xxxx

theory, and metaheuristics algorithms. Firstly, bearing in mind that the process of car arrival on toll stations is generally non-stationary, the average intensity of vehicle arrivals was predicted by several different recurrent neural network architectures such as: vanilla recurrent neural network, long–short term memory network, long–short term memory network with layer normalization, and gated recurrent unit network. Moreover, the error of predicting the average intensity of vehicle arrivals, was modelled by another network in order to provide confidence estimates, additional regularization, and robustness to the overall methodology. Secondly, the predicted average intensity of vehicles obtained was used as an input for the queuing model. The queuing model describes the behaviour of the toll-station system by relying on a non-stationary system of differential equations. Eventually, based on the toll-station model, the metaheuristic differential evolution optimization algorithm was applied to minimize the total cost (waiting and service costs) for determining the optimal number of active modules on a highway toll. Furthermore, the developed methodology can be easily adapted to a wide variety of real-world problems such as: predicting optimal velocities of ski lifts in ski resorts, predicting the number of open cashiers in supermarkets, etc.

The contributions of this paper are the following:

- We provide a solution to the, to our knowledge, hitherto unsolved problem of predicting the optimal number of active modules on toll stations, thus efficiently controlling the highway toll stations in continuous time.
- To the best of our knowledge, we are the first to combine deep learning and queuing theory in a way which allows flexible modelling of the intensity function without using complicated simulation procedures nor oversimplifications of the queuing theory differential equation models.
- We apply the proposed methodology on real-world data and demonstrate its effectiveness in practice.

Compared to the other metaheuristic (Abu-Shawish et al., 2020) based approaches that are applied in traffic control and optimization, in this paper we introduce a more sophisticated algorithm that is based on a combination of domain knowledge (modelling toll station by queuing theory), machine learning (prediction of future vehicle arrival intensities by deep learning), and metaheuristic optimization. The main advantage of proposed approach compared to the existing ones is more sophisticated modelling of the toll control process which is achieved by exploiting all of the said elements in modelling, which previous work has not done.

The paper is structured as follows. The background methodology, namely, recurrent neural network models, Markov processes, and differential evolution algorithms are presented in Section 3. The proposed mathematical methodology is presented in Section 4. In Section 2 the related work is reviewed. Experimental evaluation is presented in Section 5. Finally, the conclusions are drawn in Section 6.

2. Related work

In this section we provide a broad overview of results in different fields closely related to our work. Our work belongs to the broader field of traffic state estimation (TSE) which focuses on the estimation of specific traffic state indicators (TSIs), such as flow speed, average vehicle speed, traffic density, and other useful variables on road segments using partially observed traffic data (Seo et al., 2017). A successful prediction of a TSI in future provides useful information for determining price politics (Fosgerau and De Palma, 2013), for reduction of traffic congestion (Litman, 2016), and for strategic decision-making (constructing new roads, improving infrastructure etc.) (Koh and Lee, 2020).

TSE by classic machine learning. TSE has two major branches: the traditional domain modelling approach of predicting and inferring TSIs based on statistics and domain knowledge (Dadashova et al., 2021; Zhang et al., 2014; Yildirimoglu and Geroliminis, 2013) and the machine learning modelling approach based on historical and current traffic data. The machine learning modelling approach has become prevalent due to the availability of large amounts of data and better performance achieved compared to the traditional methods. The advantage of the machine learning approach comes from its ability to infer TSIs based on learning new representations of the data which successfully model hidden variables which cannot be observed by the traditional statistical models. Such hidden variables turned out to be very important for TSE (Liang et al., 2018; Baggag et al., 2019). The most frequently used machine learning methods for TSE are Kalman filters (Sun et al., 2003), kernel regression (Sun and Chen, 2008), k nearest neighbours, (KNN) (Luo et al., 2019), ensemble techniques (Liu and Wu, 2017), and deep learning methods (Fadlullah et al., 2017).

TSE by deep learning. Many papers deal with TSI prediction by deep learning models. Wei et al. (2019) propose a novel traffic flow prediction method, called auto-encoder long short-term memory. The auto-encoder is used to model the internal relationships of traffic flow variables by extracting the features of upstream and downstream traffic flow data. Moreover, the LSTM network utilizes thusly modelled information and the historical data to make complex non-linear traffic flow predictions. Similarly, Fu et al. (2016) used a combination of LSTM and GRU networks to predict short-term traffic flow and demonstrated that RNNs in general perform better than the auto regressive integrated moving average models (ARIMA) models. Zhou et al. (2020) presented a combination of statistic model GASVR and LSTM in order to obtain high precision of real-time prediction of traffic flow indicators. The comparison and analysis of various algorithms show that the prediction algorithm GASVR-LSTM proposed in this paper obtained a 20% higher R^2 measure than the LSTM, GRU, ARIMA, and convolutional neural networks (CNN). In order to predict traffic speed based on information provided by BeiDOU satellite navigation system Zhao et al. (2019) proposed the methodology based on LSTM that obtained better prediction performance compared to the support vector regression methods. Effective long-term forecasting of traffic flow based on the combination of hard attention mechanism with LSTM was presented in Wang et al. (2020). The goal of hard attention mechanisms is to learn similarity patterns to enhance neural memory and reduce the accumulation of error propagation. It is verified that the model can accurately learn local features and long term dependencies and has better accuracy and stability in long term sequence prediction compared to short term sequence prediction.

Queuing theory and its applications in traffic control. Queuing theory is an operations research field which presents the mathematical study of waiting lines, or queues and related waiting times. Queues are common in different aspects of everyday life, such as petrol or filling stations (Galankashi et al., 2016), supermarkets (Luo and Shi, 2020), clinics (Aziati and Hamdan, 2018), parks (Daniels et al., 2017), manufacturing firms (Ghalehkhondabi and Suer, 2018), etc. Queuing theory is also used as a baseline modelling approach in traffic problems. Swathi (2019) modelled vehicular traffic at signalized intersection using queuing theory. Based on this model Swathi showed that the developed model could be used to reduce road delays in the Maduravoyal neighbourhood of Chennai. Yang and Yang (2014) provided a general framework for queuing theory application in the traffic flow of intersections. Poomrittigul et al. (2019) proposed a simulation model for ticket system of Bangkok rapid transit train using queuing theory. The proposed model showed satisfactory performance and a possibility to be used in different locations. Similarly the analysis of traffic conditions on a particular urban highway using queuing theory approach is presented in Raskar and Nema (2017). The obtained results provided satisfactory analysis to predict traffic congestion during peak hours. Kachroo et al. (2016) showed that the queuing theory can provide a mathematical framework for dynamic congestion pricing and deriving optimal control law to determine an optimal tolling price.

Metaheuristic optimization in traffic control. Different metaheuristic optimization algorithms are applied in traffic management and traffic control. Jamal et al. (2020) presented an intelligent intersection control for delay optimization based on the genetic algorithms and differential evolution. The study results indicated that both GA and DE produced a systematic signal timing plan and significantly reduced travel time delay ranging from 15% to 35% compared to existing conditions. Furthermore, an evolutionary algorithm can also be used to determine the optimal policy for traffic lights (Mihăiţă et al., 2018). It is shown that it can yield a significant decrease of travel time during congestion periods. A genetic algorithm approach for signal setting optimization of signalized junctions in a congested road network is presented in Teklu et al. (2007). Abu-Shawish et al. (2020) presented a systematic review of metaheuristic techniques that are used in optimization of traffic light control systems. Additionally, numerous strategies and methods for traffic control and prediction is presented in Papageorgiou et al. (2003), while various modern approaches for traffic control and prediction are summarized in Castillo et al. (2015).

3. Background

In this section we introduce basic concepts from the literature that we apply in our work.

3.1. Recurrent neural networks

A recurrent neural network is a type of an artificial neural network which processes sequential data (e.g. time series). While feed forward deep neural networks assume that outputs are mutually independent for given inputs, recurrent neural networks naturally model dependencies between outputs within a sequence. Hence, the outputs in the sequence not only depend on the inputs, but also on each other. In this paper we used four types of RNN architectures:

- recurrent neural network (RNN) (Yu et al., 2019), which processes a sequence by successive parametrized transformations of the individual elements of the input sequence and the current hidden state, also yielding outputs in each processing step,
- long–short term memory (LSTM) network (Yu et al., 2019), which augments the computation of the hidden state by additional parametrized units (called gates) which make transformations of inputs, hidden states, and the outputs themselves dependent on a given input and current hidden states,
- long–short term memory with layer normalization (LSTM-ln), which includes a layer normalization technique over standard LSTM units (Ba et al., 2016), and
- gated recurrent units (GRU) (Dey and Salem, 2017), which simplifies LSTMs by dispensing with the output gate. A detailed review of different RNN architectures along with their applications is presented in Yu et al. (2019).

3.2. Queuing theory and inhomogeneous Markov processes

The most complex physical phenomena have stochastic character, but many of them can be satisfactory approximated in the form of functional relations. Most often dynamical systems' states cannot be well described deterministically. Therefore, temporal state dynamics models should incorporate some kind of stochastic laws. Many dynamical systems can be satisfactorily approximated by assuming the Markov property. The Markov property means that the evolution of the system in the future depends only on the present state and not on the prior state history. In this paper, the process of changing queue lengths at a toll station system is assumed to have the Markov property (Shortle et al., 2018).

Definition 1. The stochastic process $S(m)$ defined on the discrete state space has (continuous-time) Markov property if and only if for any integer value n, any sequence of integers $i_1, \dots, i_n, j$ and any strictly increasing sequence of real numbers $t_1, t_2, \dots, t_{n+1}$, the conditional probability can be represented as:

$$P[X(t_{n+1}) = j | X(t_1) = i_1, \dots, X(t_n) = i_n] = P[X(t_{n+1}) = j | X(t_n) = i_n] \quad (1)$$

The probability density function of time that the process will spend in a state can be expressed as (Shortle et al., 2018) :

$$f(t) = \lambda(t) \cdot \exp(\lambda(t) \cdot t) \quad (2)$$

where $\lambda(t)$ is a non-negative function of time. Therefore, the time until an event is modelled as a general point process (as opposed to simpler processes in which λ is constant).

The probability of transition of a system from state i at time t to state j at the time t' is denoted:

$$p_{ij}(t, t') = P[X(t') = j | X(t) = i] \quad (3)$$

There are two types of Markov processes: *homogeneous* and *inhomogeneous*. In the homogeneous Markov process, the state transition probability distribution is time invariant (stationary). Thus, time until the event is distributed according to the exponential distribution. In the case of inhomogeneous Markov process, the state transition probability distribution is non-stationary. The probability that the process is going to be in state j at time t is denoted:

$$p_j(t) = P[X(t) = j] \quad (4)$$

Starting from the last two equations and under Markovian assumption, it can be shown (Sundarapandian, 2009) that the change of probability of a process being in state i at time t, can be expressed by a system of differential equations in the following matrix form:

$$\frac{\mathbf{dp}(t)}{dt} = \mathbf{Q}^{\intercal}(t)\mathbf{p}(t) \quad (5)$$

where $\mathbf{p}(t) = [p_0(t), p_1(t), p_2(t), \dots, p_n(t)]^{\intercal}$ is a column vector of probabilities of the system being at state i in time t and $\mathbf{Q}(t)$ is a transition matrix which describes the instantaneous change of state probabilities, defined as:

$$\mathbf{Q}(t) = \begin{bmatrix} -\lambda(t) & \lambda(t) & 0 & 0 & \dots & 0 & 0 \\ \mu & -(\mu + \lambda(t)) & \lambda(t) & 0 & \dots & 0 & 0 \\ 0 & \mu & -(\mu + \lambda(t)) & \lambda(t) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mu & -\mu \end{bmatrix} \quad (6)$$

where μ is service intensity and $\lambda(t)$ is intensity of clients arrivals. In this formulation we assume that the service intensity is stationary.

3.3. Point process simulation - Ogata's modified thinning algorithm

The basic idea behind Ogata's modified thinning algorithm (Ogata, 1981), used for point process simulation, is to determine when the next event is going to occur so that the sample has the properties of a generating point process. To this end, a homogeneous Poisson process on some interval $[t, t + l(t)]$ for some chosen function $l(t)$[1] is simulated. This Poisson process has a chosen constant intensity $m(t)$ on the interval $[t, t + l(t)]$, such that it holds $m(t) \ge \sup_{s \in [t, t+l(t)]} \lambda(s)$. After one simulates one value s from this process, there are two possibilities. If it holds $s > l(t)$, then there is no event in the interval $[t, t + l(t)]$, so one starts again from the time $t + l(t)$. However, if it holds $t_i \le l(t)$, there might be an event in $[t, t + l(t)]$. In the latter case it has to be decided whether to keep this point or not. To get the correct intensity, the point is kept with probability $\lambda(t + s)/m(t)$, where λ is the process intensity function. Whether or not the point is kept, the algorithm continues from $t + s$. The pseudocode of Ogata's modified thinning algorithm is presented in Algorithm 1.

[1] The value of $l(t)$ may be infinite.

Algorithm 1 Ogata's modified thinning algorithm

```
Input: T - time of simulation, $\lambda(t)$ - intensity function
Output: Sample of times at which events occurred $\mathcal{T}$
  $t = 0$
  $\mathcal{T} = \emptyset$.
  while $t < T$ do
      Compute $m(t)$ and $l(t)$
      Sample independent random variables $s \sim Exp(m(t))$ and $U \sim Unif([0,1])$.
      if $s > l(t)$ then
          $t = t + l(t)$
      else
          if $t + s \leq T$ and $U \leq \lambda(t+s)/m(t)$ then
              $\mathcal{T} = \mathcal{T} \cup \{t+s\}$
          $t = t + s$
  return $\mathcal{T}$
```

3.4. Differential evolution

Differential evolution (DE) (Storn and Price, 1997) is a type of a metaheuristic algorithm used to optimize an objective function (L) by an evolutionary process. Differential evolution proved to be one of the most robust and stable population-based meta-heuristic algorithms (Pant et al., 2020). This algorithm does not require gradients, so the objective function does not need to be differentiable. The objective function is user-defined and depends on the optimization problem. The algorithm searches the space of feasible solutions by maintaining a population of individuals and creating new solutions by combining the individuals. The individual which achieved the best value of the objective function is kept until new individuals with improved value of objective function appear. The process repeats itself until a given termination criterion (e.g., required number of iterations) is satisfied. The basic DE algorithm can be described in several steps which are presented in Algorithm 2 (Pant et al., 2020). Typical choice of parameters for differential weight F (used for creating new individuals) and threshold C_r are 0.8 and 0.9, respectively.

Algorithm 2 Differential evolution - DE

```
Input: Function $L$, population size $N$, threshold $C_r \in [0,1]$, differential weight $F \in [0,2]$
Output: Minimum point of the function $L$
  Randomly generate the initial population $x_1, \dots, x_N$
  while stopping criterion is not satisfied do
      for $i = 1$ to $N$ do
          Sample distinct agents $x_a, x_b, x_c$ from population (different from $x_i$)
          Sample $s \sim Unif(0,1)$
          if $s \leq C_r$ then
              $x = x_a + F \cdot (x_b - x_c)$
          else
              $x = x_i$
      if $L(x) < L(x_i)$ then
          replace $x_i$ by $x$ in the population
  return $\text{argmin}_{x \in \{x_1,\dots,x_N\}} L(x)$
```

4. Methodology

The developed mathematical methodology for predicting the number of active modules on highway tolls is illustrated in Fig. 1. It uses raw toll station and weather data to build a model of toll station operation. This model is obtained by combining vehicle arrival and service intensity models in a way which is well-grounded in queuing theory. Toll station operation model is then used to formulate the optimization problem which yields the control function which determines the number of active modules through time. The proposed methodology can be understood as the data processing pipeline consisting of the following steps:

- **Preprocessing and analysis of raw toll data** which yields vehicle arrival intensity data, service intensity data, weather conditions data, and information on various costs of operation.
- **Modelling of vehicle arrivals and service intensities** which relies on the said data on vehicle arrivals intensities and service intensities to build the corresponding models.
- **Modelling highway toll station** which consists of specifying the queuing theoretic model of the toll station which is computed based on the predictions of vehicle arrival and service intensity models.
- **Formulation of the optimization problem** which uses the toll operation model, vehicle arrival and service intensity models, and various cost parameters in order to set up the minimization problem which defines the optimal number of active modules through time.
- **Solving the optimization problem** which consists of the direct application of the differential evolution algorithm on the formulated minimization problem.

In the following subsections we explain the steps of the methodology. In order to keep the methodology general, we postpone various specifics related to our application scenario for the experimental section. For instance, weather conditions are relevant for TSE in general, so in the methodology we assume the availability of the weather conditions feature vector. However, specific features we use are described in the experimental section since not all practitioners will have the same feature set at their disposal. Similarly, we assume the use of RNNs in the methodology. However, specific RNN design choices are not a part of our methodology section, but of the experimental section since different choices may be optimal for different application scenarios and datasets.

4.1. Data preprocessing and analysis

The integrated toll network system is capable of collecting real-time information about traffic. The toll station information can be divided into two groups: static and dynamic information. Static information is independent of the real-time observations and includes information like the toll price, operating cost of the toll station, toll station lane number and type, toll payment method, etc. Dynamic information is dependent on the real-time observations and includes information like the vehicle arrival times, the toll service time of each vehicle, etc. The collected raw information must be preprocessed and cleaned so that useful information can be easily extracted. The input to this step of the methodology is raw static and dynamic toll station information and the information on weather conditions. The output contains (i) the information of similar nature, but cleaned from missing data and outliers and (ii) vehicle arrival intensities and service intensities computed from vehicle arrival and service times.

The data preprocessing procedure consists of the following steps:

- removing instances with missing data,
- detecting outliers,
- processing raw data to obtain time spent in queue and time of service for each vehicle,
- computing intensities of vehicle arrivals and service intensities in one module, and

The intensities of vehicle arrivals and service intensities are computed from the dataset using a sliding window of an empirically chosen length. The length is chosen so that there are no high intensity values of the first difference during short time periods. It can be assumed that

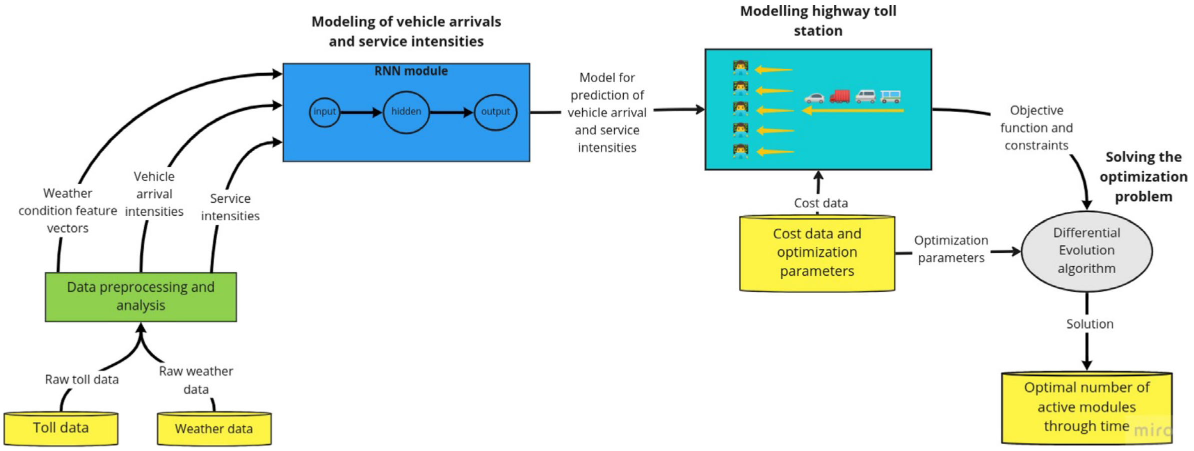
Modeling of vehicle arrivals and service intensities
RNN module
input
hidden
output
Model for prediction of vehicle arrival and service intensities
Modelling highway toll station
Objective function and constraints
Solving the optimization problem
Differential Evolution algorithm
Cost data
Cost data and optimization parameters
Optimization parameters
Solution
Optimal number of active modules through time
Weather condition feature vectors
Vehicle arrival intensities
Service intensities
Data preprocessing and analysis
Raw toll data
Raw weather data
Toll data
Weather data

Fig. 1. Stages of the proposed methodology for toll station control.

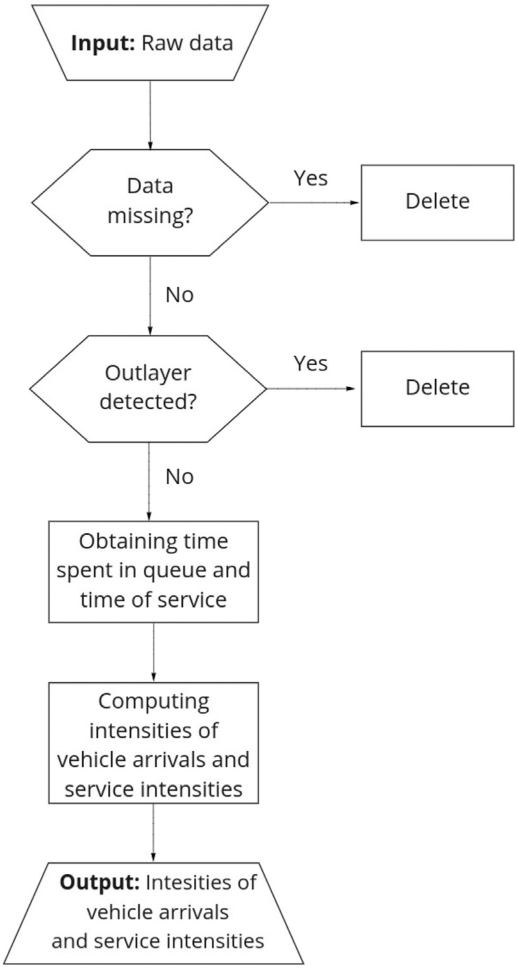
Input: Raw data
Data missing?
Yes
Delete
No
Outlayer detected?
Yes
Delete
No
Obtaining time spent in queue and time of service
Computing intensities of vehicle arrivals and service intensities
Output: Intesities of vehicle arrivals and service intensities

Fig. 2. Flow chart of data preprocessing and analysis procedure.

vehicle arrivals are uniformly distributed over the open toll modules. Thus, the vehicle arrivals intensities in one module, can be computed as total vehicles arrival intensity divided by the number of open modules. The computation of service intensities is analogous. The data preprocessing and analysis procedure is presented in Fig. 2.

4.2. Modelling of vehicle arrivals and service intensities

A predictive model of vehicle arrivals and service intensities can be made based on their values previously computed from the dataset. In order to do so, this step takes from the preprocessing step, as its inputs, the sequence of vehicle arrival intensities, the sequence of service intensities, and the sequence of weather conditions feature vectors. It outputs the predictive models of vehicle arrival and service intensities. We explain the modelling details.

We assume that waiting times (both in queue and for service) can be modelled as an inhomogeneous Markov process. Therefore, it is necessary to estimate the dynamics of point process intensities (parameters) for both arrivals and service time. In most real-world situations the event of vehicle arrival is modelled as a non-stationary point process, whereas service intensities are constant and can be represented by the exponential distribution. In order to prove that the service intensity is stationary, it is necessary to apply some kind of statistical test on the data sample (Chi-squared χ^2 test, Kolmogorov–Smirnov test). Such a test will be conducted on the empirical dataset. The service intensity is calculated (in the preprocessing step) as the inverse of the average service time, as:

$$\hat{\mu} = \frac{1}{\sum_{i=1}^{N} T_i^s} \tag{7}$$

The highway data is also used to estimate the intensity of the vehicle arrival times. If the distribution is time invariant, the intensities can be easily estimated from the sample by using classical statistics. Otherwise, the intensities are modelled by a RNN.

The inputs used for RNN based prediction of vehicle arrival intensity $\hat{\lambda}_t$ at time t are previously predicted intensities ($\hat{\lambda}_{\mathbf{hist}} = [\hat{\lambda}(t-1), \hat{\lambda}(t-2), \hat{\lambda}(t-3), \ldots]$) and weather condition feature vector $\mathbf{X}_t$. Therefore, intensities are predicted as:

$$\hat{\lambda}(t) = f_{\lambda}(\hat{\lambda}_{\mathbf{hist}}, \mathbf{X}_t, \mathbf{h}_{t-1}, t) \tag{8}$$

The expected outputs are the values already computed from the dataset. In the basic variant of the methodology, the RNN is optimized in a straightforward manner based on the mean squared error loss function. Since the information on prediction confidence can be very important in practice, we improve over the basic variant by using two different RNNs. The first RNN is trained in the same manner as in the basic variant, whereas the second RNN is used for estimating the error (standard deviation) of the first one. The error of the first RNN is predicted as:

$$\|\lambda(t) - \hat{\lambda}(t)\| = g_{\lambda}(\hat{\lambda}_{\mathbf{hist}}, \mathbf{X}_t, \mathbf{h}_{t-1}, t) \tag{9}$$

where $\mathbf{h}_{t-1}$ represents latent information from time stamp $t-1$.

4.3. Modelling of a highway toll station

The central part of the highway toll station model is the transition matrix $\mathbf{Q}(t)$. Therefore, the output of this step is the transition matrix, which is computed based on the service intensity μ and arrival intensity model $\hat{\lambda}(t)$ which are taken as an input from the previous step. The transition matrix $\mathbf{Q}(t)$ is defined as:

$$\mathbf{Q}(t) = \begin{bmatrix} -\frac{\hat{\lambda}(t)}{c(t)} & \frac{\hat{\lambda}(t)}{c(t)} & 0 & 0 & \dots & 0 & 0 \\ \mu & -(\mu + \frac{\hat{\lambda}(t)}{c(t)}) & \frac{\hat{\lambda}(t)}{c(t)} & 0 & \dots & 0 & 0 \\ 0 & \mu & -(\mu + \frac{\hat{\lambda}(t)}{c(t)}) & \frac{\hat{\lambda}(t)}{c(t)} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mu & -\mu \end{bmatrix} \tag{10}$$

where $\hat{\lambda}(t)$ is the predicted intensity of vehicle arrivals on toll stations, $c(t)$ is the number of open modules on a toll station and $\hat{\mu}$ is the estimated intensity of service.

It can be seen that the probability distribution of the system states depends on the number of open modules on a toll station: an increasing number of open modules on the toll station decreases the probability of long queues and vice versa. The analytical solution to the system of differential equations is intractable, because of nonlinear term $\lambda(t)$, hence the Runge–Kutta–Fehlberg (Griffiths and Smith, 2006) method is used. Due to the fact that the number of possible system states is infinite, to numerically solve the system of differential equations, a maximum number of vehicles in a queue must be assumed.

4.4. Formulation of the optimization problem

The number of open modules on highway toll stations at each time can be found by minimizing objective function - L (the total system cost or loss). In order to define it, this step requires as its input the static toll station information, specifically various costs (e.g., station operation cost) and models of service intensity and vehicle arrival intensity, and the transition matrix $\mathbf{Q}(t)$. The step outputs the precise minimization problem to be solved. The loss is defined as:

$$L = \int_0^T (WC(t) + SC(t))dt \tag{11}$$

where $WC(t)$ is waiting cost at time t and $SC(t)$ is service cost at time t. The service cost at time t can be defined as:

$$SC(t) = c(t) \cdot C_{tr} \tag{12}$$

where $c(t)$ is the number of open modules at time t and C_{tr} is the cost of one open module (toll ramp) on toll station per one minute of operation.

The waiting costs in moment t can be defined as the product of the cost for one vehicle waiting in the queue $C_w \cdot t$, the probability density of the time spent in the system by a vehicle $f(t)$, and the intensity of client arrivals $\lambda(t)$ in time t. Thus, the waiting cost can be represented as:

$$WC(t) = \lambda(t) \cdot f(t) \cdot C_w \cdot t \tag{13}$$

where C_w is the costs of waiting in queue for one vehicle during one unit of time. The probability density function of the time spent in the system $f(t)$ can be defined as (Bugaric, 2011):

$$f(t) = \frac{\rho(t) \cdot (\mu(t) - \lambda(t))}{1 - \rho(t)^{m+2}} \cdot \exp(-\mu(t) \cdot t) \cdot \sum_{k=0}^{m-1} \frac{(\lambda(t) \cdot t)^k}{k!} \tag{14}$$

where m is maximum number of vehicles that can be in queue and ρ is defined as $\rho(t) = \lambda(t)/\mu(t)$.

The goal is to minimize the loss L (Eq. (11)) subject to equality constraints (Eq. (5)) which define the change of the probability distribution of the process and inequality constraints which express the maximum number of open modules, with respect to the number of open modules $c(t)$ as a function of time. The mathematical formulation of optimization problem can be defined as:

$$\begin{aligned} \underset{c(t)}{\text{minimize}} \quad & \int_0^T (c(t) \cdot C_{tr} + \hat{\lambda}(t) \cdot f(t) \cdot C_w \cdot t)dt \\ \text{subject to} \quad & \frac{\mathbf{dp}(t)}{dt} = \mathbf{Q}^{\intercal}(t)\mathbf{p}(t) \\ & 1 \le c(t) \le c_{max} \text{ for all } t \in [0, T] \end{aligned}$$

If the confidence estimates are also given by a second network, the mathematical formulation of optimization problem can be defined as:

$$\begin{aligned} \underset{c(t)}{\text{minimize}} \quad & \int_0^T \int_{-\infty}^{\infty} \left(c(t) \cdot C_{tr} + (\hat{\lambda}(t) + \varepsilon) \cdot f(t) \cdot C_w \cdot t \cdot p(\varepsilon|t) \right) d\varepsilon dt \\ \text{subject to} \quad & \frac{\mathbf{dp}(t)}{dt} = \mathbf{Q}^{\intercal}(t)\mathbf{p}(t) \\ & 1 \le c(t) \le c_{max} \text{ for all } t \in [0, T] \end{aligned}$$

where $p(\varepsilon)$ is a normal distribution with mean 0 and standard deviation obtained as the prediction of the second network.

4.5. Solving the optimization problem

This step takes as its input the specific minimization problem and outputs the control function determining the number of open modules through time. This function can be found by solving the proposed minimization problem using the DE algorithm. Luckily, the proposed minimization problem admits direct application of the algorithm and no specialization to the problem is required. Only the hyperparameter tuning of the algorithm is required, which we perform by grid search on a small subset of the data. Specifics of the final algorithm configuration are left for the experimental section.

5. Experimental evaluation

In this section we describe the data, important details of the experimental setup and provide results of the experiments.

5.1. Data

We used data from the toll station "Vrčin", the southern entrance to the city of Belgrade, Republic of Serbia, on the E70 highway, in the period from August to September 2017. A 180 min sliding window was used for computing vehicle arrivals intensities. The goal is to make predictions of vehicle arrival intensities every five minutes for 6 h ahead. In order to make such predictions, three different groups of relevant features, were extracted from data: raw, descriptive, and meteorological features. The raw features are related to the number of vehicles that passed through the toll station at given time and are obviously relevant for modelling intensities. The descriptive features are common descriptive statistics computed from the speed of vehicles. They serve the purpose of better reflecting the structure of the related distribution. The meteorological features are related to weather conditions which can have big influence on traffic intensity. All used features are provided in Table 1. Regarding meteorological features, besides data from Vrčin, data from Niš toll station (the starting point of the highway section) were used as well.

Apart from these features, in each time step, as inputs we also used the intensities that were predicted by the model for the last 6 hours prior to that time step. Hence, the total number of input features is 75. In the case of confidence estimation network, as inputs we also used the confidence estimate from the previous time step.

The total, the number of data points for testing and training the network is 35,013. 18,040 points were used for training the first RNN. 9714 points were used for validating the first RNN and training the second RNN for confidence estimation. The remaining 6939 data points were used for testing. For illustration, the intensities of arrivals are shown in Fig. 3.

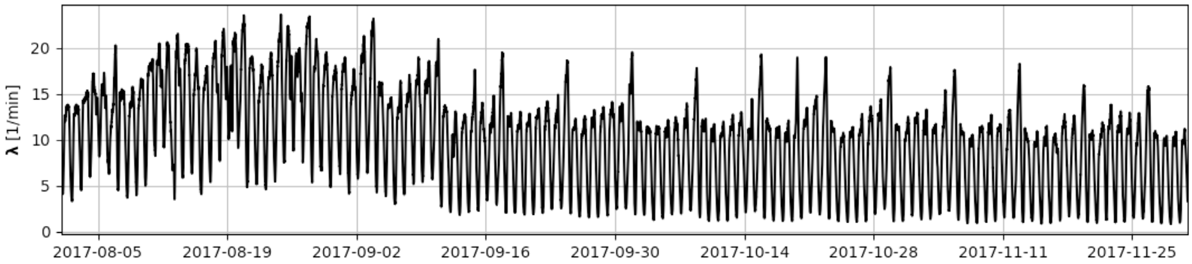

Fig. 3. Intensity of vehicles arrivals through time.

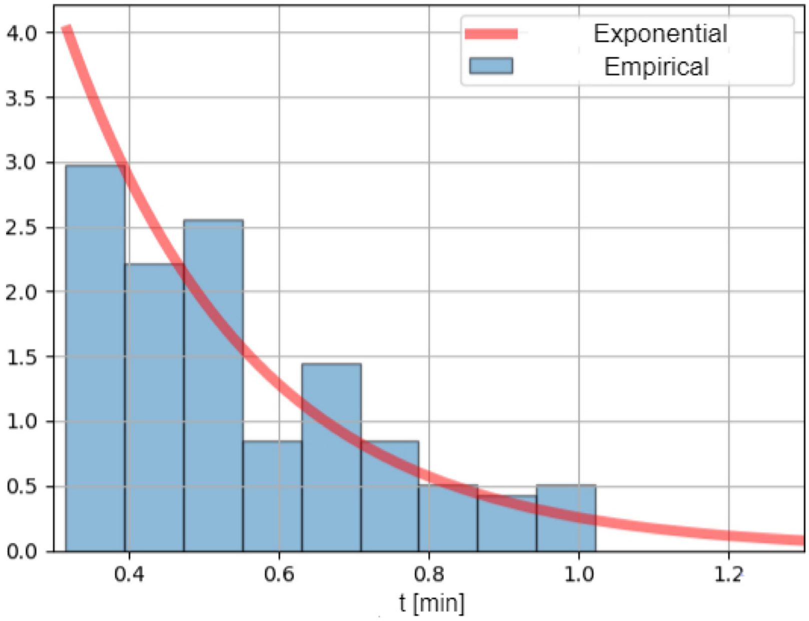

Fig. 4. The empirical data distribution and the exponential distribution.

Table 1
Features used for intensity prediction.

Feature group	Feature description
Raw	Number of vehicles Time
Descriptive	Average speed Maximal speed 10th speed percentile 25th speed percentile Speed median 75th speed percentile 95th speed percentile
Meteorological	Temperature (°C) Air humidity (%) Dew point (°C) Wind speed (m/s) Weather description Visibility (%)

5.2. Modelling client service time

We assume that the client service time can be modelled by a stationary exponential distribution. In Fig. 4, the empirical data histogram is shown along with the exponential distribution.

To test whether the vehicle service time can be modelled by an exponential distribution, the Kolmogorov–Smirnov (KS) test and Chi-squared (χ^2) test are used. These tests are suitable when the size of data sample is small and are computationally very expensive in case of big data. In order to overcome this problem, we randomly sample a manageable subset of the data 100 times, perform the tests on these samples, and average the obtained p values. KS and χ^2 statistics are shown in Table 2. All statistical tests have p values above 0.05. Thus we conclude that the data originates from an exponential distribution.

Table 2
Results of statistical tests.

Test	p value
KS	0,324
χ^2	0,059

5.3. Intensity prediction by RNN models

We used four recurrent neural network architectures for predicting the average intensity of vehicle arrivals: RNN, GRU, LSTM, and LSTM-ln (as described in Section 3.1). The number of layers and the number of nodes per layer in each of them were varied. The number of cells per layer was 8, 16, 32, and 64, whereas the number of layers ranged from 1 to 3. Additionally, dropout and early stopping were applied (20 epochs were used as early stopping patience criterion, whereas the dropout factor was set to be 0.8). The number of epochs in training was 2000. Root mean square error (RMSE) and coefficient of determination (R^2) were used as performance metrics. The best combination of hyperparameters for each architecture of RNNs has been chosen on validation set and then RNN architectures were compared on test set. A total of 96 different models have been trained, 24 for each of four architectures. The results for all RNN architectures with tuned hyperparameters are

Table 3
Results of intensity prediction by RNNs.

Name	Layers	Cells per layer	Dropout	RMSE	R^2
RNN	1	64	True	6.18	0.646
GRU	2	32	False	9,44	0.414
LSTM	3	8	True	5.317	0.681
LSTM-ln	3	32	True	2.012	0.881

presented in Table 3, whereas results on validation set for each of them are provided in the Appendix A.

It can be concluded that the layer normalization of the LSTM network significantly contributes to the prediction quality. Furthermore, from the results of validation (in the Appendix A), it can be noticed that the networks which apply dropout regularization do not have significantly better results compared to the other architectures. Moreover, GRU networks showed the worst performance. LSTM-ln network with 3 layers, 32 LSTM cells in a layer and dropout regularization showed the best results on the test set. Therefore, this network has been used for predicting the intensity of vehicle arrivals in Ogata's algorithm, and in DE algorithm it was used to predict expectations of the intensities of vehicle arrivals at the toll station.

Having in mind that LSTM-ln model achieved the best results in prediction of vehicle arrival intensities, the LSTM-ln model was used to predict confidence bounds of RNN predictions as well. This model was trained on the validation set and the hyper-parameter tuning was applied by time series rolling cross-validation with 5 splits. In Appendix A results obtained by this cross-validation are presented. The results show that the best performance are achieved by a network with 3 layers, 64 cells per layer, and with dropout regularization. RMSE of this network is 0.724, and R^2 is 0.792.

5.4. Toll station costs and DE parameters

First we specify the parameters of the optimization problem. Based on data gathered from public enterprise "Roads of Serbia", the cost of one open module at the toll station C_{tr} (Eq. (12)) is equal to the sum of the following costs:

- Electric energy costs C_{el} of toll station operation of 0.0027 EUR/min, with additional costs of heating and cooling C_{h-cl} of 0.0013 EUR/min.
- The costs of the work force C_{wf} of 0.0972 EUR/min. The workforce consists of the cost of the toll station operators and supervisors for 9 toll ramps, each requiring 800 EUR.
- Other costs C_{m-ot} of maintenance and material costs of 0.0607 EUR/min.

Therefore, the total cost of a toll ramp is $C_{tr} = C_{el} + C_{h-cl} + C_{wf} + C_{m-ot}$, i.e. 0.162 EUR/min.

The costs of waiting in queue per vehicle C_w can be expressed as sum of the following costs:

- The cost of fuel C_{fuel} for an average fuel consumption of 1.3 l/h is 0.0325 EUR/min.
- The cost of lost time C_{lt} according to the gross average salary of 700 EUR per month per person in the vehicle. Based on assumption that on average 1.5 persons are in the vehicle, this cost is 0.109 EUR/min.
- Other costs of waiting C_{w-ot} include vehicle maintenance and costs of potential losses induced by fatigue. These costs are assumed to be 10% of the other listed costs.

Therefore, the cost of waiting is equal to $C_w = C_{fuel} + C_{lt} + C_{w-ot}$, i.e. 0.155 EUR/min per vehicle.

After experimenting on a small data sample, we obtained the following DE parameters to be used in the optimization procedure:

- "DE/rand/1/bin" strategy is used (Storn and Price, 1997).
- Differential weight is $F = 0.8$.
- Threshold is $C_r = 0.9$.
- The population size is 500 individuals.
- The maximum number of generations over which the population is evolved is 1000.

5.5. Toll station control results

We evaluate the toll station control in the following way. Using described methodology, based on previously defined costs and DE parameters, we obtain the control function which determines the optimal number of open active modules on toll station in each step over the future time interval of 300 min. We assume that the function is piecewise constant and that it can change its value (the number of active modules) at regular time intervals — after a fixed time has elapsed since the previous change. We call a function which changes value after time t a t *decision making strategy*. We evaluate several such strategies for diminishing values of t. Each such strategy is compared in terms of total cost (or loss) L to the strategy currently used by the company operating the highway. Moreover, real and predicted toll station state probabilities (probabilities of a number of vehicles on toll station) for optimized opening strategy are compared.

The Fig. 5 presents the results of experiments with 150 min decision making strategy. It shows the real and predicted toll station state probabilities (top), currently applied and optimized decision making strategies (middle) and instantaneous cost which is a function integrated to obtain total cost L (bottom). Firstly, it can be observed from the top plot that although LSTM-ln achieved satisfactory prediction performance, the predicted system state probabilities (thick lines) have higher values compared to the real state probabilities. Since the figure shows probabilities of smaller number of vehicles in the queue, there is obviously slight bias towards shorter queues. Secondly, in the middle plot, the optimized strategy for opening modules in future significantly differs from currently used strategy. Most notably, the number of open modules is much smaller, hinting at the potential of savings in operation costs. Finally, the total cost (area below the line) of optimized strategy, shown in the bottom plot, is evidently smaller than that of the currently applied strategy. Furthermore, instantaneous cost is almost consistently lower than the cost of the currently applied strategy (with expectation of two peaks before 200 min and after 250 min).

Similar conclusions can be made based on Fig. 6 which presents the results for the 60 min decision making strategy. The state probabilities of the real and the predicted system states for 60 min strategy differ less than for the 150 min strategy. Similar conclusions can be made based on Figs. 7 and 8 which present the results for 30 min and 20 min opening strategy, respectively.

The results for the extreme case of 1 min decision making strategy are presented in Fig. 9. The difference between real and predicted system state probabilities is the lowest in this case. In practice, reasonable time intervals for opening modules are at least 50 min.

We exploit LSTM with confidence estimates to show expectation and samples of predicted toll station probabilities in Fig. 10 (top) for 50 min opening strategy. Moreover, currently applied and optimized decision making strategies (middle) and cost (bottom) are presented. The obtained results do not differ much from the values obtained by the model without the confidence intervals.

In all figures, the peaks in the instantaneous costs can be observed when the queue is short. These peaks are the result of the optimization procedure. Further studies should consider ways to eliminate this peaks by usage of dynamic programming or Pontryagin maximum principle.

In order to provide overall results, the optimization procedure is repeated 50 times and the mean total cost for various lengths of decision intervals is presented in Table 4. The best results were obtained in the case when the optimal strategy is chosen at 1 min intervals, but the realistic scenarios are the ones with decision intervals of at least 50

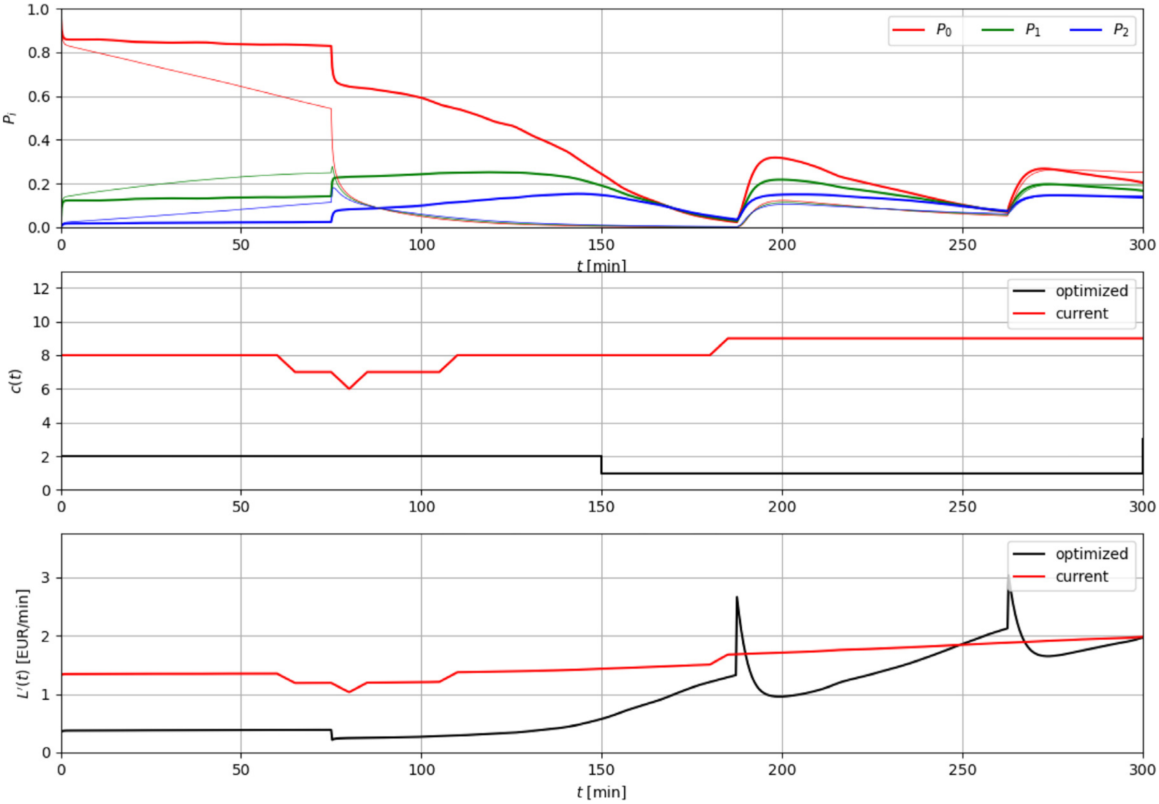

Fig. 5. Results of 150 min decision making strategy. Top plot shows the dynamics of probabilities of first three system states given predicted arrival intensities (thick line) and given real vehicle arrival intensities (slim line). Middle plot shows the number of open channels according to the optimized (black) and currently used (red) decision making strategy. Bottom plot shows instantaneous cost over time for the optimized (black) and currently used (red) decision making strategy.

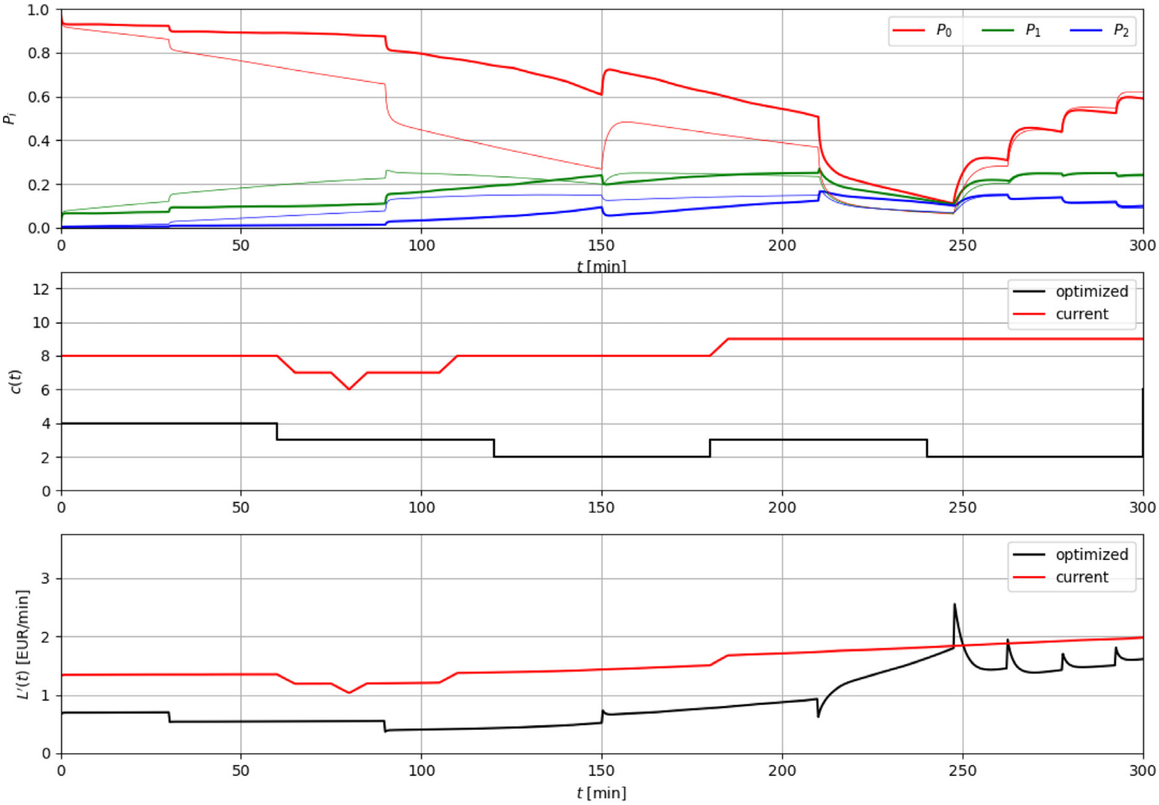

Fig. 6. Results of 60 min decision making strategy. Top plot shows the dynamics of probabilities of first three system states given predicted arrival intensities (thick line) and given real vehicle arrival intensities (slim line). Middle plot shows the number of open channels according to the optimized (black) and currently used (red) decision making strategy. Bottom plot shows instantaneous cost over time for the optimized (black) and currently used (red) decision making strategy.

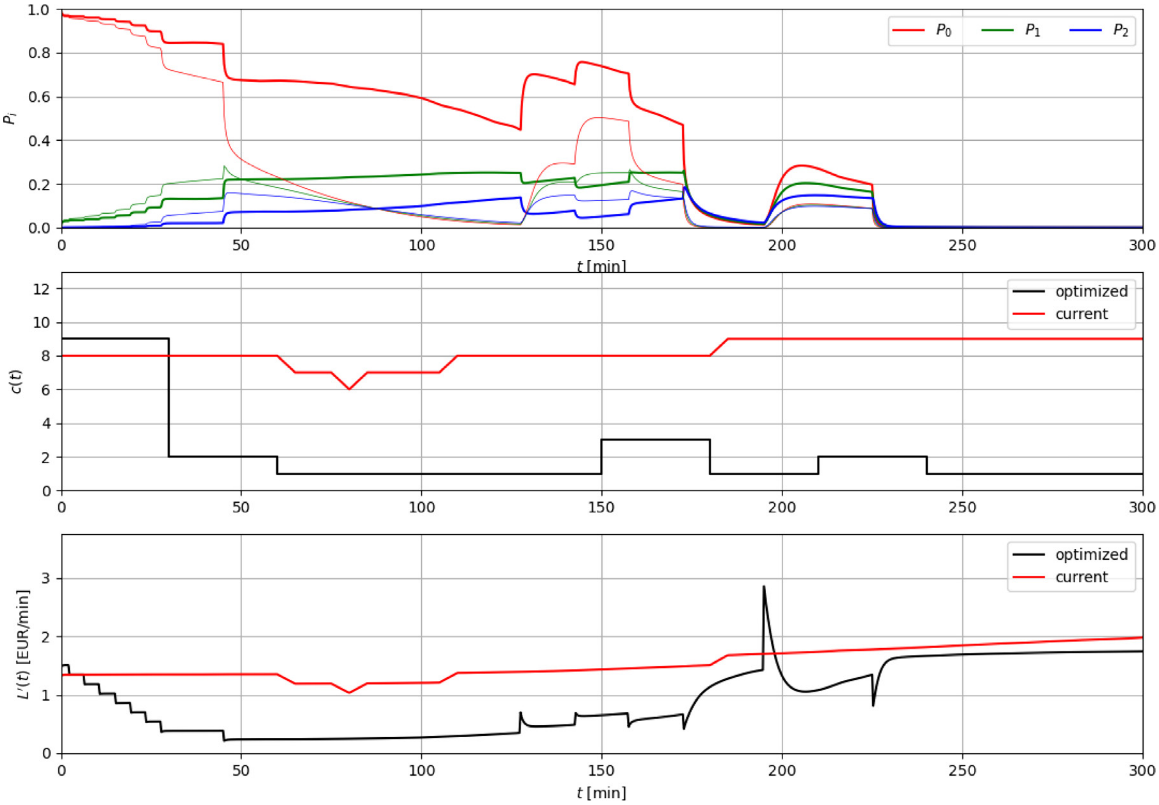

Fig. 7. Results of 30 min decision making strategy. Top plot shows the dynamics of probabilities of first three system states given predicted arrival intensities (thick line) and given real vehicle arrival intensities (slim line). Middle plot shows the number of open channels according to the optimized (black) and currently used (red) decision making strategy. Bottom plot shows instantaneous cost over time for the optimized (black) and currently used (red) decision making strategy.

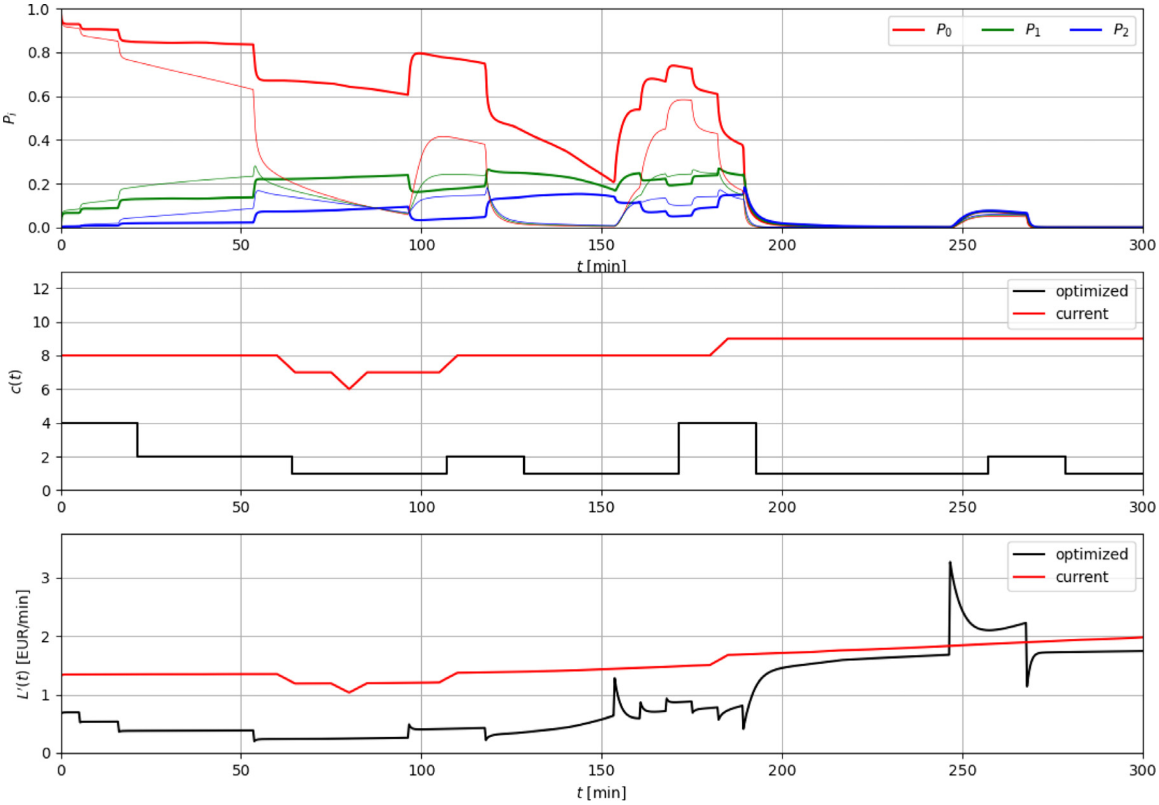

Fig. 8. Results of 20 min decision making strategy. Top plot shows the dynamics of probabilities of first three system states given predicted arrival intensities (thick line) and given real vehicle arrival intensities (slim line). Middle plot shows the number of open channels according to the optimized (black) and currently used (red) decision making strategy. Bottom plot shows instantaneous cost over time for the optimized (black) and currently used (red) decision making strategy.

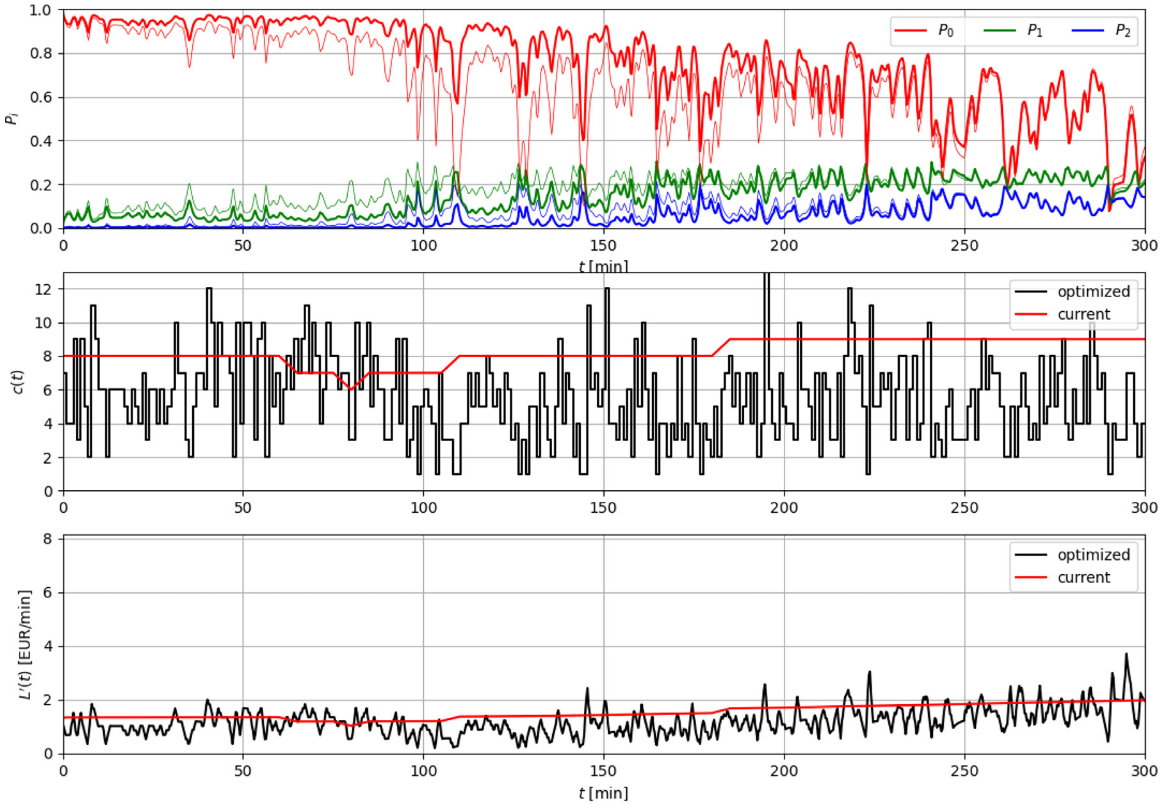

Fig. 9. Results of 1 min decision making strategy. Top plot shows the dynamics of probabilities of first three system states given predicted arrival intensities (thick line) and given real vehicle arrival intensities (slim line). Middle plot shows the number of open channels according to the optimized (black) and currently used (red) decision making strategy. Bottom plot shows instantaneous cost over time for the optimized (black) and currently used (red) decision making strategy.

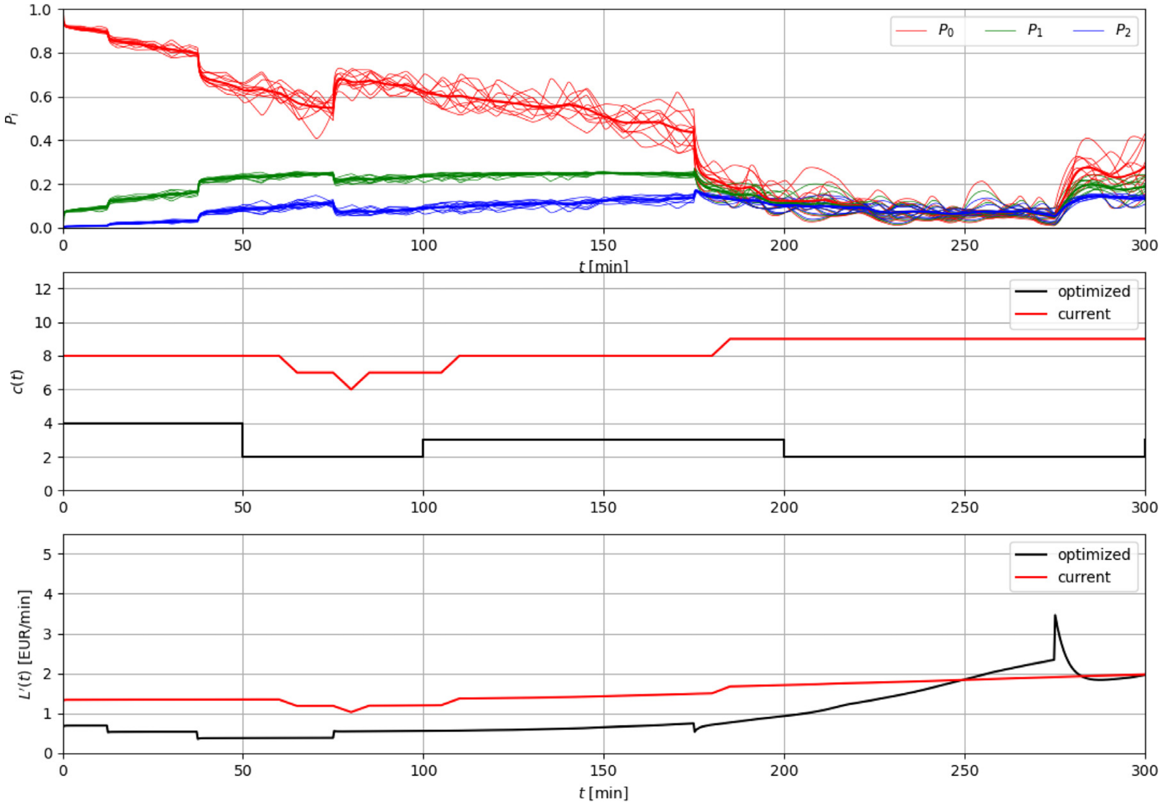

Fig. 10. Results of 50 min decision making strategy based on LSTM with confidence estimates. Top plot shows the expectation (thick lines) and samples (slim lines) of probabilities of first three system states given predicted arrival intensities. Middle plot shows the number of open channels according to the optimized (black) and currently used (red) decision making strategy. Bottom plot shows instantaneous cost over time for the optimized (black) and currently used (red) decision making strategy.

Table 4
Total costs for various length of decision intervals expressed in EUR/min.

Time of decision intervals	Without confidence estimates	With confidence estimates
150	267.36	260.32
60	258.52	255.55
50	259.95	250.12
30	268.79	258.34
20	255.61	248.90
1	245.52	246.66
Current control strategy	457.56	

Table 5
Simulation errors.

# Decision strategy [min]	# of simulation samples				
	Absolute error				
	200	1000	3000	5000	10 000
1	0,044	0,021	0,017	0,014	0,013
60	0,055	0,025	0,022	0,018	0,016
	Relative error				
1	0,237	0,142	0,119	0,108	0,095
60	0,169	0,121	0,106	0,091	0,085

min. Nevertheless, values of the optimized strategy do not differ much for different decision intervals. Moreover, in all cases the total cost of optimized strategy is much less than the cost of currently applied strategy. It can be observed that the methodology with confidence estimates provides slightly better results than the methodology without them.

5.6. Validation

Results obtained by optimization are validated by a simulation procedure based on the predicted intensities. The Ogata's modified thinning algorithms was used and the errors of simulation with respect to the model obtained by optimization are computed. The errors are expressed in terms of relative error, absolute error, and estimated error score. The absolute error is computed as the mean of the absolute difference between the number of simulated vehicles in the system ($K_i(t)$) and the numerical solution obtained from a system of differential equations ($K(t)$).

$$\frac{1}{N}\sum_{i=1}^{N}|K_i(t) - K(t)| \tag{15}$$

N is the number of simulations and $K(t)$ is the mean number of vehicles within a single channel, determined as:

$$K(t) = \sum_{i=1}^{N} i \cdot p_i \tag{16}$$

Note that probabilities p_i are obtained by solving the system of differential equations given by Eq. (5). The relative error is computed as:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{|K_i(t) - K(t)|}{K(t)} \tag{17}$$

The errors obtained by simulating the toll station system for different number of simulation epochs are presented in Table 5.

The dynamics of first three state probabilities is presented in Fig. 11. It can be observed that the simulated probabilities agree with the results obtained by the optimized model. Thus, it can be concluded that the model appropriately describes the expectation of the true underlying stochastic nature of highway toll station process. The apparent greater width of coloured lines representing simulated probabilities is the consequence of greater variance due to the lower sample size. For greater sample sizes (lower plots), the coloured lines appear slimmer.

6. Conclusions

We introduced a novel mathematical methodology for deciding the optimal number of active modules on highway tolls. The methodology is based on a combination of RNNs with queuing theory and metaheuristic algorithms. We experimented with four different RNN architectures for prediction of average intensity of vehicle arrivals. Moreover, the confidence estimates of the predicted average intensity of vehicle arrivals were modelled by another RNN in order to provide robustness to the methodology and additional information for the user. The predicted intensities were used as inputs to the queuing model to obtain a non-stationary system of differential equations. Based on this model, differential evolution metaheuristic algorithm was applied to minimize the total cost by determining the optimal number of open modules on highway tolls in the future. To the best of our knowledge, this is the first algorithm that combines deep learning and queuing theory in a way which allows flexible modelling of intensity functions without using complicated simulation procedures nor oversimplifications of the queuing theory differential equation models. The proposed methodology was applied on real-world data and the obtained control strategy for opening modules achieved significantly lower total cost compared to the currently used control strategy. Besides the evaluation on real data, the usefulness of the methodology was validated in simulations using Ogata's modified thinning algorithm.

Further studies should address extending the proposed methodology by directly optimizing the likelihood of the underlying point process generating vehicles arrivals by applying neural ordinary differential equations. Moreover, present study cannot accommodate directly for unexpected events (i.e accidents, road works) which cause substantial delays. RNN which is predicting average vehicle intensities might learn to predict such delays (increase in the average vehicle intensities) through input features. However, further studies should deal specifically with this problem and try to model unexpected events explicitly and try to combine such model with the currently used model for predicting average vehicle intensities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

Acknowledgements

This work was supported in part by the ONR/ONR Global under Grant N62909-19-1-2008. In addition, this research is partially supported by the Ministry of Science, Education and Technological Development of the Republic of Serbia grants OI174021 and TR41008. The authors would also like to express gratitude to the company Saga New Frontier Group Belgrade, for supporting this research.

Appendix A. Model architecture

Prediction performance results of RNN, GRU, LSTM and LSTM-ln models for different values of hyperparameters on validation set are presented in Tables A.6, A.7, A.8, A.9, respectively. The results of LSTM-ln confidence estimate evaluation are presented in Table A.10.

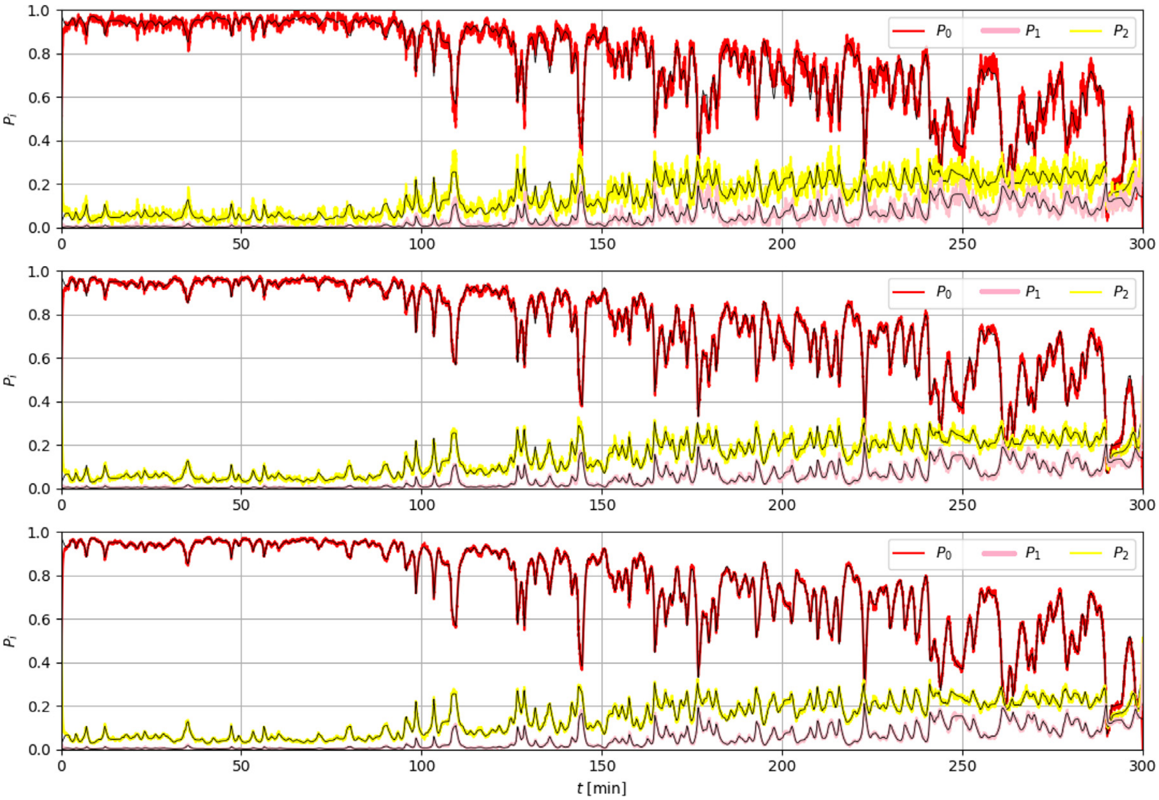

Fig. 11. Dynamics of the probabilities of first three states obtained from the system of differential equations (black lines) and estimated from simulations (coloured) based on 200 (top), 3000 (middle) and 10 000 (bottom) samples.

Table A.6
Prediction performance of the RNN on validation set.

R^2 score with dropout regularization					R^2 without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	0.484	0.453	0.555	0.648	0.328	0.382	0.523	0.551
2	0.446	0.501	0.385	0.383	0.596	−0.389	0.427	0.544
3	0.382	0.275	0.28	−0.193	0.626	−0.151	0.239	0.133
RMSE with dropout regularization					RMSE without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	9.017	9.563	7.779	6.19	11.743	10.794	8.333	7.843
2	9.691	8.729	10.743	10.79	7.051	24.306	10.009	7.962
3	10.8	12.679	17.003	38.363	6.541	20.119	13.307	15.161

Table A.7
Prediction performance of the GRU on validation set.

R^2 score with dropout regularization					R^2 without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	0.594	0.644	0.413	0.212	−0.534	−2.423	−5.596	−0.584
2	0.084	−8.209	−14.333	−1.254	−7.663	0.157	−0.288	0.418
3	0.213	0.142	−3.623	0.231	−1.248	0.324	0.147	0.212
RMSE with dropout regularization					RMSE without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	32
1	6.752	5.923	9.475	12.342	25.577	57.077	110.002	21.999
2	15.263	153.531	255.652	65.232	29.395	14.049	21.481	9.489
3	12.587	10.325	72.244	12.674	17.325	9.876	10.021	12.122

Table A.8
Prediction performance of the LSTM on validation set.

R^2 score with dropout regularization					R^2 without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	−0.095	−0.1	−0.377	−4.205	0.14	−0.319	−4.361	−1.031
2	0.267	0.218	−1.97	−2.253	−0.936	0.071	0.516	−2.278
3	0.683	0.633	0.391	0.483	0.632	0.578	0.619	−0.004
RMSE with dropout regularization					RMSE without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	18.264	18.353	79.58	86.784	14.331	21.991	89.377	188.64
2	12.222	13.024	49.628	17.048	32.284	15.486	8.058	54.657
3	5.320	6.108	10.142	8.613	6.124	7.032	6.347	16.747

Table A.9
Prediction performance of the LSTM-ln on validation set.

R^2 score with dropout regularization					R^2 without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	0.726	0.703	0.624	0.692	0.679	0.517	0.431	0.77
2	0.812	0.834	0.803	0.817	0.855	0.803	0.868	0.856
3	0.781	0.772	0.879	0.653	0.824	0.849	0.843	0.849
RMSE with dropout regularization					RMSE without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	4.558	4.942	6.253	5.125	5.348	8.04	9.481	3.822
2	3.119	2.761	3.27	3.044	2.408	3.27	2.194	2.391
3	3.698	3.797	2.015	5.784	2.931	2.515	2.609	2.503

Table A.10
Prediction performance of the LSTM-ln confidence estimate on validation set.

R^2 score with dropout reg.					R^2 without dropout reg.			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	0.714	0.729	0.642	0.659	0.594	0.550	0.608	0.634
2	0.635	0.773	0.750	0.762	0.642	0.432	0.596	0.594
3	0.634	0.791	0.674	0.792	0.684	0.651	0.619	0.675
RMSE with dropout regularization					RMSE without dropout regularization			
# Layers	# cells by layer				# cells by layer			
	8	16	32	64	8	16	32	64
1	0.870	0.854	0.739	0.821	0.892	0.944	0.877	0.849
2	0.816	0.806	0.742	0.777	0.840	1.020	0.893	0.892
3	0.847	0.785	0.802	0.724	0.792	0.829	0.865	0.802

References

Abu-Shawish, I., Ghunaim, S., Azzeh, M., Nassif, A.B., Metaheuristic techniques in optimizing traffic control lights: A systematic review.

Ahmed, A.H., Adaptive intelligent traffic control systems for congestion management.

AlRukaibi, F., AlKheder, S., 2019. Optimization of bus stop stations in Kuwait. Sustainable Cities Soc. 44, 726–738.

Ata, A., Khan, M.A., Abbas, S., Ahmad, G., Fatima, A., 2019. Modelling smart road traffic congestion control system using machine learning techniques. Neural Netw. World 29 (2), 99–110.

Aziati, A.N., Hamdan, N.S.B., 2018. Application of queuing theory model and simulation to patient flow at the outpatient department. In: Proceedings of the International Conference on Industrial Engineering and Operations Management Bandung, Indonesia.

Ba, J.L., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

Baggag, A., Abbar, S., Sharma, A., Zanouda, T., Al-Homaid, A., Mohan, A., Srivasatava, J., 2019. Learning spatiotemporal latent factors of traffic via regularized tensor factorization: Imputing missing values and forecasting. IEEE Trans. Knowl. Data Eng..

Bharadwaj, S., Ballare, S., Chandel, M.K., et al., 2017. Impact of congestion on greenhouse gas emissions for road transport in Mumbai metropolitan region. Transp. Res. Procedia 25, 3538–3551.

Bugaric, P., 2011. Modeling of Queuing Theory Systems (in Serbian). Masinski fakultet Beograd.

Bump, J.B., Reddiar, S.K., Soucat, A., 2019. When do governments support common goods for health? Four cases on surveillance, traffic congestion, road safety, and air pollution. Health Syst. Reform 5 (4), 293–306.

Castillo, E., Grande, Z., Calviño, A., Szeto, W.Y., Lo, H.K., 2015. A state-of-the-art review of the sensor location, flow observability, estimation, and prediction problems in traffic networks. J. Sensors 2015.

Chaimatanan, S., Delahaye, D., Mongeau, M., 2018. Hybrid metaheuristic for air traffic management with uncertainty. In: Recent Developments in Metaheuristics. Springer, pp. 219–251.

Chowdhury, S., Dey, S., Tripathi, S.N., Beig, G., Mishra, A.K., Sharma, S., 2017. "Traffic intervention" policy fails to mitigate air pollution in megacity Delhi. Environ. Sci. Policy 74, 8–13.

Dadashova, B., Li, X., Turner, S., Koeneman, P., 2021. Multivariate time series analysis of traffic congestion measures in urban areas as they relate to socioeconomic indicators. Socio-Econ. Plan. Sci. 75, 100877.

Daniels, E.C., Burley, J.B., Machemer, T., Nieratko, P., 2017. Theme park queue line perception. Int. J. Cult. Herit. 2 (105–108), 20.

De, U.K., Rajbongshi, G., 2020. Statistical application for the analysis of traffic congestion and its impact in a Hill City.

Devi, S., Neetha, T., 2017. Machine learning based traffic congestion prediction in a IoT based Smart City. Int. Res. J. Eng. Technol. 4 (5), 3442–3445.

Dey, R., Salem, F.M., 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems. MWSCAS, IEEE, pp. 1597–1600.

Elfar, A., Talebpour, A., Mahmassani, H.S., 2018. Machine learning approach to short-term traffic congestion prediction in a connected environment. Transp. Res. Rec. 2672 (45), 185–195.

Fadlullah, Z.M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., Mizutani, K., 2017. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun. Surv. Tutor. 19 (4), 2432–2455.

Fosgerau, M., De Palma, A., 2013. The dynamics of urban traffic congestion and the price of parking. J. Public Econ. 105, 106–115.

Fu, R., Zhang, Z., Li, L., 2016. Using LSTM and GRU neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation. YAC, IEEE, pp. 324–328.

Galankashi, M.R., Fallahiarezoudar, E., Moazzami, A., Yusof, N.M., Helmi, S.A., 2016. Performance evaluation of a petrol station queuing system: a simulation-based design of experiments study. Adv. Eng. Softw. 92, 15–26.

Ghalehkhondabi, I., Suer, G., 2018. Production line performance analysis within a MTS/MTO manufacturing framework: a queueing theory approach. Production 28.

Griffiths, D.V., Smith, I.M., 2006. Numerical Methods for Engineers. CRC Press.

Guo, C., Li, D., Zhang, G., Ding, X., Curtmola, R., Borcea, C., 2020. Dynamic interior point method for vehicular traffic optimization. IEEE Trans. Veh. Technol. 69 (5), 4855–4868.

Jamal, A., Tauhidur Rahman, M., Al-Ahmadi, H.M., Ullah, I., Zahid, M., 2020. Intelligent intersection control for delay optimization: Using meta-heuristic search algorithms. Sustainability 12 (5), 1896.

Kachroo, P., Gupta, S., Agarwal, S., Ozbay, K., 2016. Optimal control for congestion pricing: Theory, simulation, and evaluation. IEEE Trans. Intell. Transp. Syst. 18 (5), 1234–1240.

Koh, Y., Lee, G.M., 2020. Infrastructure investment and travel time. Econom. Lett. 187, 108901.

Kumar, A., Rajalakshmi, K., Jain, S., Nayyar, A., Abouhawwash, M., 2020. A novel heuristic simulation-optimization method for critical infrastructure in smart transportation systems. Int. J. Commun. Syst. 33 (11), e4397.

Kurniawan, J., Syahra, S.G., Dewa, C.K., et al., 2018. Traffic congestion detection: learning from CCTV monitoring images using convolutional neural network. Procedia Comput. Sci. 144, 291–297.

Liang, Y., Cui, Z., Tian, Y., Chen, H., Wang, Y., 2018. A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation. Transp. Res. Rec. 2672 (45), 87–105.

Litman, T., 2016. Smart congestion relief: Comprehensive analysis of traffic congestion costs and congestion reduction benefits.

Liu, Y., Wu, H., 2017. Prediction of road traffic congestion based on random forest. In: 2017 10th International Symposium on Computational Intelligence and Design, Vol. 2. ISCID, IEEE, pp. 361–364.

Luo, X., Li, D., Yang, Y., Zhang, S., 2019. Spatiotemporal traffic flow prediction with KNN and LSTM. J. Adv. Transp. 2019.

Luo, R., Shi, Y., 2020. Analysis and optimization of supermarket operation mode based on queuing theory: Queuing and pricing of personalized service. In: Proceedings of the 2020 4th International Conference on Management Engineering, Software Engineering and Service Sciences. pp. 221–224.

Malim, M.R., Halim, F.A., Abd Rahman, S.S., 2019. Optimising traffic flow at a signalised intersection using simulation. Malays. J. Comput. 4 (2), 261–269.

Mihăiţă, A.S., Dupont, L., Camargo, M., 2018. Multi-objective traffic signal optimization using 3D mesoscopic simulation and evolutionary algorithms. Simul. Model. Pract. Theory 86, 120–138.

Ogata, Y., 1981. On Lewis' simulation method for point processes. IEEE Trans. Inform. Theory 27 (1), 23–31.

Pant, M., Zaheer, H., Garcia-Hernandez, L., Abraham, A., et al., 2020. Differential evolution: A review of more than two decades of research. Eng. Appl. Artif. Intell. 90, 103479.

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y., 2003. Review of road traffic control strategies. Proc. IEEE 91 (12), 2043–2067.

Poomrittigul, S., Koomsubsiri, A., Sasithong, P., Deenuch, D., Wuttisittikulkij, L., 2019. The simulation of queuing model for bangkok rapid transit train ticket system using Python. In: 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC). IEEE, pp. 1–4.

Raskar, C., Nema, S., IoT based real time traffic monitoring system using M/G/1 queuing.

Requia, W.J., Higgins, C.D., Adams, M.D., Mohamed, M., Koutrakis, P., 2018. The health impacts of weekday traffic: A health risk assessment of PM2. 5 emissions during congested periods. Environ. Int. 111, 164–176.

Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. Annu. Rev. Control 43, 128–151.

Shortle, J.F., Thompson, J.M., Gross, D., Harris, C.M., 2018. Fundamentals of Queueing Theory, Vol. 399. John Wiley & Sons.

Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.

Sun, S., Chen, Q., 2008. Kernel regression with a Mahalanobis metric for short-term traffic flow forecasting. In: International Conference on Intelligent Data Engineering and Automated Learning. Springer, pp. 9–16.

Sun, X., Muñoz, L., Horowitz, R., 2003. Highway traffic state estimation using improved mixture Kalman filters for effective ramp metering control. In: 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475), Vol. 6. IEEE, pp. 6333–6338.

Sundarapandian, V., 2009. Probability, Statistics and Queuing Theory. PHI Learning Pvt. Ltd..

Swathi, N., Vehicular traffic at signalized intersection using queuing theory.

Teklu, F., Sumalee, A., Watling, D., 2007. A genetic algorithm approach for optimizing traffic control signals considering routing. Comput.-Aided Civ. Infrastruct. Eng. 22 (1), 31–43.

Wang, Z., Su, X., Ding, Z., 2020. Long-term traffic prediction based on lstm encoder-decoder architecture. IEEE Trans. Intell. Transp. Syst..

Wei, W., Wu, H., Ma, H., 2019. An autoencoder and LSTM-based traffic flow prediction method. Sensors 19 (13), 2946.

Yang, S., Yang, X., 2014. The application of the queuing theory in the traffic flow of intersection. Int. J. Math. Comput. Sci. 8, 986–989.

Yildirimoglu, M., Geroliminis, N., 2013. Experienced travel time prediction for congested freeways. Transp. Res. B 53, 45–63.

Yu, Y., Si, X., Hu, C., Zhang, J., 2019. A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. 31 (7), 1235–1270.

Zhang, Y., Zhang, Y., Haghani, A., 2014. A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. Transp. Res. C 43, 65–78.

Zhao, J., Gao, Y., Bai, Z., Wang, H., Lu, S., 2019. Traffic speed prediction under non-recurrent congestion: Based on LSTM method and BeiDou navigation satellite system data. IEEE Intell. Transp. Syst. Mag. 11 (2), 70–81.

Zhou, J., Chang, H., Cheng, X., Zhao, X., 2020. A multiscale and high-precision LSTM-GASVR short-term traffic flow prediction model. Complexity 2020.