DwaRa: A Deep Learning-Based Dynamic Toll Pricing Scheme for Intelligent Transportation Systems

Arpit Shukla, Pronaya Bhattacharya, Sudeep Tanwar, *Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—In Internet-of-Vehicles (IoV) ecosystems, intelligent toll gates (ITGs) connect nearby metropolitan cities through smart highways. At ITGs, existing solutions integrate blockchain (BC) and deep-learning schemes to leverage trusted and responsive analytics support for connected smart vehicles (CSVs) at ITGs. BC eliminates third-party intermediaries, and secures payments between vehicle owners (VO) and governing authorities (GA). Deep-Learning, on the other hand, facilitates accurate predictions for diverse and complex urban traffic conditions. However, due to fixed toll pricing schemes based on connected smart vehicles (CSV) type, VOs suffer from variable delays at different lanes due to dynamic congestion scenarios. To address the research gaps of such a fixed pricing schemes, we propose a BC-envisioned scheme *DwaRa*, that operates in three phases. In the first phase, future traffic is predicted based on Markov queues to balance the congestion at different lanes at ITGs efficiently. Then, we propose a novel spatially induced-long-short term memory (SI-LSTM) model to predict current traffic and weather based on historical repositories. Second, based on inputs by the Markov model, *SI-LSTM*, lane type, and vehicle type, a dynamic pricing algorithm is presented to improve the quality of experience (QoE) of the VO. Finally, based on dynamic price fixation between the VO and the GA, smart contracts (SCs) are executed and transactional data is secured through BC. The proposed scheme is compared against parameters like average mean-squared error (MSE), predicted traffic, scalability, interplanetary file system (IPFS) storage, computation (CC), and communication cost (CCM). At $n = 100$ test samples, and arrival rate $\beta = 80$, the obtained MSE is 0.0012, with a peak average value of 0.00526. The overall CC is 45.88 milliseconds (ms) and CCM is 53 bytes that indicate the proposed scheme efficacy against conventional approaches.**

***Index Terms*—Intelligent transportation systems, deep-learning, blockchain, Markov queues, smart contracts, IPFS.**

Manuscript received March 17, 2020; revised July 1, 2020; accepted August 31, 2020. Date of publication September 7, 2020; date of current version November 12, 2020. The review of this article was coordinated by Dr. F. Tang.
(Corresponding author: Neeraj Kumar.)

Arpit Shukla, Pronaya Bhattacharya, and Sudeep Tanwar are with the Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat 382481, India (e-mail: 18bce370@nirmauni.ac.in; pronoya.bhattacharya@nirmauni.ac.in; sudeep.tanwar@nirmauni.ac.in).

Neeraj Kumar is with the Thapar Institute of Engineering and Technology, Deemed to be University, Patiala 147002, India, with the King Abdulaziz University, Jeddah 11663, Saudi Arabia, and also with the Department of Computer Science and Information Engineering, Asia University, Taichung City 41354, Taiwan (e-mail: neeraj.kumar@thapar.edu).

Mohsen Guizani is with the Department of Computer Science and Engineering, Qatar University, Doha 2713, Qatar (e-mail: mguizani@ieee.org).
Digital Object Identifier 10.1109/TVT.2020.3022168

I. Introduction

In the past decade, with rapid urbanization and proliferation of IoV, ITGs have emerged as a responsive and automated solution for toll payments at ITGs. To facilitate toll collections, ETC technology is widely applied that leverages automated payment collection through sensor monitoring nodes. The sensors communicate through low-powered wireless channels and gather CSVs data, toll utilization, and HOV lane occupancy based on radio transponder units [1]. At the point when CSV passes a toll unit, sensor readings are recorded and are mapped to the vehicle registration number and lane utilization that facilitates automated and seamless fund transfer between VO and GA through third-party gateways. Although real-time toll collection through ETC allows low-latency, user QoE, and fewer queue-buildups at ITGs, they suffer challenges of security, privacy, and location-based attacks on VO sensitive data in IoV-based ecosystems. Any malicious intruder can launch sensor attacks like- jamming, Sybil, and DDoS and report incorrect updates at ITGs, which results in financial losses to both VO and GA. According to Cybersecurity Ventures, cyber-crime attacks costs have doubled to $ *6 trillion* by 2021 from $ *3 trillion* in 2015 [2]. Moreover, ETC also suffers from network limitations like- optimal route utilization, long outstanding queues of CSVs, fixed toll pricing schemes for all CSVs, higher waiting time, variable delays, traffic congestion at toll gates, and complex payment mechanisms. Thus, there is a need to address the aforementioned issues of ITG and design an efficient, robust, and scalable ITGs for improved ITS. This ensures seamless and automated dynamic payments between CSVs and GA.

The limitations of optimal route utilization and traffic congestion at ITGs are addressed through balancing loads at real-time by Markov queue-estimation models. These models analyze the input traffic distribution and output future traffic estimation to handle dynamic fluctuation in traffic loads. Thus, lane-traffic is properly managed at ITGs. To correctly predict the dynamic pricing between VO and GA, in addition to lane-management, historical traffic generated by embedded sensor units installed at CSVs, RSU, and ITG systems also needs to be analyzed in parallel. Traditionally, researchers have focused on ML/ DL models to predict input conditions to dynamic pricing schemes. However, as traffic datasets are updated in real-time, traditional ML/DL algorithms become inefficient over a while to indicate correct input dependent parameters. For time series prediction of traffic, LSTM is a preferred choice among researchers, as it solves long-term dependencies with ahead sequence of steps. However, basic LSTM is not appropriate to handle the multi-step traffic prediction as it contains spatial dependencies along with time. Hence, advanced DL algorithms are required to model multi-variant time-series predictions as parameters to a dynamic pricing scheme.

Thus, based on the inputs generated by queue-estimation and advance DL approaches, a dynamic toll pricing mechanism is applicable for ITG between VO and GA. It allows price fixation based on Type of Vehicle, lane-type, and services offered by different lanes, queue-estimation of future traffic, and historical time-series spatial predictions. The dynamic pricing scheme minimizes the overall traffic congestion at ITG and optimizes real-time responsive traffic management. Once the dynamic price is fixed between the VO and GA, SC facilitates the secure transfer of funds in BC without the involvement of third-party gateways. The transactional records are chronologically stored in BC that allows decentralized trust among ITG stakeholders- VOs, ITG, and GA [3], [4]. Transactional data can be stored in the IPFS, which store large amounts of data rather than storing it in chain ledgers.Hence, only the metadata of transactions are stored in blocks, which allows more transaction to be added per block in the same time-quantum and improves the overall scalability of BC.

In literature, the author's proposed solutions to decentralize and automate the payments of smart toll systems through BC. For example, Dobre *et al.* [5] proposed a BC solution to protect CSVs data in IoV ecosystems, which was used for efficient navigation systems. Later on, Deshpande *et al.* [6] improved the work carried out by the authors in [5] on a permissioned BC to improve and enhance the safety-critical operations for CSVs. Wu *et al.* [7] proposed a software-defined networking switch over BC in IoT scenarios for application-aware workflows. Similarly, authors have proposed solutions for dynamic traffic estimation based on queuing models. For example, Phu *et al.* [8] proposed a two-layer lane model on IoV probe vehicles to propose queue estimates on arriving vehicles on lane queues. Results are simulate based on bi-directional microscopic traffic and *Vein* communication simulator to obtain accurate results. However, issues of communication asymmetry were not considered. Duan *et al.* [9] proposed a stacked auto-encoder model for learning the traffic flow parameters, and then trained the model using an iterative greedy step function. Authors in [10] proposed a LevenbergMarquardt model the Taguchi method for novel stacked auto-encoders to improve the accuracy of [9]. Kachroo *et al.* [11] proposed a dynamic pricing mechanism for congested IoVs to optimize toll-pricing using control theory approaches. Zhang *et al.* [12] improved the work of [11] by optimizing dynamic pricing for high occupancy toll lanes based on random forest and nested models. A list of acronyms and their descriptions is presented in Table I.

TABLE I
ABBREVIATIONS AND THEIR DESCRIPTIONS

Abbreviation	Description
BC	Blockchain
CC	Computation Cost
CCM	Communication Cost
CSVs	Connected Smart Vehicles
DDos	Distributed Denial-of-service
DL	Deep Learning
ETC	Electronic Toll Collection
GA	Government Authorities
GPRS	General Packet-Radio Service
HOV	High-Occupancy Vehicle
IoV	Internet-of-Vehicles
IPFS	InterPlanetary File System
ITGs	Intelligent Toll Gates
JSON	Java-Script Object Notation
MAE	Mean-Average Error
ML	Machine Learning
MSE	Mean-Square Error
QoE	Quality-of-Experience
QoS	Quality-of-Service
RFID	Radio-Frequency Identification
RSUs	Road-Side Units
SCs	Smart Contracts
SG	Smart Grids
SI-LSTM	Spatially Induced Long-Short-Term Memory
TPPG	Third-Party Payment Gateways
VOs	Vehicle Owners

A. Motivation

The above discussion highlightes the need of dynamic pricing schemes at ITGs to facilitate desired QoE to VOs. Researchers in earlier studies explored efficient solutions for smart toll collection [7], traffic-prediction through auto-encoders [9], [10], and dynamic pricing schemes [12] in isolation, but traffic models are complex and the inherent interrelations are not explored in detail. The proposed scheme, *DwaRa*, addresses the gaps in earlier study through integration of queue-estimation and DL models that form an effective and validated dynamic pricing strategy. The proposed scheme estimates future traffic arrivals based on Poisson arrivals to formulate a posterior probability. This allows ITGs to estimate in advance the traffic loads at various toll lanes, so heavy traffic at peak hours are distributed. Then, in parallel, to validate the traffic estimators, historical traffic and weather samples are discretized, sampled, and analyzed through SI-LSTM to predict future traffic for L steps. The inputs from queue models, SI-LSTM, and chosen lane type are fed to dynamic pricing algorithm that forms a validated, accurate, and dynamic pricing decision for VO in ITGs. Finally, based on price fixation, SCs are executed at lanes to facilitate automated payments among VO and GA.

B. Research Contributions

The contributions of the paper are summarized below.

- A Markov queue-estimation traffic model based on Jeffry prior Bayesian inference is proposed to predict traffic arrivals for efficient lane management.
- A novel *SI-LSTM* model is proposed on historical traffic and weather data-sets to predict the real-time responsive traffic.
- Queue-estimation, *SI-LSTM* outputs and lane-type fed as inputs to a dynamic pricing algorithm price fixation of toll payments at ITG.
- The agreed dynamic price is then transacted between VO and GA through SCs for automation of funds-transfer. Post

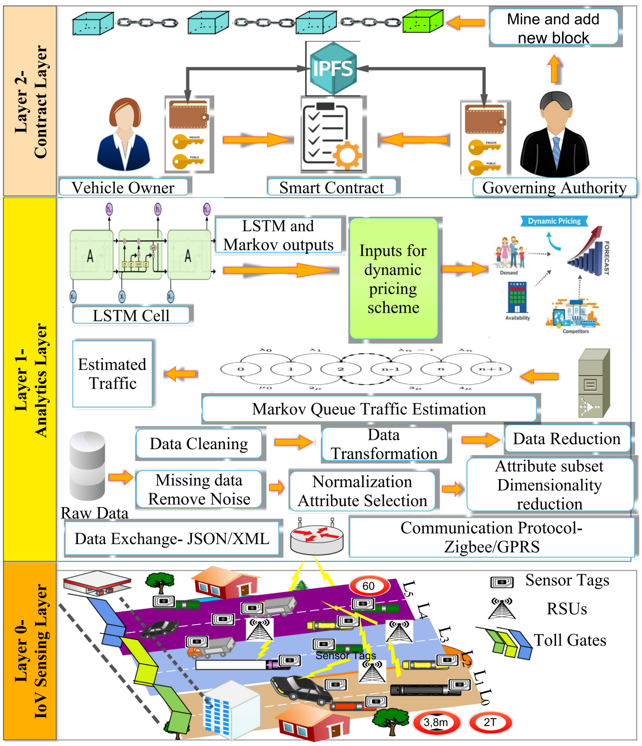


Fig. 1. *DwaRa:* System model.

execution of SCs, meta-information of the transaction is added in public BC as an immutable ledger.

C. Organization

The rest of the paper is organized as follows. Section II discusses the system model and the proposed scheme. Section III discusses the proposed scheme *DwaRa*, which integrates DL and BC to finalize dynamic toll pricing and secure funds transfer. Section IV discusses the performance evaluation and finally, Section V concludes the paper.

II. System Model and Problem Formulation

This section describes the system model and the problem formulation.

A. System Model

The integration of the DL-BC scheme, *DwaRa*, is proposed to fix the dynamic pricing and secure automated transfer of funds between the VO and GA, as shown in Fig. 1. The data flow in *DwaRa*, is from *Layer 0* to *Layer 2*. *Layer 0* is the IoV sensing layer, that consists of vehicular entities like CSVs, RSU, and ITGs. At toll lanes, RFID tags are scanned and transactional data is exchanged in JSON format through low-powered protocols like Zigbee and GPRS. JSON ensures uniformity and structured open constructs among different network entities. An outgoing switch/router R forwards gathered JSON tagged data for processing at *Layer 1*. In *Layer 1*, or analytics layer, the gathered data is processed in three sub-phases. In the first phase, the gathered raw data is pre-processed to remove missing values and outliers through *Pandas* library to generate categorical data. The categorical data is normalized and presented as inputs to Markovian traffic estimator model *M/M/1/*β that predicts an estimate of future traffic T_{est} derived through Jeffry's prior [13]. In parallel, historical vehicular and weather data ϕ_i is presented to a multi-variant time series prediction *SI-LSTM* model, to predict future steps of estimated values. Based on outputs T_{est} from *M/M/1/*β, and θ_i from *SI-LSTM*, a a dynamic pricing algorithm is presented for toll-price fixation that depends on lane-type L_{type}, and vehicle-type V_{type}. In *Layer 2*, or Contract layer, SCs are executed between VO and GA, based on dynamic pricing.

B. Problem Formulation

In *DwaRa*, there are three entities $E = \{E_{CSV}, E_{RSU}, E_{ITG}\}$. As depicted in Section II-A, at *Layer 0*, we consider p RSUs, as $E_{RSU} = \{R_1, R_2, R_3, \ldots, R_p\}$ with coverage ranges as $C = \{C_1, C_2, \ldots, C_p\}$. Inside C_p, the p^{th} RSU unit employs a smart low-powered RFID tagging mechanism for E_{CSV}. The exchange facilitates a robust vehicle-to-road infrastructure (V2R) communication infrastructure. E_{ITG} embeds IoT sensor units to read tags T attached to E_{CSV}.

E_{CSV} in p^{th} range contains q vehicles $V = \{V_1, V_2, V_3, \ldots, V_q\}$ embedded with RFID tags $T = \{T_1, T_2, T_3, \ldots, T_q\}$ that communicates inside C_p. The tags are designed to process payments with low-powered computations with E_{ITG} and payment information is processed by E_{RSU} in C_p. There are w E_{ITG} $T = \{I_1, I_2, I_3, \ldots, I_w\}$, where each of them have RFID scanners consisting of dipole antennas present to read tag information as sequences of electro-magnetic waves, transmitted through T_q. The total energy loss in reading q tags is defined as γ, where single unit loss for q^{th} tag is γ_q. Thus, in the proposed scheme *DwaRa*, $\frac{\gamma}{T_q}$ is minimized i.e. energy dissipated per tag. To achieve the same, we classify $\frac{\gamma}{T_q}$ into discrete energy slot units T_i, T_s, T_c as follows.

$$\begin{aligned} T_i &\leftarrow V_q \notin C_p \\ T_s &\leftarrow \exists V_q = 1 : V_q \in C_p \\ T_c &\leftarrow \exists V_q > 1 : V_q \in C_p \end{aligned} \tag{1}$$

where T_i is idle slot, T_s is single slot, and T_c is a collision slot. Based on the Eq. 1, the slot-efficiency η is defined as follows [14], [15].

$$\eta = \frac{T_c}{T_i + T_c + T_s} \tag{2}$$

To minimize $\frac{\gamma}{T_q}$ and maximize η_{max}, we compute probability that w tags occupy a slot T_c within a frame R as follows.

$$P_q(T_c, R) \leftarrow (^nC_q)(1/R)^q(1-(1/R))^{n-q} \tag{3}$$

In case of idle slot, i.e. $q = 0$, then $P_q(T_c, R)$ can be approximated as-

$$P_q(T_c, R) \approx (R)(e)^{-q} \tag{4}$$

In case of single-slot, i.e. $q = 1$, $P_q(T_c, R)$ is approximated as-

$$P_q(T_c, R) \approx (n)(1 - (1/R))^{n-1} \approx R \propto \frac{n/\alpha}{n/(\alpha - 1)} \tag{5}$$

where α is n/R. Each k^{th} tag is uniquely mapped to a q^{th} vehicle defined by a mapping $M : V \rightarrow T$. Any q^{th} vehicle with k^{th} tag is defined as V_q^k to be scanned at E_{ITG}. The set of w ITGs $\{I_1, I_2, \ldots, I_w\}$, operate in parallel as a multi-lane model, where, w^{th} ITG is sub-divided based on lane type, denoted as L_{type}. In *DwaRa*, a five layer multi-lane model is considered, defined as follows, $L_{type} = \{L_0, L_1, L_{2-4}\}$ where L_0 is the *best effort service* lane that provides equal services to all CSVs V_q. L_1 is *discounted pooling lane* that consists subset of E_{CSV} vehicles V_q with pooling condition for VOs (i.e., VOs > 2 / CSV). Lanes L_{2-4} are categorised as dynamic pricing lanes, which provides E_{CSV} with QoS choices. Any q^{th} VO can choose these lanes at variable-toll prices. L_2 is the reduced traffic lane to minimize delays, L_3 is Economy Lane to allow discounted rates, and L_4 is green lane for CSVs that are environment-friendly, i.e., emit fewer toxic emissions. A mapping is defined with parameters $P, M : L \rightarrow P$ between chosen L_{type} and vehicle type is denoted by V_{type}.

The gathered lane data is processed by R through low-powered IoT-ZigBee protocol stack. The chosen specifications are defined as $Z_{rs} = \{868 \text{ MHz}, 915 \text{ MHz}, 2.4 \text{ GHz}\}$ with data rate support of 250 kbps. The transferred data from E_{CSV} to E_{ITG} consists of following parameters $D_q = \{ID_{R_p}, V_q^k, W_q, C_k, T\}$ where ID_{R_p} is the unique identifier of $p^{th} E_{RSU}$, connected to E_{CSV}, V_q^k is $q^{th} E_{CSV}$ with k^{th} tag, W_q is the wallet of the q^{th} E_{CSV}, C_k is the computation cost of processing the k^{th} tag based on slot-efficiency η and T is the current timestamp.

Layer 1 consists of three sub-layers defined as AL_{sub}, which are as follows, $AL_{sub} = \{AL_{pproc}, AL_{pred}, AL_{dp}\}$ where, AL_{pproc} is the data pre-processing sub-layer, AL_{pred} is the traffic prediction sub-layer and AL_{dp} is the dynamic pricing sub-layer. In AL_{pproc}, the captured data from *Layer 0* is feature-scaled in linear fashion. The transformation is defined as follows.

$$U_i \leftarrow \frac{\iota - x_1}{x_2 - x_1}(y_2 - y_1) + y_1 \tag{6}$$

where ι is the raw data value, x_1 and x_2 are the minimum and maximum boundaries for original data and y_1 and y_2 are the new minimum and maximum boundaries defined for scaled data. For outlier removal, an inter quartile range (IQR) is defined as follows.

$$\begin{aligned} IQR &\leftarrow Q_2 - Q_1 \\ Q_1 &\leftarrow \zeta(D_{lh}) \\ Q_2 &\leftarrow \zeta(D_{uh}) \end{aligned} \tag{7}$$

where Q_1 and Q_2 stores respectively the lower $\zeta(D_{lh})$ and upper half $\zeta(D_{uh})$ of computed median (ζ). The IQR at AL_{pproc} is fed to AL_{pred} sub-layer as real-time traffic. AL_{pred} layer considers Poisson arrival of vehicles, denoted by β to be allocated simultaneously to the queue system. Based on β, likelihood information is computed on probability K simultaneous users, denoted by $L(\rho, \kappa)$. On $L(\rho, \kappa)$, we compute the posterior probability of future traffic estimation for efficient lane management, even at high loads. In parallel, historical traffic and weather samples are discretized, and sample length L for q^{th} CSV historical data is presented to SI-LSTM model. SI-LSTM predicts the traffic congestion for the next step, that is represented for every cell as follows.

$$\begin{aligned} \widehat{c_{fh}} &\leftarrow tanh(w_c[h_{ph}, x_{fh}] + b_c) \\ c_{fh} &\leftarrow (f_{fh} \times c_{fh-1}) + (i_t \times \widehat{c_{fh}}) \\ H_{nh} &\leftarrow O_g \times tanh(c^{nh}) \end{aligned} \tag{8}$$

where $\widehat{c_{fh}}$ represents LSTM cell state at current timestamp, c_{fh} is cell state at fixed hour timestamp (fh), w_c is the weight of cell c, h_{ph} is the output of LSTM block of previous hour (ph), x_{fh} is the input in the fixed hour, b_c represents the gate bias, f_{fh} is contents of forget gate, and i_t is the input gate and o_t is the output gate. Based on $\widehat{c_{fh}}$, predicted traffic T_{fh} is computed. Then, inputs $L(\rho, \kappa)$ and T_{fh} are presented to dynamic price algorithm that computes the dynamic price of q^{th} vehicle for next hour based on lane parameters, represented as follows.

$$DP_{fh} \leftarrow \xi(T_{fh}, V_{type}, L_{type}, GPS_{te}, Loc_{nat}) \tag{9}$$

where GPS_{te} is Global Positioning System distance and time estimate, and Loc_{nat} specifies the lat-long co-ordinates of native residence of VOs. Based on obtained DP_{fh}, a dynamic price DP_q^w is obtained as agreed dynamic price among q^{th} E_{CSV} and w^{th} ITG in a servicing C_p. For execution of SCs, public/private key pairs of VO and GA are fetched as $IPFS_{hash}$ as follows.

$$\begin{aligned} IPFS_{H_{VO}} &\leftarrow \{PU_{VO}, PR_{VO}\} \\ IPFS_{H_{GA}} &\leftarrow \{PU_{GA}, PR_{GA}\} \end{aligned} \tag{10}$$

where $\{PU_{VO}, PR_{VO}, PU_{GA}, PR_{GA}\}$ are public/private key pairs of VO and GA, respectively. As storing SCs data on BC ledger is costly, IPFS is used to store identification data pertaining to ledgers of VO and GA, respectively. Post execution of SCs, W_{VO}, and W_{GA} are updated with transactional meta-data M as follows.

$$\begin{aligned} W_{VO} &\leftarrow \{A_{VO}, T_q, C_p, \gamma, N\} \\ W_{GA} &\leftarrow \{A_{GA}, T_q, C_p, \gamma, N\} \end{aligned} \tag{11}$$

where A_{VO}, A_{GA} is the post-balance after execution of SCs and N denotes the random nonce identifier. The SCs are executed in p^{th} RSU unit as C_p. Details of W_{VO} and W_{GA} is updated in BC. The problem formulation P_f of *DwaRa* scheme is defined as follows.

$$P_f : max_{L(\rho,\kappa)}\{T_{fh}, DP_{fh}, V_q, C_p, T_w, \eta\} \tag{12}$$

s.t.

$$\begin{aligned} &C_1 : y_1 < \iota < y_2 \\ &C_2 : \widehat{c_{fh}} > \widehat{f_{th}} \\ &C_3 : \{W_{VO}, W_{GA}\} \geq 0 \\ &C_4 : T(M) > N \end{aligned} \tag{13}$$

Constraint C_1 dictates raw data values to within IQR to prevent outliers, C_2 specifies current cell-state timestamp to be greater than fixed hour timestamp for accurate *SI-LSTM* prediction, C_3 specifies wallets to be non-empty for SCs execution, and C_4 specifies transactional meta-data M stored in BC has time-stamp larger than wallet nonce N to prevent replay attacks.

III. *DwaRa*: The Proposed Scheme

As indicated in Section II-B, $L(\rho, \kappa)$ and T_{fh}, and the dynamic price DP_{fh} are calculated. Then, based on the obtained DP_{fh}, SCs are executed from W_{VO} and W_{GA}, respectively by fetching keys from $IPFS_{H_{VO}}$ and $IPFS_{H_{GA}}$. Details of the proposed scheme are ss follows.

A. DwaRa: Markov Queue Traffic Estimation

In *DwaRa*, to estimate the future traffic $L(\rho, \kappa)$, we consider Poisson distribution for arrival of q vehicles by E_{CSV} with rate β as $M/M/1/\beta$. As indicated, the service time at E_{ITG} is also Poisson with single server. The probability of K users in a single-server system is denoted as $\kappa = \{0, 1, 2, \ldots, \beta\}$ and represented as follows [13].

$$P(K=\kappa) \equiv P_\kappa = \begin{cases} \frac{\rho^\kappa(1-\kappa)}{1-\rho^{\beta+1}} & , for\ \rho \neq 1 \\ \frac{1}{\beta+1} & , for\ \rho = 1 \end{cases} \tag{14}$$

where ρ denotes the overall q E_{CSV} traffic intensity approaching towards w tolls in E_{ITG}. Then, based on κ, we estimate the likelihood traffic estimate function as follows.

$$L(\rho, \kappa) = \frac{\rho^y \times (1-\rho)^n}{(1-\rho^{\beta+1})^n} \tag{15}$$

where $y = \sum_{i=1}^{n} \kappa_i$ and n is the considered sample size. The queue model ensures relaibility in traffic predictions through Jeffrey's non-informative prior distribution [16], that allows prior information of unknown parameters in the model. To formulate the same, Fisher information $I(\rho)$ represented as follows.

$$I(\rho) = E\left[-\frac{\partial^2 \, log\, p\,(K|\rho)}{\partial\, \rho^2}\right] \tag{16}$$

Prior distribution based on $p(\rho)$ is proportional to fisher information $I(\rho)$ as follows.

$$p(\rho) \propto [I(\rho)]^{1/2} \tag{17}$$

To extract entropy of fisher information, we compute conditional probability $logp\ (K|\rho)$. Then, $I(\rho)$ is defined as follows.

$$I(\rho) = \frac{E[K]}{\rho^2} - \frac{1}{(1-\rho)^2} - \frac{(\beta+1)(\beta\rho^{-\beta-1}+1)}{(\rho^{-\beta}-\rho)^2} \tag{18}$$

where $E[K]$ is the expectation value computed as follows.

$$E[k] = \frac{\rho}{1-\rho} - \frac{(\beta+1)\rho^{\beta+1}}{1-\rho^{\beta+1}} \tag{19}$$

Algorithm 1: *DwaRa:* Bayes Traffic Estimate Queuing Model.

Input: Arrival rate of q vehicles β, traffic intensity ρ, and sample size n.
Output: $L(\rho, \kappa)$ in terms of future mean estimate est_{mean} and variance est_{var}

```
procedure SAMPLE_GENERATION n, ρ, β
  T ←Exec_if(n, 0, 1)
  C ← 1 − ρ^(β+1)
  LRho ← log(ρ)
  κ ← log(1 − T × c)%/%LRho
end procedure
procedure
 BAYES_INFERENCE_JEFF_PRIOR Sample, β, ρ, n
  Sumxi ← Σ_{i=1}^{n} x_i
  Log_posterior ← (Sumxi) × (log(ρ)) + n × log(1 −
  ρ) − n × log(1 − ρ^(β+1))
  I_rho ←
  (1/ρ^2) × (ρ/(1 − ρ) − (β + 1) × ρ^(β+1)/(1 − ρ^(β+1))) −
  1/(1 − ρ)^2 − (β + 1) × (β × ρ^(−β−1) + 1)/(ρ^(−β) − ρ)^2
  if (!(nan(I_rho))/AndI_rho > 0) then
    Log_posterior ← Log_posterior + 0.5 × log(I_rho)
  end if
  return Log_posterior
end procedure
procedure
 MONTE_CARLO_SIMULATION n, ρ, β, Final_est
  Epoch ← 10000
  Sample ← numeric(n)
  est ← numeric(Epoch)
  Iterator ← 1
  while (Iterator ≤ Epoch) do
    Sample ← SAMPLE_GENERATION(n, ρ, β)
    Seed_old ← Global_env.randomSeed
    est_Iterator ← Final_est
    Global_env.randomSeed ← Seed_old
  end while
  return c(est_mean, est_var)
end procedure
```

Based on $E[K]$, the posterior probability distribution $p_2(\rho|\kappa)$ for future traffic arrival is computed as follows.

$$\begin{aligned} p_2(\rho|\kappa) &\propto L(\rho, \kappa) \times p(\rho) \\ &\propto \frac{\rho^y \times (1-\rho)^n}{(1-\rho^{\beta+1})^n} \\ \times \Bigg[\frac{1}{\rho^2} &\left(\frac{\rho}{1-\rho} - \frac{(\beta+1)\rho^{\beta+1}}{1-\rho^{\beta+1}}\right) - \frac{1}{(1-\rho)^2} \\ &- \frac{(\beta+1)(\beta\rho^{-\beta-1}+1)}{(\rho^{-\beta}-\rho)^2}\Bigg]^{1/2} \end{aligned} \tag{20}$$

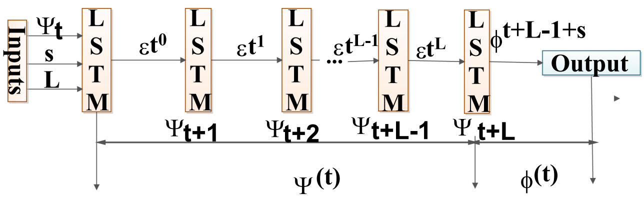


Fig. 2. *DwaRa:* SI-LSTM flow for multi-step prediction.

where $0< \rho <1$. $p_2(\rho|\kappa)$ is fed as input $L(\rho,\kappa)$ to predict future arrival rate of q vehicles based on L_{type}. As raw data ι is processed through Fisher information, scaled traffic is maintained among the proposed window boundaries $\{y_1, y_2\}$. Hence, the constraint C_1 is satisfied in P_f. The details of Bayes queue estimate are presented in Algorithm 1.

B. DwaRa: SI-LSTM Model for Multi-Step Traffic and Weather Prediction

To model the historical traffic and weather samples based on $\widehat{c_{fh}}$ as described in section II-B, a multi-variant time series prediction *SI-LSTM* is proposed, which involves spatial dependencies on historical samples that predict current traffic and weather conditions for the next step, based on defined step-size S. The samples are collected from traffic datasets with arrival and departure times of VOs, and vehicle counts. For weather dataset, temperature, humidity readings are recorded as sorted timestamps. These historical samples are broken into discrete smaller sub-samples through sequence-split method that improve the overall learning rate of *SI-LSTM*. Consider a sub-sample ψ_l^{th} input sequence of length L with output prediction after L steps as θ_l^{th}.

$$\begin{aligned} \psi^{(l)} &= [\psi_l, \psi_{l+1}, \psi_{l+2}, \ldots, \psi_{l+L-1}] \\ \theta^{(l)} &= \psi_{l+L-1+s} \end{aligned} \tag{21}$$

where $\psi^{(l)} \in \mathbb{Z}^L$ and $\theta^{(l)} \in \mathbb{R}$. The multi-step variable is denoted as s to indicate the number of steps ahead to be predicted.

The flow of the designed *SI-LSTM* model is shown in Fig. 2. The data is processed in a sequential manner with cell-state values $\widehat{c_{fh}}$ are updated based on weight w_c. The intermediate cell information is represented as ϵt^i with $0 \leq i \leq L$. The model processes ϵt^i from previous cell in a recurrent hidden step as h_{ph} to denote the output from the previous hour. A dense layer has been utilized for predicting the future behaviour of the system as follows.

$$\widehat{\theta}^{(l)} = (\omega_d^L \times \psi_{l+L-1+s}) + \varrho_d \tag{22}$$

where $\omega_d^L \in \mathbb{R}^{n\times 1}$ represents the weight and $\varrho_d \in \mathbb{R}$ represents bias of the final dense layer of SI-LSTM.

Then, the cell state space assuming $\psi^{(\lambda)}$ as unseen sequence is as follows.

$$\begin{aligned} C_{l,\lambda} &= f(c_{l-1,\lambda}, h_{l-1,\lambda}, \psi_l; \omega_{lstm,\lambda}, \varrho_{lstm,\lambda}) \\ F_{l,\lambda} &= g(h_{l-1,\lambda}, c_{l-1,\lambda}, \psi_l; \omega_{lstm,\lambda}, \varrho_{lstm,\lambda})) \end{aligned} \tag{23}$$

where ω and ϱ are model parameters, which depend on new given data point $\psi^{(\lambda)}$. The output Φ^t is prediction after dense layer based on spatial similarities in feature and test point vectors to compute optimized-cost function $\widehat{\theta}_\lambda^{(l)}$ as follows.

$$\widehat{\theta}_\lambda^{(l)} = \left(\omega_{d,\lambda}^L \times \psi_{l+L-1+s}\right) + \varrho_{d,\lambda} \tag{24}$$

Model parameters ω and ϱ defined in Eq. (23) are updated as follows.

$$\begin{aligned} (\widehat{\omega}_{lstm,\lambda}, \widehat{\omega}_{d,\lambda}, \widehat{\varrho}_{lstm,\lambda}, \widehat{\varrho}_{d,\lambda}) &= (\widehat{\omega}_\lambda, \widehat{\varrho}_\lambda) = min_{\omega_\lambda,\varrho_\lambda} J \\ J_\lambda &= \frac{1}{L}\sum_{l=1}^{L} s_{l,\lambda}(\widehat{\theta}_\lambda^{(l)} - \theta^{(l)})^2 + \Omega_\lambda \omega_\lambda^T \omega_\lambda \end{aligned} \tag{25}$$

where $s_{l,\lambda} \in \mathbb{R}^+$ is assumed to be similarity between L length sequences with updated model parameters refer to new point λ. The hidden state information is updated for a new point $\psi^{(\lambda)}$. Based on h_{ph} for each point i in ϵt^i at time-sequence t, the final prediction is defined as follows.

$$\widehat{h}_{l,\lambda} = g\left(\widehat{h}_{l-1,\lambda}, c_{l-1,\lambda}, \widehat{\psi}_l^{(\lambda)}; \widehat{\omega}_{lstm,\lambda}, \widehat{\varrho}_{lstm,\lambda}\right) \tag{26}$$

The final prediction $\widehat{h}_{l,\lambda}$ is differentiated based on traffic and weather information, denoted by $(\widehat{t}_{l,\lambda})$ and traffic intensity $(\widehat{w}_{l,\lambda})$, respectively. Based on tuple information, $\{\widehat{h}_{l,\lambda}, \widehat{w}_{l,\lambda}\}$, T_{fh} are updated. Constraint C_2 as satisfied as cell-state $\widehat{c_{fh}}$ is processed before final prediction.

C. DwaRa: Proposed Dynamic Pricing Scheme

Based on generated values $L(\rho,\kappa)$, *SI-LSTM* $\{\widehat{h}_{l,\lambda}, \widehat{w}_{l,\lambda}\}$, and L_{type} as inputs, a dynamic pricing scheme is proposed for price fixation of variable toll-payments between E_{CSV} and E_{ITG} is proposed. The dynamic price DP_q^w depends on a predictive model, which computes a net function based on two functions-weight-mapping and weight-summation with inputs. The following is depicted in Fig. 3. The weight-mapping function considers a mapping from input set $I = \{I_1, I_2, I_3, I_4\}$ and weight-set $W = \{W_1, W_2, W_3, W_4\}$, denoted as map-function $M : I \leftarrow W$ where I set is mapped to $\{L(\rho,\kappa), \widehat{h}_{l,\lambda}, \widehat{w}_{l,\lambda}, L_{type}\}$. The proposed Algorithm 2 calls three sub-procedures. The mapping M is depicted as *WEIGHT_ASSIGNMENT* sub-procedure. Then, based on I_4, sub- procedure *LANE_DISCRIMINATOR* is called that chooses L_{type} based on E_{CSV} choice with defined constraints for each lane-type. A base-price B is fixed $\forall L_{type}$. VO chooses a particular L_{type}. The lane-discriminator variable D_L is mapped with L_{type} and initial-base-price B is selected according to chosen lane (CL) from L_{type}. The process is iterated for each incoming vehicle V_q and VO is notified of B for L_{type}. Finally, based on real-time traffic volumes τ_1 and τ_2, sub-procedure *DYNAMIC_PRICE_FIXATION* is called. It computes the dynamic price of q^{th} vehicle based on mapping M. Once B_L is fixed, the value is updated in D_L and net-summation is performed by adding bias based on mean-estimated traffic μ_q for q^{th} vehicle. The summation computes the total weight as follows.

$$D_L \leftarrow \sum_{i=1}^{4} I_i.w_i \tag{27}$$

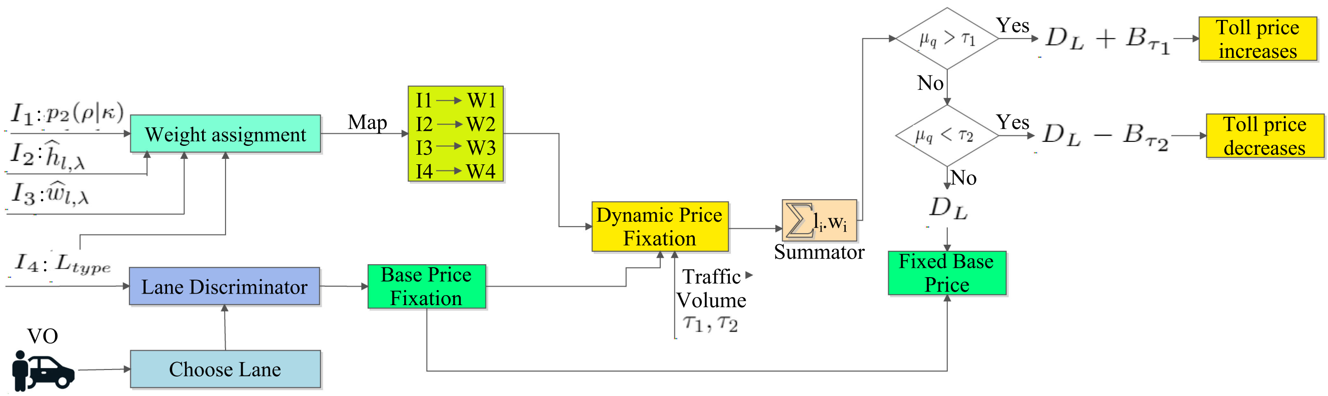


Fig. 3. *DwaRa:* Dynamic pricing strategy.

The bias μ_q is compared with traffic volumes for maximal and minimal threshold points τ_1 and τ_2, respectively and defined as follows.

$$\tau \leftarrow max_{lnD_L} = max\left(\sum_{i=1}^{q} lnM(V_q, \theta_1, \theta_2, \ldots, \theta_q)\right) \quad (28)$$

where $\{\theta_1, \theta_2, \ldots, \theta_q\}$ are likelihood estimators those are passed through an objective function O_τ that either maximizes or minimizes the estimate. In case $O_\tau \leftarrow max(J_\lambda)$, we obtain τ_1, and $O_\tau \leftarrow min(J_\lambda)$, we obtain τ_2, respectively. Then, checking conditions on comparison of μ_q with τ_1 and τ_2 are imposed to compute DP_q^w. DP_q^w increases by B_{τ_1} on fixed price D_L if $\mu_q > \tau_1$ and decreases vice-versa to meet the demands and traffic congestion of respective lanes.

D. Secure and Automated Payments in DwaRa

In Algorithm 2, the post-fixation of DP_q^w, SCs are called to initiate toll payment transfer. Algorithm 3 presents the contract structure to automate payments between VO and GA. Dynamic price is first stored in T_{dp} to initiate the transfer. Firstly, entity role $R = \{VO, GA\}$, is registered based on registration request. The registration requires them to fetch keys from IPFS ledger as stored hashes, $IPFS_{H_{VO}}$ and $IPFS_{H_{GA}}$ for authentication of stakeholders. Once successful registration is done, a VO becomes eligible to pay a toll based on DP_q^w. Constraint C_3 is satisfied as SCs execution is initiated only when sufficient funds are present in W_{VO} and W_{GA}. During the payment of DP_q^w, if transacting amount is less than DP_q^w, a variable Pay_{left} is invoked to ensure the left amount $R_{vo \leftarrow amount}$ is stored. The VO can pay back the debt amount ($R_{vo}.PD_{amount}$) during next transaction, but if $R_{vo}.PD_{amount}$ exceeds the threshold amount (THD_{amount}) decided by the GA, then the wallet of q^{th} VO will be blacklisted. Subsequent transactions from W_{VO} will be frozen till pending debts are paid, based on transactional metadata M. M will be updated since the last successful transaction nonce N. Hence, constraint C_4 is satisfied and the payment system is trust-worthy and secure against replay attacks. If debts are paid, q^{th} black-listed VO state changes to *ACTIVE* state, and nonce N is updated to reflect the same.

IV. Performance Evaluation

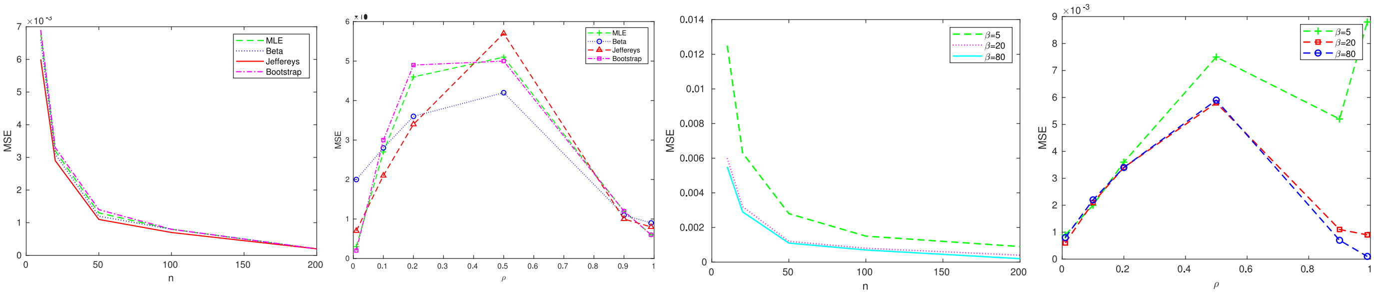
In this section, the performance evaluation of *DwaRa* scheme is discussed based on simulation results of Jeffery's queue estimation, *SI-LSTM* model, storage requirements in IPFS, and security evaluations.

A. Simulation Results

1) Dataset Description and Experimental Setup: For traffic prediction through *SI-LSTM* model, a real-world dataset of New York State Thruway Authority is considered [17]. The size of the dataset is *671* megabytes with multivariate features of toll entry and exit points. The data is considered for the period of January 1, 2019 to December 31, 2019. For weather prediction, dataset from National Oceanic and Atmospheric Administration for average temperature estimation is taken with data ranging from January 2013 to December 2019 [18].

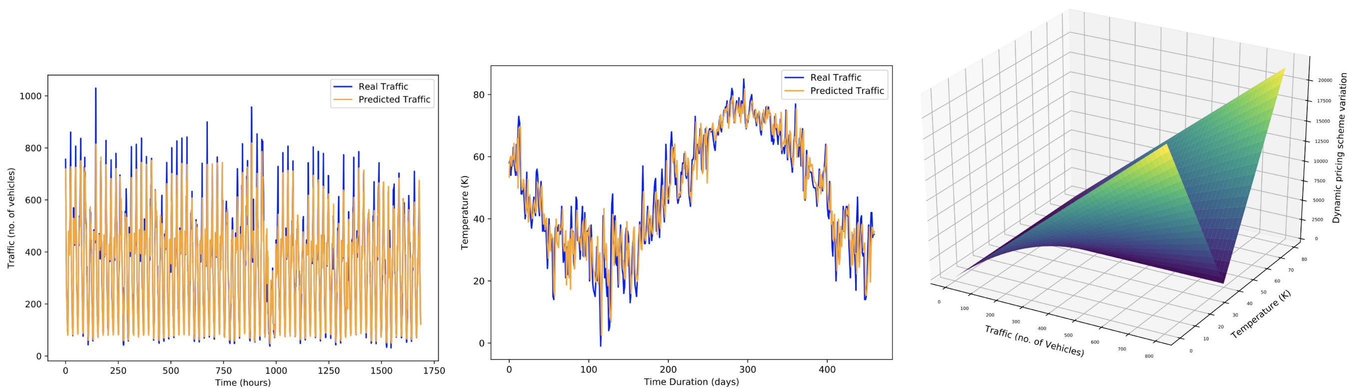
2) Experimental Setup: For Jeffery queue estimation, *Matlab Online R2019b v 9.7* is used. New-York toll dataset and Oceanic dataset for *SI-LSTM* is pre-processed through *Pandas v1.0.2* and *NumPy v1.8.2* library packages in python. After pre-processing, the LSTM model is structured using *Keras v2.3.0* with Tensor-flow package. The model is trained using *Tesla K80* graphics processing unit (GPU) with 12 GB RAM. Finally, the graphs are plotted using *Matlab Online R2019b v 9.7* tool. SCs are built on *Remix v0.9.4* and are deployed and tested on BC using the *Truffle* Suite with npm *v5.1.0*.

3) Traffic Estimation: To estimate traffic arrival towards E_{ITG}, we simulate Markov queue based on Bayesian Jeffery inference. The results of the same are presented in Fig. 4(a) and Fig. 4(b). Fig. 4(a) and 4(b) shows plot of mean square error (MSE) against considered sample size n and traffic intensity ρ. The figures indicate Jeffery's inference outperforms other conventional techniques for both n and ρ. This is because of Jefferys prior forms invariance against prior inference and posterior probability that reflects a close approximation of future estimates based on current indicators. Hence, $\rho_2(\rho|\kappa)$ is close to prior available fisher information $I(\rho)$. Further, Fig. 4(c) and 4(d) depicts the performance analysis of Jeffrey's inference to estimate V_q at different intensity β values. We consider a sample size n to be 1000, and simulate Monte Carlo randomization. The system utilization is computed as $\widehat{p} = 0.84053$ less than the



(a) (b) (c) (d)

Fig. 4. Performance analysis of Jeffrey's Queuing estimation in *DwaRa*. (a) Comparison of Jeffreys estimation along average MSE with n. (b) Comparison of Jeffreys estimation along average MSE with ρ. (c) Average MSE of different β with n. (d) Average MSE of different β with ρ.



(a) (b) (c)

Fig. 5. *DwaRa* SI-LSTM input prediction and Dynamic pricing scheme deviation. (a) Traffic prediction on test dataset. (b) Temperature prediction on test dataset. (c) DwaRa deviation of dynamic pricing scheme on traffic and weather.

set target value of 0.87. In general, the approximated prediction error is < 0.016 for sample size $n = 10$ and queue with $K \geq 30$. It is evident from the graph that as the traffic arrival rate β is increased with a larger sample size n, the average MSE is reduced. At $n=$ 100 samples, and $\beta=$ 80, the average value of obtained MSE is 0.0012. As the traffic intensity of ρ is increased, the MSE increases for different β values at first. At $= 0.5$, the peak average MSE is achieved with a value of 0.00526. Then, MSE reduces with increasing the value of ρ. It can be attributed to the fact that as n is kept constant, the obtained information $I(\rho)$ for $\rho_2(\rho|\kappa)$ is not accurate. As the entropy increases, $I(\rho)$ improves, and the likelihood estimate improves. This lowers the average MSE values.

4) SI-LSTM Predictions and Dynamic Pricing Scheme: The results of traditional traffic and weather estimates are presented based on the *SI-LSTM* model. Fig. 5(a) presents the original and predicted values based on the number of vehicles V_q. The model is trained for traffic and weather prediction $\widehat{t_{l,\lambda}}$ and $\widehat{w_{l,\lambda}}$ respectively with the previous 17 days as inputs and temperature vary from 0 to 80 Kelvin (K). The results are then tested on traffic prediction of 73 days and actual temperature, measured in K. Fig. 5(a) and 5(b) show the obtained results. The obtained results are compared with other similar approaches for values of MSE and MAE, depicted in Table II. As evident from the table, *SI-LSTM* outperforms other approaches as there is spatial and inductive dependency preservation with strong co-relation to location points $\psi^{(\lambda)}$. Fig. 5(c) presents the fluctuations in dynamic price behavior, based on inputs to the algorithm. As evident from the Fig. rise in $\widehat{t_{l,\lambda}}, \widehat{w_{l,\lambda}}$ increases the current traffic prediction for next L steps as output to the *SI-LSTM* model. The reflected change from the output of *SI-LSTM* updates the fisher information in Jefferey queue estimation, with a rise in incoming traffic arrival β. Thus, $L(\rho, \kappa)$ also increases, which indicates a high chance of congested lanes. In such scenarios, ITGs employ high dynamic prices on VOs, over the base price B_L based on chosen L_{type} by VO. Due to increased traffic, those VOs who opt for QoS lanes, have to pay more hike in base prices. General-purpose lanes would be congested, indicating high-waiting times at tolls. Thus, it exhibits a peer-profit strategy for both VO and GA. Thus, the pricing strategy exhibits a trade-off between waiting time and high cost to VO, that favors GA at ITGs.

B. Security Evaluation

To facilitate security evaluation of *DwaRa*, the cost values of CC and CCM of various identifiers follow the same as in [19]. The details are presented as follows.

Algorithm 2: *DwaRa:* Dynamic Pricing Scheme to Facilitate Toll Payments.

Input: Markov queue traffic estimate $p_2(\rho|\kappa)$, multi-step traffic and weather prediction by *SI-LSTM* $\widehat{t}_{l,\lambda}$ and $\widehat{w}_{l,\lambda}$ respectively, lane type L_{type}, traffic volume tau_1 and tau_2.
Output: DP_q^w
Initialization: i=1
procedure WEIGHT_ASSIGNMENT$p_2(\rho|\kappa)$, $\widehat{t}_{l,\lambda}$, $\widehat{w}_{l,\lambda}$, L_{type}
$\mu_q \leftarrow$ *MONTE_CARLO_SIMULATION*$(n, \rho, \beta, Final_{est})$
for $i \leftarrow 1$ to l **do**
$\widehat{t}_{i,\lambda} = g(\widehat{t}_{i-1,\lambda}, c_{i-1,\lambda}, \widehat{\psi}_l^{(\lambda)}; \widehat{\omega}_{lstm,\lambda}, \widehat{\varrho}_{lstm,\lambda})$
$\widehat{w}_{i,\lambda} = g(\widehat{w}_{i-1,\lambda}, c_{i-1,\lambda}, \widehat{\psi}_l^{(\lambda)}; \widehat{\omega}_{lstm,\lambda}, \widehat{\varrho}_{lstm,\lambda})$
end for
$L_{type} \leftarrow \{L_0, L_1, L_{2-4}\}$
W $\leftarrow \{w_1, w_2, \ldots, w_4\}$
$I_1 \leftarrow \mu_q$, $I_2 \leftarrow \widehat{t}_{l,\lambda}$
$I_3 \leftarrow \widehat{w}_{l,\lambda}$, $I_4 \leftarrow L_{type}$
$M \leftarrow$ *Map*$\{I_1 \leftarrow w_1, I_2 \leftarrow w_2, I_3 \leftarrow w_3, I_4 \leftarrow w_4\}$
end procedure
procedure LANE_DISCRIMINATORI_4, V_q
$V_q.[VO] \leftarrow$ Choose_Lane_Type(D_L, I_4)
$B \leftarrow$Initial_base_price(L_{type})
for $i \leftarrow 1$ to q **do**
if $V_i.[VO] > 2$ **then**
$D_L \leftarrow L_1$
$B_{L_1} \leftarrow$ *Initial_toll_price*(L_1)
else if $V_i.[VO] \leftarrow C_L(2,4)$ **then**
$D_L \leftarrow C_L$
$B_{C_L} \leftarrow$ *Initial_toll_price*(C_L)
else
$D_L \leftarrow L_0$
$B_{L_0} \leftarrow$ *Initial_toll_price*(L_0)
end if
end for
end procedure
procedure DYNAMIC_PRICE_FIXATIONM, τ_1, τ_2, μ_q
$B_L \leftarrow$ *LANE_DISCRIMINATOR*(I_4, V_q)
for $i \leftarrow 1$ to q **do**
for $j \leftarrow 1$ to w **do**
$D_L \leftarrow \sum_{i=1}^{4} I_i.w_i$
if $\mu_q > \tau_1$ **then**
$DP_q^w \leftarrow D_L + B_{\tau_1}$ |
$V_i.[VO]$ *NOTIFY("Increase toll prices due to high demand"*
else if $\mu_q < \tau_2$ **then**
$DP_q^w \leftarrow D_L - B_{\tau_2}$ |
$V_i.[VO]$ *NOTIFY("Toll prices reduced due to less congestion"*
else
$DP_q^w \leftarrow D_L$
end if
end for
end for
return DP_q^w
end procedure

Algorithm 3: Toll Payment Using Smart Contract.

Input: $R, THD_{amount}, W_i, IPFS_{hash}, Req_{vo}[msg.sender], T_{dp}$
Output: $Payment_{status}, Pay_{left}, T_{dp}, R \in \{R_{ta}, R_{vo}\}$
procedure TOLL_PAYMENT_SCR, W_i, THD_{amount}
if $(R \in R_{vo})$ **then**
if (R_{vo}.blacklisted == False) **then**
if $(R_{vo\leftarrow amount} < (R_{vo}.PD_{amount}$+$T_{dp}))$ **then**
Event $\leftarrow$ emit paymentSuccessful(R_{VO},false)
$R_{vo}.PD_{amount} \leftarrow R_{vo}.PD_{amount}$ + T_{dp} - msg.value
if $(R_{vo}.PD_{amount} > THD_{amount})$ **then** R_{vo}.blacklisted$\leftarrow$ True
end if
Event$\leftarrow$Emit_Event $Pay_{left}(R_{vo}.PD_{amount}$+$T_{dp}$-msg.value);
else
$R_{vo}.PD_{amount} \leftarrow 0$
Event$\leftarrow$Emit_Event paymentSuccessful(true)
end if
else
$R_{vo} \leftarrow$ NOTIFY("Suspended untill clearing debt.")
end if
if $(R_{vo}.PD_{amount} > 0)$ **then**
$R_{vo}.PD_{amount} \leftarrow R_{vo}.PD_{amount}$ - msg.value
if $(R_{vo}.PD_{amount} < THD_{amount})$ **then**
R_{vo}.blacklisted$\leftarrow$ False
Event$\leftarrow$Emit_Event $Pay_{left}(R_{vo}.PD_{amount})$;
end if
else
$R_{vo} \leftarrow$ NOTIFY("No pending amount left of the stakeholder.")
end if
else
$R_{ta} \leftarrow$ Request_Registration$(W_i, IPFS_{hash})$
$Req_{vo}[msg.sender] \leftarrow$ True
end if
if $(R \in R_{ta})$ **then**
set the current dynamic price of Toll.
$T_{dp} \leftarrow$ Set_T_{rate}(nRate)
else if $(R \in R_{ta}/And\ REQ_{vo}[msg.sender] \leftarrow$ False) **then**
$R_{vo} \leftarrow$ Grant_Registration()
$Req_{vo}[msg.sender] \leftarrow$ False
else
$R \leftarrow$ NOTIFY("Access not granted, not authority.")
end if
end procedure

1) *Computation and Communication Cost:* To estimate the CC of *DwaRa*, we first evaluate Algorithm 1. It consists of sample generation with n, ρ, β as random nonces. Based on the initial seed value by function $Global_{env}.randomSeed()$ is 0.00032 seconds (sec). Hence cost required for Algorithm 1 is $\approx$ 0.00032 sec. Next, we evaluate the proposed Algorithm 2 to fix DP_q^w. It consists of one random nonce i.e. $\approx$ 0.00032

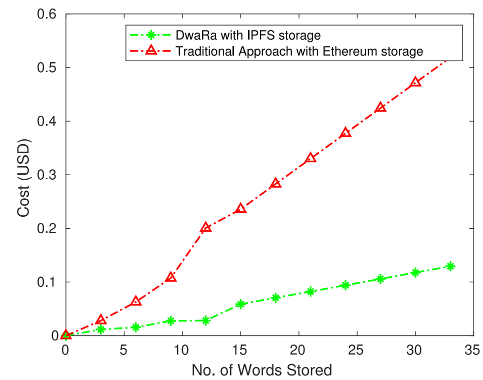
TABLE II
SI-LSTM PREDICTION COMPARISON WITH TRADITIONAL MODELS

Comparative Analysis		Weather		Traffic congestion	
		$\widehat{Y}_{train}$	$\widehat{Y}_{test}$	$\widehat{Y}_{train}$	$\widehat{Y}_{test}$
MSE	*LSTM*	36.198	39.918	273.374	300.213
	OT-LSTM	35.994	40.983	268.123	296.132
	SI-LSTM	35.249	41.274	270.318	294.213
MAE	*LSTM*	5.128	5.229	33.983	33.347
	OT-LSTM	4.582	4.723	30.948	28.945
	SI-LSTM	4.064	4.941	29.274	28.699

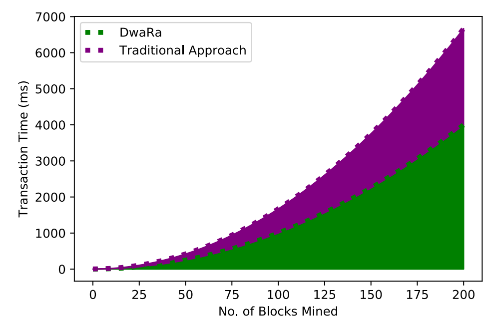
TABLE III
COMPARISON OF OVERALL COMPUTATION AND COMMUNICATION COST AGAINST EXISTING SCHEMES

Scheme	CC	CCM	ME
Hathaliya *et al.* [20]	$9H_m + 3E_{sym} + 4T_{eccm} \approx 96.64ms$	176 bytes	3
Kabra *et al.* [3]	$2E_{asym} + 1T_{signgen} + 2T_{signver} + H_m \approx$ 192.14 ms	203 bytes	4
Proposed *DwaRa*	$3H_m + 2nonce + 2T_{append} + 1E_{asym} \approx 45.88ms$	53 bytes	7

E_{asym}: Asymmetric encryption cost; H_m: Hash output cost; T_{pair}: Bilinear pairing cost; E_{sym}: Symmetric encryption cost; T_i : Transaction append cost; E_{eccm}: ECC encryption cost; S_m: Signing cost; V_m: Verification cost; $T_{signgen}$: Signature generation cost;$T_{signver}$: Signature verification cost, $nonce$: Time-stamp cost; $merkle-root$: timestamp cost to refer genesis block hash; T_{append}: Cost of appending blocks to chain.



(a)



(b)

Fig. 6. Comparative analysis of *DwaRa*. (a) Data storage cost comparison. (b) Scalability comparison in BC.

secs. Then, smart contracts are executed among VO and GA in Algorithm 3, based on W_i and THD_{amount}. W_{id} consists of identifiers for VO and GA, i.e. 2 message hashes, 1 asymmetric encryption $E_a(W_i)$, and 1 hash operation for creating $IPFS_{hash}$, 1 operation for timestamp and 1 block append each for E_{VO} and E_{GA}. Thus, the total cost of algorithm 3 is 2 $\times(0.00032 + 0.00032 + 0.0215)$ + 0.00032 + 0.00032 + 0.00032 $\approx$ 0.04524 secs. Thus, the overall computation cost of *DwaRa* is $0.00032 + 0.00032 + 0.04524 \approx 0.04588$ sec or 45.88 milliseconds.

Similar to the computation cost, the communication cost of *DwaRa* is calculated as follows. Algorithm 1 uses a 16 bits random nonce $Seed_{old}$ in Monte Carlo simulation. In Algorithm 2, map M requires 4 bits, choosing lane L_{type} requires 2 bits, and notify user is done twice i.e. 2 bits. For algorithm 3, W_i assignment for registration request between VO and GA consists of $\{W_{id}, W_{nonce}, W_{hashkey}\}$ occupying 32, 160 and 32 bits respectively. After registration, $IPFS_hash$ is stored and mapped with W_i consisting of 160 bits. Post-registration, conditions are checked based on R, R_{vo}.blacklisted$, R_{vo}.PD_{amount}$ and $R_{vo\leftarrow amount}$, where each check requires 1 bit. For publishing of events in BC, notifying the stakeholder in BC, 1 bit is required. In total, there are 3 events and 3 notify calls i.e. 6 bits. State transfer operation requires 8 bits. Finally, request registration, toll payment and remaining pay notification is 3 bits. Hence, the communication cost of Algorithm. 3 is $(32 + 160 + 32) + 160 + 6 + 8 \approx 398bits \approx 50bytes$. Thus, the total cost is $16 + 6 + 398 \approx 520$ bits, or 53 bytes. The obtained CC and CCM are compared with existing schemes to indicate efficacy of *DwaRa* scheme. The results are shown in Table III.

TABLE IV
COMPARATIVE ANALYSIS WITH EXISTING SCHEMES

Parameters	Deng *et al.* [21]	Wu *et al.*[7]	Qiu *et al.*[22]	Zhang *et al.* [12]	Proposed *DwaRa*
A1	✓	✓	✗	✓	✓
A2	✓	✓	✓	✗	✓
A3	✓	✓	-	-	✓
A4	✗	✗	✓	✓	✓
A5	✗	✗	✗	✗	✓
A6	✓	✗	-	✗	✓

A1: Decentralized; A2: Smart Toll; A3: Blockchain; A4: Queue estimation A5: Dynamic Pricing; A6: Smart Contracts; ✓ shows parameter is present; ✗ shows parameter is absent; & - shows parameter is not considered.

2) Efficiency of DwaRa Against Conventional Approaches: Fig. 6 shows the performance comparison of *DwaRa* scheme compared to conventional approaches addressed in terms of scalability and storage cost. Fig. 6(a) shows a plot depicting the storage cost of data in *DwaRa* in IPFS compared to the storage in traditional approaches in Ethereum BC. To formulate the same, we consider the number of machine-length words against cost units. As indicated in the graph, more words are stored at lower cost in IPFS, due to decentralized peer storage and cost-effective retrievals. Fig. 6(b) shows plot depicting scalability comparison of *DwaRa* with the approach. As only transactional metadata is stored in the BC due to IPFS storage, more transactions can be added per time quantum. This results in improved scalability. Finally, the proposed scheme is compared against traditional approaches based on selected parameters as defined in Table IV against state-of-the-art approaches.

V. Conclusion

The paper proposed a BC-envisioned DL scheme *DwaRa* that addresses the challenges of lane management, traffic estimation, and fixed pricing schemes at ITGs. To address the research gaps of fixed pricing, variable congestion delays at ITGs, and automation of payments between VOs and GA, authors have presented an integration of queue estimation models and DL. Through Poisson arrivals, future traffic at toll lanes is balanced, and in parallel, a *SI-LSTM* model is presented that takes historical traffic and weather samples, to predict current traffic at the next step, based on a sample length. Thus, the integration forms a validation of future traffic and based on chosen lane type, dynamic pricing strategies are proposed. Once price fixation is executed, the SC is executed through authorized key-pairs of VOs and GA fetched from IPFS. The proposed results indicate the efficacy of the proposed scheme against existing schemes. In the future, we plan to work on reducing the communication complexity through femtocell design. Smaller cells would allow effective communication among CSVs and balanced loads inside the spatial coverage of RSU at the same desired accuracy level.

References

[1] K. Kitajima, T. Higuchi, K. Terasaka, and K. Oshima, "On-board device, automatic toll collection machine, toll collection system, and toll collection method," U.S. Patent App. 16/080 487, Jun. 2020.

[2] "Cybercrime damages $6 trillion by 2021," 2016. Accessed: Mar. 7, 2020. [Online]. Available: https://cybersecurityventures.com/hackerpocalypsecybercrime-report-2016/

[3] N. Kabra, P. Bhattacharya, S. Tanwar, and S. Tyagi, "Mudrachain: Blockchain-based framework for automated cheque clearance in financial institutions," *Future Gener. Comput. Syst.*, vol. 102, pp. 574–587, 2020.

[4] A. Srivastava, P. Bhattacharya, A. Singh, A. Mathur, O. Prakash, and R. Pradhan, "A distributed credit transfer educational framework based on blockchain," in *Proc. IEEE 2nd Int. Conf. Adv. Comput., Control. Commun. Technol.*, Allahabad, India, 2018, pp. 54–59.

[5] L. Hrtan and C. Dobre, "Blockchain privacy-preservation in intelligent transportation systems," in *Proc. IEEE Int. Conf. Comput. Sci. Eng.*, Bucharest, Romania, Oct. 2018, pp. 177–184.

[6] V. Deshpande, L. George, and H. Badis, "Safe: A blockchain and secure element based framework for safeguarding smart vehicles," in *Proc. 12th IFIP Wireless Mobile Netw. Conf.*, Paris, France, Sep. 2019, pp. 181–188.

[7] J. Wu, M. Dong, K. Ota, J. Li, and W. Yang, "Application-aware consensus management for software-defined intelligent blockchain in IoT," *IEEE Netw.*, vol. 34, no. 1, pp. 69–75, Jan./Feb. 2020.

[8] C. N. Van Phu and N. Farhi, "Estimation of urban traffic state with probe vehicles," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.2975120.

[9] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Ap. 2015.

[10] H. Yang, T. S. Dillon, and Y. P. Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 28, pp. 2371–2381, Oct. 2017.

[11] P. Kachroo, S. Gupta, S. Agarwal, and K. Ozbay, "Optimal control for congestion pricing: Theory, simulation, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1234–1240, May 2017.

[12] K. Zhang, Z. Zhou, Y. Qi, and Y. Wang, "Dynamic pricing strategy for high occupancy toll lanes based on random forest and nested model," *IET Intell. Transp. Syst.*, vol. 13, no. 4, pp. 636–643, 2019.

[13] D.Gross and J. M. Thompson, *Fundamentals of Queueing Theory*. 4th ed. New York, NY, USA: Wiley, 2009.

[14] P. Bhattacharya, S. Tanwar, R. Shah, and A. Ladha, "Mobile edge computing-enabled blockchain framework—A survey," in *Proceedings of ICRIC 2019*, P. K. Singh, A. K. Kar, Y. Singh, M. H. Kolekar, and S. Tanwar, Eds., (Cham), Berlin, Germany: Springer, 2020, pp. 797–809.

[15] P. Bhattacharya, A. Singh, A. Kumar, A. K. Tiwari, and R. Srivastava, "Comparative study for proposed algorithm for all-optical network with negative acknowledgement (ao-nack)," in *Proc. 7th Int. Conf. Comput. Commun. Technol.*, New York, NY, USA, 2017, pp. 47.–51.

[16] F. R. B. Cruz, M. F. S. V. D'Angelo, and T. V. Woensel, "Traffic intensity estimation in finite markovian queueing systems," *Hindawi Math. Problems Eng.*, vol. 2018, pp. 1–15, Jun. 2018, doi: 10.1155/2018/3018758.

[17] N. Y. S. T. Authority, "Open data NY." 2019. Accessed: Jan. 2020. [Online]. Available: https://data.ny.gov/Transportation/NYS-Thruway-Origin-and-Destination-Points-for-All-/chzq-388p

[18] N. Oceanic and A. Administration, "NOAA," 2020. [Online]. Available: https://www.ncdc.noaa.gov/cdo-web/search

[19] J. Srinivas, A. K. Das, N. Kumar, and J. Rodrigues, "Cloud centric authentication for wearable healthcare monitoring system," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 942–956, Sep./Oct. 2020.

[20] J. J. Hathaliya, S. Tanwar, S. Tyagi, and N. Kumar, "Securing electronics healthcare records in healthcare 4.0: A biometric-based approach," *Comput. Elect. Eng.*, vol. 76, pp. 398–410, 2019.

[21] X. Deng and T. Gao, "Electronic payment schemes based on blockchain in vanets," *IEEE Access*, vol. 8, pp. 38296–38303, 2020.

[22] J. Qiu, L. Du, D. Zhang, S. Su, and Z. Tian, "Nei-TTE: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city," *IEEE Trans. Ind. Inform.*, vol. 16, no. 4, pp. 2659–2666, Apr. 2020.