Deep and Shallow Neural Networks for Blockchain in Edge Computing Environment

Samina Shiraj Mulani
BITS Pilani
Amit Dua
BITS Pilani
Sudeep Tanwar (sudeep.tanwar@nirmauni.ac.in)
Institute of Technology, Nirma University https://orcid.org/0000-0002-1776-4651
Neeraj Kumar
Thapar Institute of Engineering and Technology

Research Article

Keywords: Blockchain, Artificial Intelligence, Edge Computing, LSTM

Posted Date: April 29th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-462304/v1

Deep and Shallow Neural Networks for Blockchain in Edge Computing Environment

Samina Shiraj Mulani*, Amit Dua†, Sudeep Tanwar‡, Neeraj Kumar§
*† Department of Computer Science and Information Systems, BITS Pilani, Pilani (Rajasthan), India
*(e-mail: f20180314@pilani.bits-pilani.ac.in) † (e-mail: amit.dua@pilani.bits-pilani.ac.in)
‡Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India, (email: sudeep.tanwar@nirmauni.ac.in)
§Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India (email: neeraj.kumar@thapar.edu)

***Abstract*—With the rise in traffic congestion and associated costs, it becomes crucial to readily make available accurate traffic reports to the general public and also to predict the traffic levels to mitigate further congestion. Various tools and technologies have emerged to solve the aforementioned problem, which comprises of secure and accurate data collection, storage, utilisation of this data for the purpose of prediction, and making required data available to the public. Motivated from the aforementioned discussion, in this paper, various approaches to solve this larger puzzle have been discussed and analysed, and a holistic solution combining the power of blockchain, InterPlanetary File System (IPFS), and neural networks has been suggested. Simulation results show that an LSTM model with 50 time steps and 200 units in the hidden layer, followed by a dense layer leads to minimum Root Mean Square Error (RMSE) value, with a randomly generated but complete dataset. Security analysis of the proposed solutions shows its efficacy compared to state-of-the-art approaches.**

Keywords: Blockchain, Artificial Intelligence, Edge Computing, LSTM

I. INTRODUCTION

Increased traffic congestion has been a rising predicament, leading to several problems, including higher levels of pollution, wastage of fuel, sleep disturbance due to exposure to traffic noise, and surging road rage. The Indian city of Bengaluru bagged the top spot in the TomTom Traffic Index report of 2019 [2], with an average of 71% extra travel time stuck in traffic. 239 countries marked an increase in traffic congestion since 2018. Thus, it becomes imperative to accurately estimate congestion levels and provide them to the general public for beneficial usage.

In this paper, we tackle the problem of traffic information collection, prediction and making this data available to the public in real time. Several works employ Intelligent Transport Systems, which often make use of Vehicular Ad hoc Networks (VANETs) to collect information pertaining to vehicles on road. A VANET consists of groups of moving or stationary vehicles connected by a wireless network [3]. On Board Units (OBUs) installed on vehicles manage to communicate data like vehicle velocity, position, etc (obtained using attached sensors and technologies like GPS) to other vehicles and Road Side Units (RSUs). RSUs are fixed along road segments and can be used to communicate with servers that provide a service

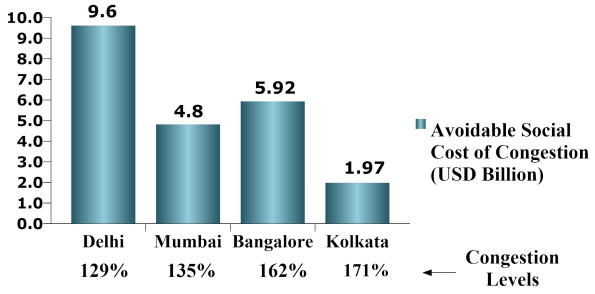

Fig. 1: Avoidable social cost of congestion (USD Billion) for 4 cities of India [1]

to the users (like traffic information, infotainment and so on). Drawbacks include infrastructure installation and maintenance, along with the potential loss of information due to the dynamic and mobile nature of the network. Other techniques employ usage of cameras followed by image processing techniques, on-road sensors, or crowdsourcing to capture relevant information to detect traffic levels. Crowdsourcing faces the problem of potential breach of privacy and lack of motivation amongst participants.

For traffic prediction, methods can be divided into two main classes – linear prediction and non-linear prediction. Linear prediction utilises traditional models like ARIMA, HoltWinters algorithm and so on. Non-linear prediction methods employ machine learning and deep learning models, which are known to be more effective than linear models. Machine learning and deep learning employ algorithms to analyse received data, learn from parsed data, and make the desired decisions. With abundant and cheap computation readily available in current times, such research is quickly expanding and improving. Often, training and re training the model periodically is computation intensive.

To enable one or more of the above techniques, some works employ a centralised approach for ease of collection of data and handling computation, a well-known example being Google Maps. However, they possess a single point of failure problem. To overcome this issue, blockchain technology is utilised. Blockchain is a technology, which can be used to eliminate the need of a trusted intermediary during non-

reversible transactions. It involves a distributed ledger in which all nodes have a copy of, making it logically centralised. Blocks of transactions are linked via cryptographic hashes, making the chain practically tamper resistant. Usage of public and private keys to sign transactions aids authenticity and privacy.

Distributed consensus protocols are used to agree on the next block to be added to the chain. Each protocol has its advantages and disadvantages. Proof of Work consensus offers complete decentralisation, but is computationally expensive. Protocols like Proof of Stake, Delegated Proof of Stake and Proof of Authority have elements of centralisation embedded in them despite offering greater speed. The trust model assumed plays a crucial role in determining the consensus algorithm. In VANETs, the trust model is usually of two types – entity centric and data centric. Entity centric models evaluate the trust level of individual vehicles. However, due to high mobility of vehicles, it is difficult to collect enough information to evaluate the real time reputation of a certain vehicle. Data centric models evaluate trust level of the data received, typically using content similarity. Some methods even utilise received signal strength, email-based social trust, etc.

Apart from aiding authentication, security and transparency, certain solutions use blockchain to incentivise users to share information (in crowdsourcing), detect traffic congestion and implement data or entity centric trust models. Since blockchain isn't efficient for storage of vast amounts of data, Inter Planetary File System (IPFS) has been used in certain solutions that aim to integrate IoT with blockchain. IPFS is a global distributed file system that uses a secure hash of contents as a file location identifier. Locations of a file are resolved using Distributed Hash Tables. By having the blockchains store only the IPFS file hashes of the data, storage space required is vastly reduced.

To inform users of the congestion level, some solutions use a central web server to display the traffic data collected and predicted, while others disseminate the messages to relevant users in the neighbouring areas. Message dissemination techniques in VANET can be broadly classified into single hope and multi hop. Single hop has the limitation of a smaller reach, while multi hop communication suffers from the problem of broadcast storm. Several methods aim to mitigate the latter issue by using counter based techniques, distance-based schemes and so on. The European Telecommunications Standards Institute has specified guidelines for Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM) structures and dissemination using central, roadside and vehicle ITS stations. CAM is used to send continuous status information of vehicles, whereas DENM is used for asynchronous notification of events.

In light of the above discussion, this paper has the following research contributions

- Propose a holistic solution to provide a real time and predicted traffic information to the users whenever required.
- A security analysis of the proposed solution is carried out to protect against internal and external attacks.
- An implementation of the neural network model with testing on a randomized dataset.

The rest of the paper is organised into the following sections. Section 2 describes some of the existing solutions for sub parts of the problem like data collection, organisation, prediction and message dissemination. Section 3 describes the architecture of the environment in which the problem is being solved and quantifies the problem in terms of specific parameters. Section 4 details the proposed solution and Section 5 outlines the outcome of the simulations carried out and finally, Section 6 concludes the paper.

II. RELATED WORK

We will first examine the solutions to the sub problems of the main task, which consist of traffic estimation, traffic prediction, and disseminating traffic information to relevant users. This will be followed by a description of solutions that aims to solve one or more of the above sub problems.

For the first sub problem of traffic estimation, many solutions (described in later sections) employ the VANET architecture to collect data to estimate traffic flow, like vehicle speed, count, location and density. Authors in [4] outline an interesting blockchain based approach that utilises roadside beacons to detect WiFi signals from vehicles. Each road segment maintains a blockchain of sets of vehicle IDs detected by the beacons, rewarding both the vehicle owner and the beacon maintainer when a block is created, operating a PoW consensus mechanism. In the case of a congestion on a road segment, different portions of the same road segment see different sets of vehicles (which doesn't change much with time), causing forks in the blockchain maintained by the whole road segment. The fork serves both the purposes of detecting a congestion and providing an incentive to vehicle owners to reduce traffic by themselves (as chances of monetary reward reduces with a fork in the chain).

For the second sub problem of traffic prediction, [5] and [6] use Radial Basis Function Neural Networks (RBFNN) to predict traffic flow levels on datasets collected from central authorities. The RBFNN is a three-layered feed-forward neural network. The first and third layer are linear while the middle hidden layer uses radial basis functions as activation functions. The learning in an RBFNN is fast and is generally divided into two phases. The first consists of unsupervised learning between input layer and hidden layer, while the second consists of supervised learning between hidden layer and output layer. [7] uses the VANET architecture to collect vehicle speeds per road segment (or link). Traffic Performance Index (a ratio of velocities), which is a measure of magnitude of congestion, is used to distinguish congested traffic conditions from non-congested traffic conditions. A Deep Neural Network (DNN) model with supervised learning is utilised to estimate link-based traffic flow conditions using real traffic data. [8] also uses vehicle velocities to predict traffic levels. However, it makes use of historical data and in the case of a large enough deviation from historical patterns, uses an Extended Kalman Filter for estimation and prediction. [9] uses an ensemble of classifiers (Fuzzy logic, KNN, ANN-MLP). Weights are

TABLE I: Comparative analysis of data collection methods

Technique	Advantages	Disadvantages
VANET	Enables collection of a large amount of information with very little computation cost	Infrastructure installation, high mobility of network leading to loss of information
Image processing	Ease of installation	Higher computation cost, sensitivity to weather events
Crowd sourcing	Can lead to faster responses, is cheaper	Privacy concern of users, lesser amount of data

assigned to each classifier based on their accuracy rate, which helps alleviate the problem of a classifier being biased towards a particular dataset.

Additionally, authors in [9] describe a Pub/Sub message dissemination architecture. A broker (or server) is present in each road segment where services are registered through a subscription in the Pub/Sub bar event. A store-carry-forward mechanism used so that there is no problem created by temporary loss of connection between vehicles. In the case of a congestion, Dijkstra's algorithm is used to find an alternate route. Shrestha et al. [10] describes a blockchain based event dissemination in VANET which uses a threshold-based mechanism along with a trust level for each vehicle (entity centric trust model) to disseminate honest messages about events in a particular region. A public/private key mechanism is used by the RSUs to provide a Proof of Location (use of GPS is avoided as it can be spoofed) for a vehicle that desires to broadcast information about an event. The European Telecommunications Standard Institute (ETSI) specifies Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM), for continuous status information about surrounding vehicles and asynchronous notification of events, respectively [11]. Yang et al. [12] uses this along with a new Proof of Event consensus to detect and notify users about events. Single hop CAM is used by vehicles to send verifiable vehicle information (like location and speed) to Road Side Units in the same road segment. The Proof of Event consensus mechanism uses two threshold-based algorithms in successive rounds of event verification. When the first threshold is passed by the messages collected via CAM, a notification is sent out using the multi hop DENM. The greater reach enables more information collection, which is checked against the second threshold. If passed and the verdict has not expired (i.e., it's still within the time limit), a final announcement is sent out and the event is stored on the local blockchain. Each block has a time frame within which it is valid. It collects all events occurring within that time frame before it is put permanently on the chain. The local blockchains for each road segment are synced with the global chain periodically.

Hassija et al. [13] combines Ethereum blockchain and neural networks to estimate and predict traffic jams. An artificial neural network (ANN) makes use of the historical data (by accepting timestamp of received data as input) and a Long Short Term Memory (LSTM) network takes into consideration live data (vehicle velocity, condition of road, anticipated slowness, etc) that is crowdsourced by participants on the road. Users that are the first to provide information are rewarded, thus providing an incentive for crowdsourcing. A threshold-based mechanism is implied for checking the correctness of the crowdsourced information, along with a public/private key system for protecting user identities. The outputs from the LSTM and ANN are combined using conflated probabilities and sent to the user that is requesting the traffic information. Since mobile wallets are used for executing transactions, a Proof of Authority consensus is employed to reduce computationally intensive operations. Wang et al. [14] also uses a crowdsourcing mechanism to collect passing time cost of a road segment from users. Unlike the previous solution, it uses a local and global blockchain to reduce network communication overhead with computing nodes owned by individuals or edge routers functioning as miners competing via PoW. The global blockchain stores aggregated reports from local chains and can be queried by any user to receive a traffic report at any location. Mean-Around-Krum scheme [15] is used to aggregate passing time cost and quantity of reports and respectively prevent Byzantine and Sybil attacks. An LSTM, which is continually trained and updated, is used to predict the congestion level.

While the above solutions talk about a decentralised solution mainly to avoid the problems created by a central server (single point of failure, lack of transparency, etc), authors in [16] describe a centralised solution that uses blockchain as a security aid to prevent unauthorised entities from accessing and thus modifying data. Video feed from cameras is analysed by a Jetson Nano to obtain vehicle count and speed using ImageAI and Speed Detector. This data is passed to another intelligent device that uses reinforcement learning to predict the traffic level. The central server is used to re-train the reinforcement learning model and provide updated model parameters periodically. It also controls traffic lights to maximise the flow of vehicles. Users are additionally provided with a web interface to view the traffic conditions and predicted output.

III. SYSTEM MODEL

An established VANET architecture is assumed. Vehicles have OBUs to communicate with other vehicles and RSUs. A central identity management authority is responsible for assigning a pair of public and private keys for each vehicle, and for maintaining the list of valid public keys of requesters who can request for traffic information. This authority can revoke certain identities if they are found to misbehave. The road map is divided into road segments. RSUs of each local segment maintain their own local blockchain which keeps a record of the data received from vehicles about their status.

TABLE II: Summary of related work

Authors	Contribution	Pros	Cons
Haviluddin et al. [6]	RBFNN for traffic prediction	For trained networks, RBF networks perform more robustly and tolerantly than traditional neural networks, when dealing with noised input data set	For function approximation problems, traditional neural networks are preferred, for surfaces without regular peaks and valleys
Yi et al. [7]	DNN for traffic prediction	Uses real time data	Only vehicle velocity used as a parameter to classify traffic, is centralised, only a small fraction of collected data used
Kim et al. [8]	Extended Kalman filter for traffic prediction	Uses historical traffic pattern and real time data making it realistic	Uses velocities of vehicles only in that road segment
Filho et al. [9]	Ensemble of classifiers for traffic prediction, Pub/Sub architecture for message dissemination	Ensemble of classifiers (Fuzzy logic, KNN, ANN-MLP) mitigates bias to a particular dataset	Centralised Pub/Sub architecture for message dissemination
Shrestha et al. [10]	Blockchain for maintaining user trust level and message dissemination	Flaws of GPS overcome by avoiding its use and using a Proof of Location, external attacks reduced through entity centric trust model	Broadcast storm, possibility of internal attacks, computation expensive
Fujihara et al. [4]	Blockchain to incentivise information collection and for congestion detection	Lack of motivation in crowdsourcing is solved	Lots of beacons required, computationally expensive, latency of block creation
Yang et al. [12]	Proof of Event consensus for traffic event validation and trust verification	Lesser time than consensus algorithms like PoW, data centric trust models to reduce internal and external attacks, division into local and global chains reduce load and computation required	Greater time than consensus algorithms like PoS, PoA, pre-decided thresholds to be defined to detect an event (not dynamic)
Hassija et al. [13]	Decentralised traffic prediction using blockchain, LSTM and ANN	Uses historical and real time data, crowdsourcing is incentivised and privacy of users is guaranteed	Relies solely on crowdsourcing for collection of data, internal attacks not prevented
Wang et al. [14]	Blockchain to store traffic information, prediction and mitigation of Byzantine and Sybil attacks via LSTM	Two-layer blockchain to reduce load, crowdsourcing is incentivised and privacy of users is guaranteed, continual learning makes it dynamic, internal and external attacks mitigated	Relies solely on crowdsourcing for collection of data
Tiba et al. [16]	Blockchain-Based Traffic Load Balancing Using Edge Computing and Reinforcement Learning	Blockchain mainly used for security and not storage	Centralised server has single point of failure problem, computation expensive

A global blockchain holds an aggregate of these local reports along with the predicted traffic value. This predicted value is calculated by the global nodes using an LSTM model. Each road segment that maintains a local blockchain has at least one RSU that is a global node. It is also assumed that the global RSU has a list of nearby global RSUs to which it will send a signal if congestion is detected so that the congestion event news can be propagated to only the relevant road segments.

LSTM network is preferred over RNNs to avoid the exploding or vanishing gradient problem. Each LSTM cell has an input gate, forget gate and output gate, which control what information is passed to the next cell. The corresponding standard equations are as follows.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$$\tilde{C}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t)$$

$$h_t = tanh(C_t) * o_t$$

where,

$i_t =$ input gate, $\quad f_t =$ forget gate, $\quad o_t =$ output gate

$w_x =$ respective weights, $\quad b_x =$ respective biases

$x_t =$ current input, $\quad h_{t-1} =$ output of previous block

$C_t =$ cell state, $\quad \tilde{C}_t =$ candidate cell state

The final dense layer uses the following loss function.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$MSE =$ mean squared error

$n =$ number of data points

$Y_i =$ observed values

$\hat{Y}_i =$ predicted values

The square root of the above value, called Root Mean Square Error (RMSE) is to be minimised.

Congestion is quantified based on ranges of average vehicle velocity. Additionally, the Mean Around Krum method [15] can be used to mitigate Byzantine attacks. External attacks are mitigated via the central identity management authority.

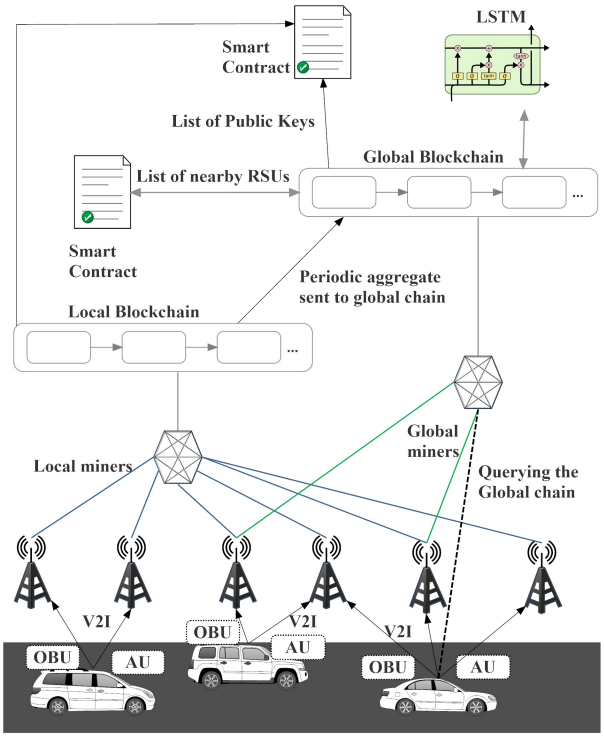

Fig. 2: The proposed system model

IV. PROPOSED WORK

This section describes the individual components of the proposed solution and how each component connects to the others, explaining the entire work flow.

A. LSTM Model

A sequence to vector LSTM model is used. One data point in a sequence consists of the average velocities in each local road segment. This means that the input sequence (with n time steps and say, 3 road segments) is of the form $[[v_1(t), v_2(t), v_3(t)], [v_1(t+1), v_2(t+1), v_3(t+1)], ..., [v_1(t+n-1), v_2(t+n-1), v_3(t+n-1)]]$, were $v_i(t)$ is the average velocity of ith road segment in time slot t. The output is a vector for this input sequence would be of the form $[v_1(t+n), v_2(t+n), v_3(t+n)]$, where these velocities are the predicted average velocities for time slot $t+n$.

B. Key management

An external authority manages distribution of private and public keys. Valid public keys are stored on the global blockchain, with an indicator if they are authentic or have been revoked. Only the admin who deploys this smart contract can modify this list. This smart contract is also accessible by the

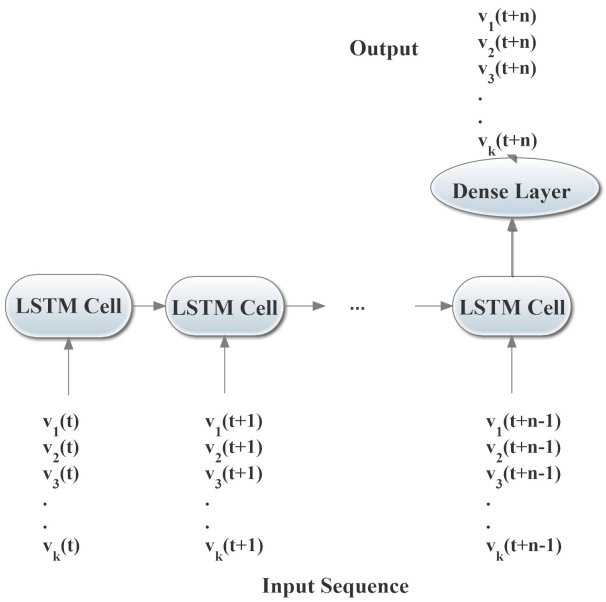

Fig. 3: LSTM model

local chain when checking validity of velocity reports received by it. The global nodes on receiving a local block also verify all identities as a safety measure.

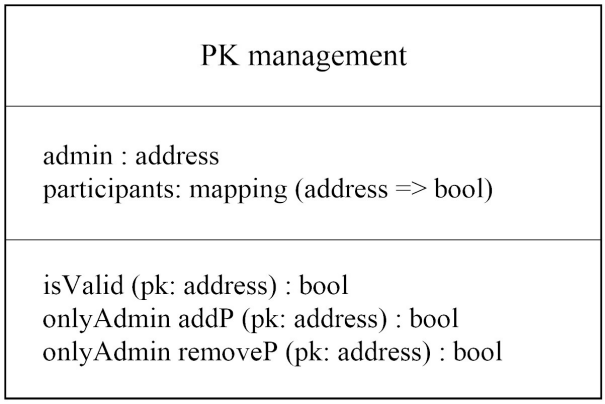

Fig. 4: Structure of PK.sol

C. Neighbour Address

One RSU has the address of neighbouring RSUs. If it detects congestion in its segment, it tells the neighbouring RSUs to send a notification to their own local segments. Only a central authority is allowed to add/delete addresses from the list. This smart contract is accessible only by the global nodes.

D. Blockchain Structure

Each road segment maintains its own local chain which records reported vehicle velocities in a timeslot. The algorithm is described below.

(**Algorithm 1**). On receiving a velocity report, the RSU checks the validity of the address by invoking the utility of the smart contract, PK Management. If the data is from a

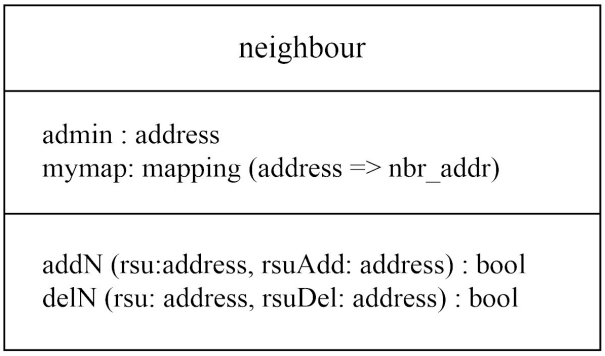

Fig. 5: Structure of nbr.sol

Algorithm 1: Local block

Individual velocity reports of vehicles in segment S_i Local block for segment i, and time period condensed into one value of timeslot t
Function `Receive(`*data*`)`**:**
  if *isValid(data.address)* **then**
    Add $data$ to IPFS
    Send hash of data to other RSUs, adding it to pool of $data_i$
  else
    Discard $data$
end
On formation of local $block_i(t)$ via PoW, broadcast block to global nodes

registered user, it is added to the IPFS and the hash of the file is added to the pool of data circulated among all the RSUs of that segment. If the data is invalid, it is discarded. PoW consensus leads to formation of a local block which consists of the hashes of the IPFS files of the user velocity reports.

Once a block is made, it is broadcast to the global chain, which calculates the average velocity for that period of time, generating one data point. Due to the sheer amount of information and the inefficiency of blockchain to store vast amounts of data, the information can be stored on IPFS, thus requiring only the hashes of the files containing the information to be stored on the chain.

The global blockchain thus stores the aggregated average velocities and number of reports obtained for each local segment. Additionally, the predicted values (obtained using LSTM) of average velocities and number of reports for each local segment in the next time slot is also stored. This value can be queried by any user to obtain not only the current traffic report but the predicted one as well. The algorithm for the generation of a global block is described below.

(**Algorithm 2**). On receiving a local block from any segment, a global RSU proceeds to extract the IPFS hashes of the files containing the velocity reports. Validity of the user is checked for each file. If the report is invalid, the entire block is discarded. If all the reports are valid, the calculated average velocity and total number of reports are stored on

Algorithm 2: Global block

Local blocks of segment i, timeslot t $b_i(t)$ Global block
Function `Receive(`$b_i(t)$`)`**:**
  foreach *IPFS hash* $\in b_i(t)$ **do**
    Get $data$
    if *isValid(data.address)* **then**
      $sum_i(t) \leftarrow sum_i(t) + data.velocity$
      $numReports_i(t) \leftarrow numReports_i(t) + 1$
    else
      Reject $b_i(t)$
  $v_{avg(i)}(t) \leftarrow sum_i(t)/numReports_i(t)$
  Add new data aggregates to IPFS
  Send hash of data to other global nodes, adding it to pool of global data
  Accept $b_i(t)$
end
`/* Once all` $b_i(t)$ `for every` i `have been accepted, proceed */`
$\forall i$, feed $v_{avg(i)}(t-n+1)$,..., $v_{avg(i)}(t)$, to LSTM where n = number of timesteps
Obtain predicted $v_{avg(i)}(t+1)$ for each i
Store in IPFS
Broadcast hash to other global nodes, adding it to pool of global data
Form global block via PoW

the IPFS and the hash is added to the pool of data circulating amongst the global nodes. After local blocks from all segments for a particular time slot have been received and verified, the aggregated average velocities are fed into the LSTM model along with data from previous time slots and the predicted average velocities for each local segment is obtained. This data is also stored on IPFS and its corresponding hash is circulated amongst all global nodes. The global block consisting of IPFS file hashes of the aggregated velocity reports and the predicted velocities is eventually formed via PoW.

PoW consensus is used and a large latency in block production is avoided by the structure of local and global chains. If the predicted average velocity is below a certain pre-decided threshold, the global node belonging to that local segment queries the smart contract to get the addresses of its neighbouring nodes, to which it sends a request to alert the vehicles in their respective segments of a congestion. The predicted number of reports can be used to detect potential spurious attacks. If such an attack is detected, the public key of the vehicle can be revoked by the central identity management authority.

V. SIMULATION AND RESULTS

This section analyses the security aspects of the proposed solution and also to experimentally obtain the most suitable values for the LSTM model parameters using a randomly generated dataset.

A. Dataset

A sample 5 road segments have been considered. A local block, spanning a time period has 5 data points (consisting of reported vehicle velocities) each. There are 5000 such blocks, implying 25000 data points for each road segment. These points have been generated randomly.

B. The Proposed Model

There is a single hidden LSTM layer followed by a standard feed-forward output layer. The LSTM layer has 200 units, followed by a dense layer with 5 outputs (representing the predicted average velocities for the 5 road segments). A 65 to 35 train to test split was used, with a time step of 50. The value of time step and number of units in the LSTM layer and their corresponding total RMSE (Root Mean Square Error) values is depicted in the following graph, which explains why eventually, a time step of 50 with 200 as number of units was used. The units of RMSE is the same as that of velocity.

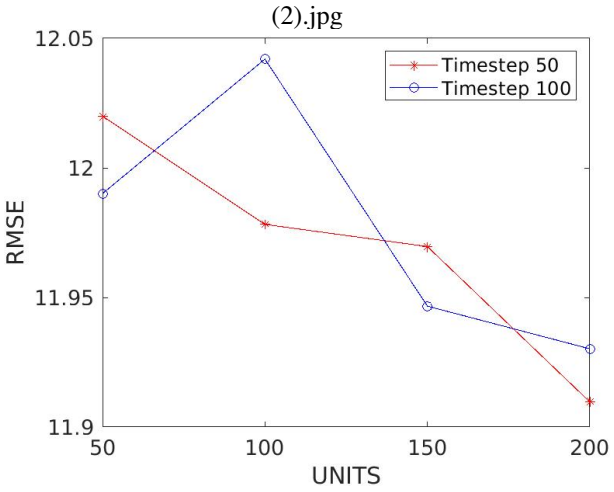

Fig. 6: RMSE with varying values of time step and LSTM units

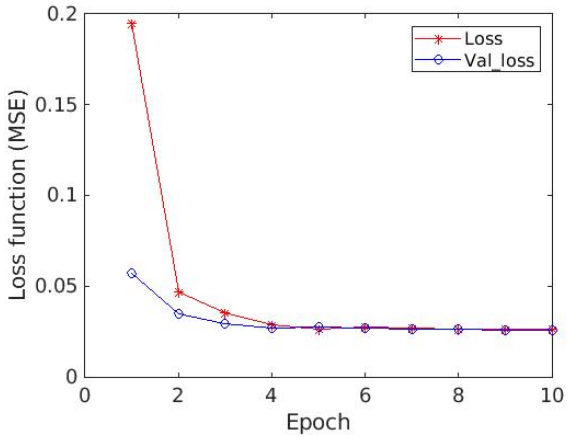

Fig. 7: Loss function when training the model (Time step: 50 and Units: 200)

C. Security Analysis

There are mainly two kinds of attacks that can occur in a system – internal and external attacks. External attacks are those carried out by outsiders (no prior access to the system) with malicious intent while internal attacks are carried out by authorised users. Attackers may attempt to corrupt recorded data, provide false data, overwhelm or disrupt the system by making a large number of continuous queries, and so on.

External attacks are prevented in the proposed system by the requirement of an external authority that registers users that will provide the data and have the rights to query the system for traffic reports. The public key of the verified user is recorded in the smart contract and the validity of the user can be revoked by the same central authority. Before the data is recorded on the chain (in the form of hash of the IPFS file that has the real data), it is checked if it has come from a valid user using the concept of digital signatures. Information that is signed using user's private key is verified using user's public key. Validity of this public key is checked using a smart contract (PK management). Thus, false data entry by external attackers is prevented. Internal attacks can be mitigated by employment of the Mean Around Krum method as described in [14] and [15].

The Proof Of Work (POW) consensus algorithm makes the chain of recorded information tamper resistant and thus safe from internal and external attacks to corrupt previously stored data. It involves the nodes or miners (in our case, RSUs) to solve a mathematical puzzle of finding a nonce such that the hash of the block including nonce is a subset of a range of values. By narrowing or expanding this range of values, the difficulty of this computation and thus time taken to solve it can be adjusted.

The hash function can be represented as a function

$$H : X \rightarrow Y \; with \; H(x) = y \; where \; x \in X, \; y \in Y$$

. The following properties account for the strength of the POW consensus mechanism (which, however, is still susceptible to the 51% attack):

- It is infeasible to find two inputs x_1 and x_2 such that $H(x_1) = H(x_2)$ (Collision resistant)
- Given an output y, it is infeasible to find the corresponding input x_1 such that $H(x_1) = y$ (Preimage resistance)
- Given x_1 and y such that $H(x_1) = y$, it is infeasible to find another input x_2 such that $H(x_2) = y$ (Second preimage resistance)

By including the hash of the previous block in the current block, an attempt to tamper data of a previous block involves redoing the Proof Of Work for all subsequent blocks. This requires a significant proportion of computational power (> 51%) in the hands of the attacker.

VI. CONCLUSION

Increasing vehicular congestion on roads poses a lot of risks and problems associated with degradation of environmental resources and negative impacts on human health. This paper provides a holistic solution that targets the problem of traffic information management and traffic level prediction. VANET

TABLE III: Summary of certain attacks with techniques to combat them

Common attacks	**Prevention tactic**	**Advantages**	**Disadvantages**
External/Internal attempt to tamper data	Proof of Work consensus mechanism	Completely decentralised	High power consumption
External attempt to provide false data	Central authority for identity verification, Smart contract to manage public keys	Reduced costs and transparency to some extent	Requires trust in a third party
Internal attempt to provide false data	Mean around Krum method	Byzantine resilient	Reports that it can be broken using inner product manipulation [17]

architecture is utilised for information collection. IPFS and a two-level blockchain are leveraged for the purpose of information storage and security, along with a central identity registration authority. LSTM networks are utilised for traffic level prediction. Traffic information can be queried by users at anytime and in the case a congestion is detected, users in the vicinity are alerted, thus dealing with the problem of accessibility of traffic level information. An analysis of the proposed solution from a security viewpoint shows that it is secure from attacks that aim to corrupt data. This work can be extended by testing it further in a real time environment, enabling continuous learning.

DECLARATIONS

Funding: No funding available for this manuscript.
Conflicts of interest/Competing interests: The authors declare that they have no conflict of interest statement.
Availability of data and material: Not Applicable.
Code availability: Not Applicable

REFERENCES

[1] Chin, Vincent & Jaafar, Mariam & Subudhi, Suresh & Shelomentsev, Nikita & Do, Duang & Prawiradinata, Irfan. (2018). Unlocking Cities - The impact of ridesharing across India. https://image-src.bcg.com/BCG-Unlocking-Cities-Ridesharing-India_tcm21-185213.pdf

[2] TomTom Traffic Index, accessed October 13, 2020. https://www.tomtom.com/en_gb/traffic-index/

[3] Dan C. Marinescu, in Cloud Computing (Second Edition), 2018.

[4] Fujihara, Akihiro. (2019). Proposing a System for Collaborative Traffic Information Gathering and Sharing Incentivized by Blockchain Technology: The 10th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2018). 10.1007/978-3-319-98557-2_16.

[5] Jun, Ma & Ying, Meng. (2008). Research of Traffic Flow Forecasting Based on Neural Network. 2.10.1109/IITA.2008.207.

[6] Haviluddin, Haviluddin & Tahyudin, Imam. (2015). Time Series Prediction Using Radial Basis Function Neural Network. International Journal of Electrical and Computer Engineering (IJECE). 5. 31-37. 10.11591/ijece.v5i4.pp765-771.

[7] Yi, Hongsuk & Jung, Heejin & Bae, Sanghoon. (2017). Deep Neural Networks for traffic flow prediction. 328-331. 10.1109/BIGCOMP.2017.7881687.

[8] Kim, Sung-Soo & Kang, Yong Bin. (2007). Congestion Avoidance Algorithm Using Extended Kalman Filter. Convergence Information Technology, International Conference on. 913-918. 10.1109/ICCIT.2007.147.

[9] Filho, Geraldo & Meneguette, Rodolfo & Neto, Jos & Valejo, Alan & Weigang, Li & Ueyama, Jó & Pessin, Gustavo & Villas, Leandro. (2020). Enhancing intelligence in traffic management systems to aid in vehicle traffic congestion problems in smart cities. Ad Hoc Networks. 102265. 10.1016/j.adhoc.2020.102265.

[10] Shrestha, Rakesh & Bajracharya, Rojeena & Nam, Seung Yeob. (2018). Blockchain-based Message Dissemination in VANET. 161-166. 10.1109/CCCS.2018.8586828.

[11] Santa, José & Pereniguez-Garcia, Fernando & Moragón, Antonio & Skarmeta, Antonio. (2014). Experimental evaluation of CAM and DENM messaging services in vehicular communications. Transportation Research Part C: Emerging Technologies. 46. 98–120. 10.1016/j.trc.2014.05.006.

[12] Yang, Yao-Tsung & Chou, Li-Der & Tseng, Chia-Wei & Tseng, Fan-Hsun & Liu, Chien-Chang. (2019). Blockchain-Based Traffic Event Validation and Trust Verification for VANETs. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2903202.

[13] Hassija, Vikas & Gupta, Vatsal & Garg, Sahil & Chamola, Vinay. (2020). Traffic Jam Probability Estimation Based on Blockchain and Deep Neural Networks. IEEE Transactions on Intelligent Transportation Systems. pp. 1-10. 10.1109/TITS.2020.2988040.

[14] Wang, Qianlong & Ji, Tianxi & Guo, Yifan & Yu, Lixing & Chen, Xuhui & Li, Pan. (2020). TrafficChain: A Blockchain based Secure and Privacy-Preserving Traffic Map. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2980298.

[15] Blanchard, Peva & El Mhamdi, El Mahdi & Guerraoui, Rachid & Stainer, Julien. (2017). Byzantine-Tolerant Machine Learning.

[16] Tiba, Kevin & Parizi, Reza & Zhang, Qi & Dehghantanha, Ali & Karimipour, Hadis & Choo, Kim-Kwang Raymond. (2020). Secure Blockchain-Based Traffic Load Balancing Using Edge Computing and Reinforcement Learning. 10.1007/978-3-030-38181-3_6.

[17] Cong Xie, Sanmi Koyejo, & Indranil Gupta. (2019). Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation.

Figures

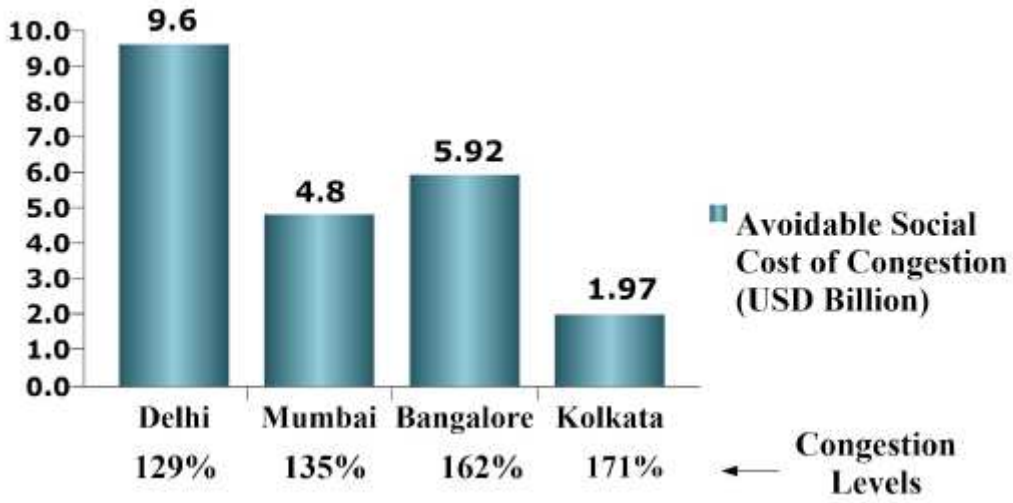

Figure 1

Avoidable social cost of congestion (USD Billion) for 4 cities of India

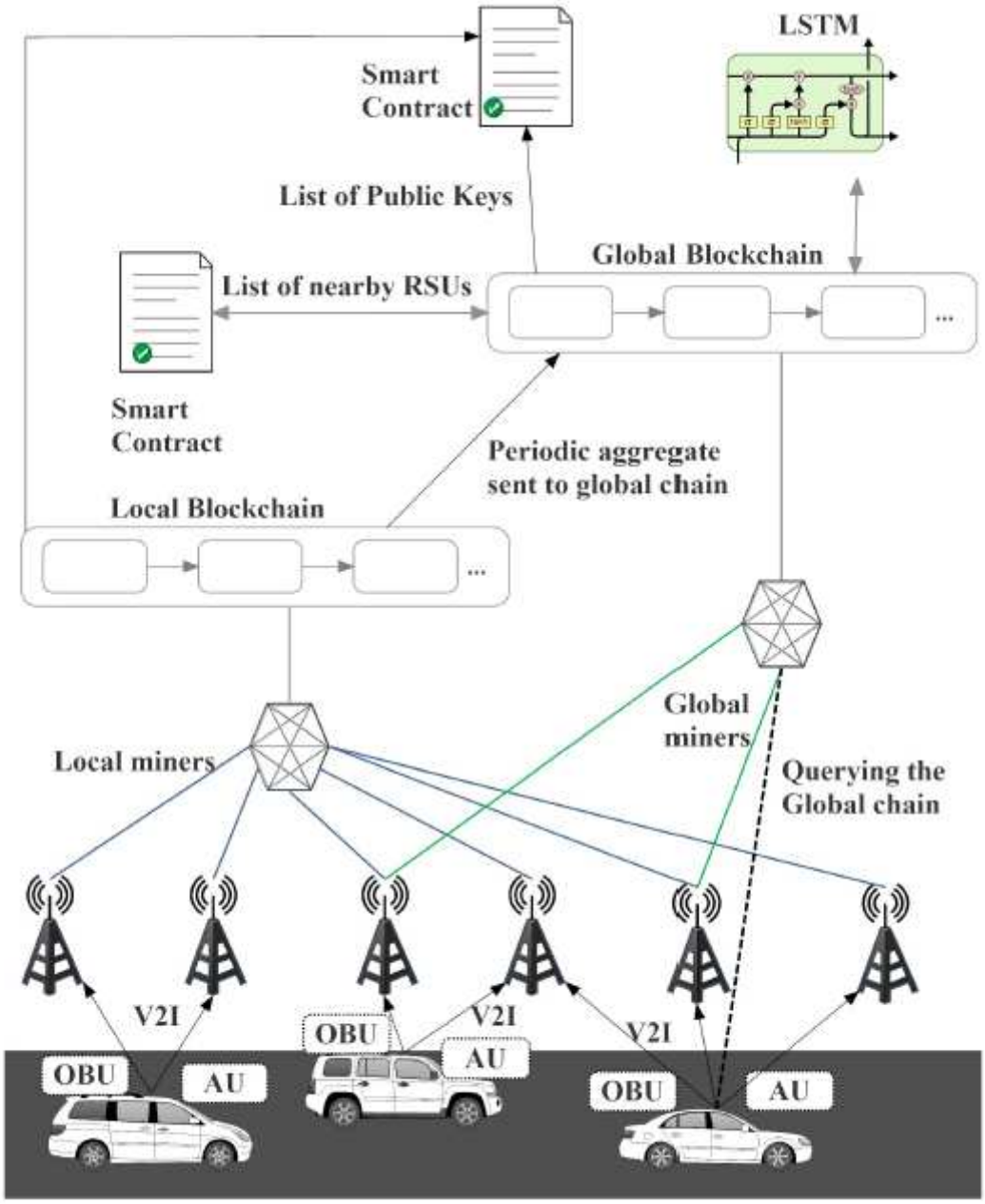

Figure 2

The proposed system model

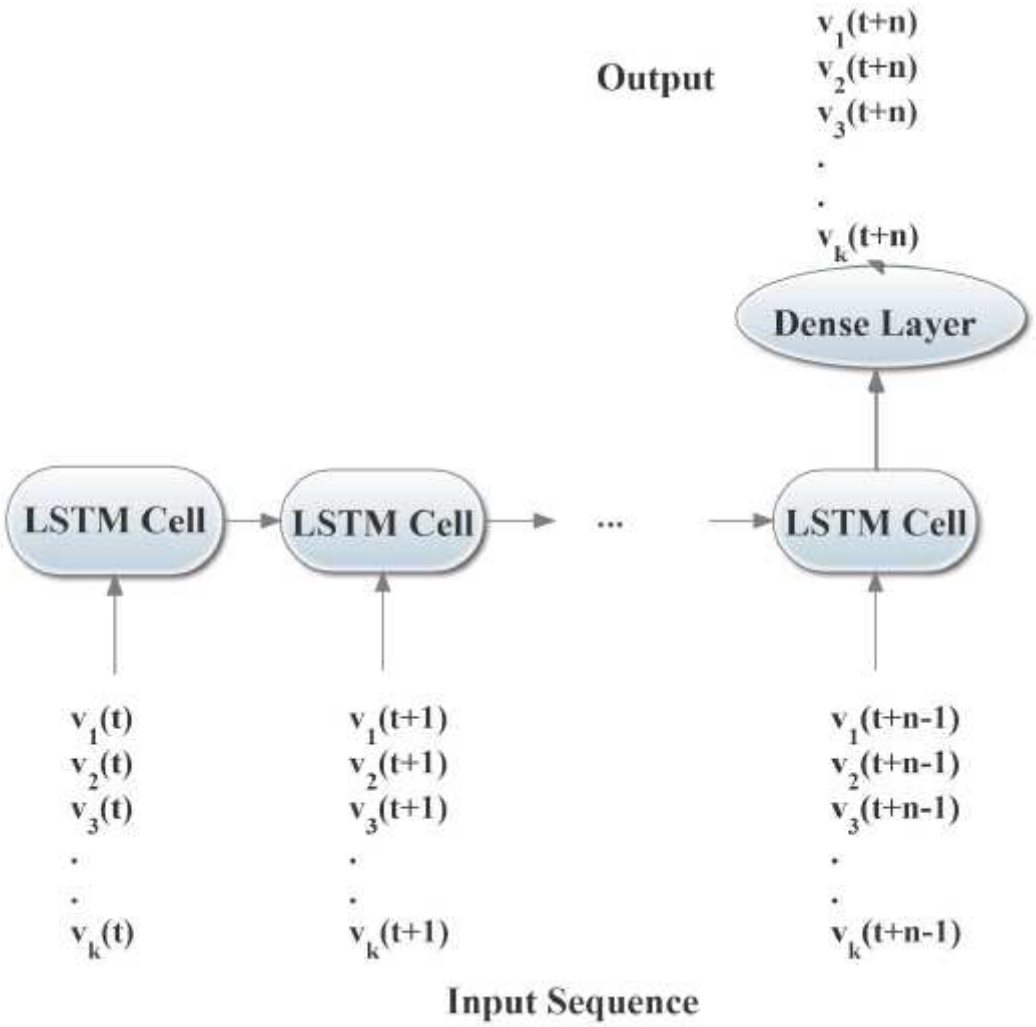

Figure 3

LSTM model

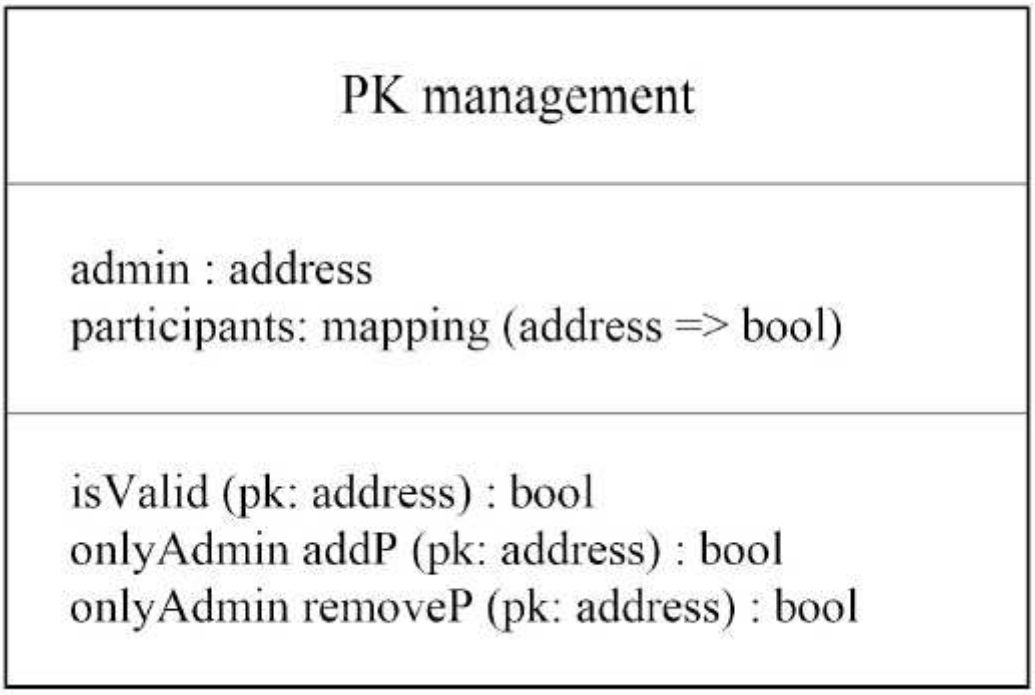

Figure 4

Structure of PK.sol

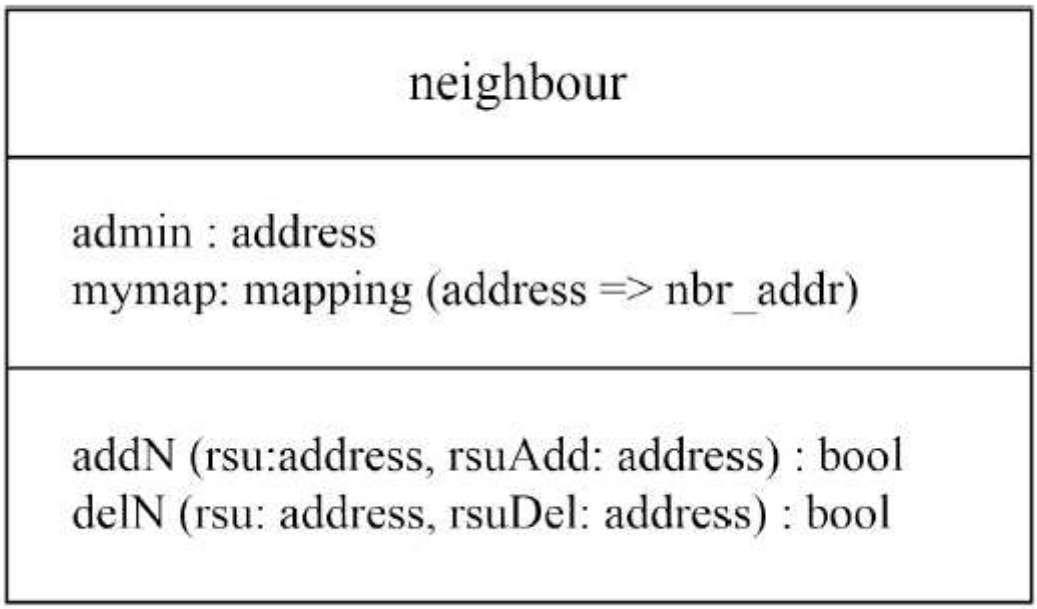

Figure 5

Structure of nbr.sol

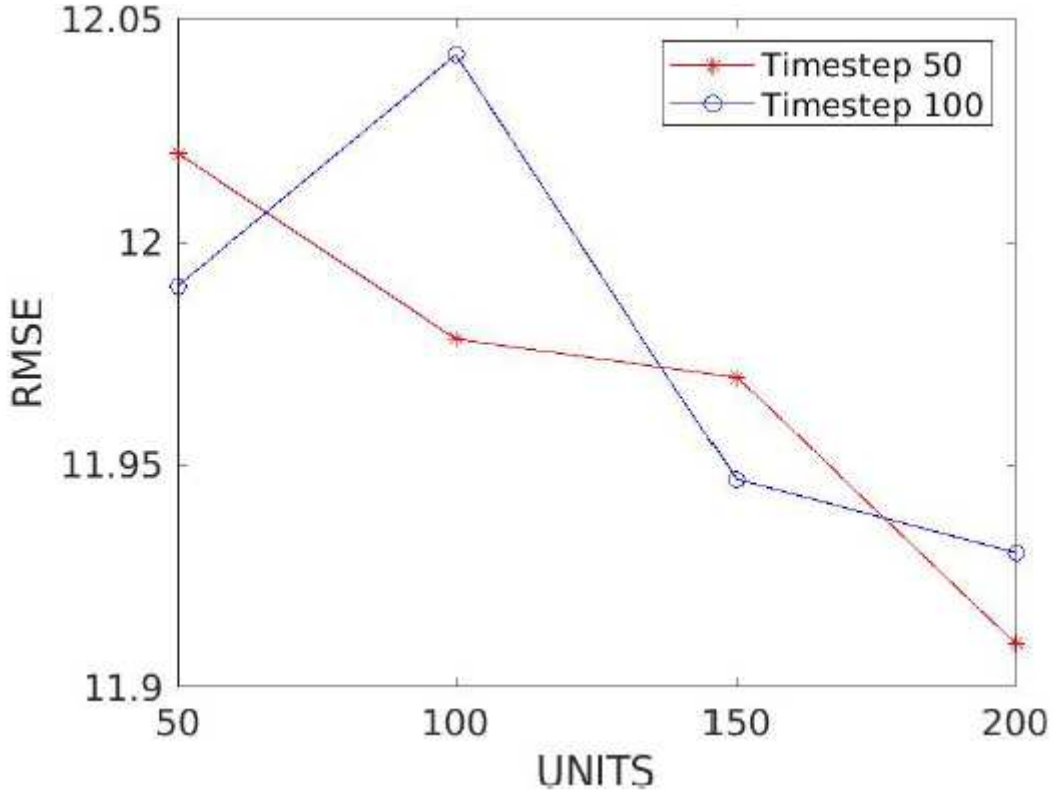

Figure 6

RMSE with varying values of time step and LSTM units

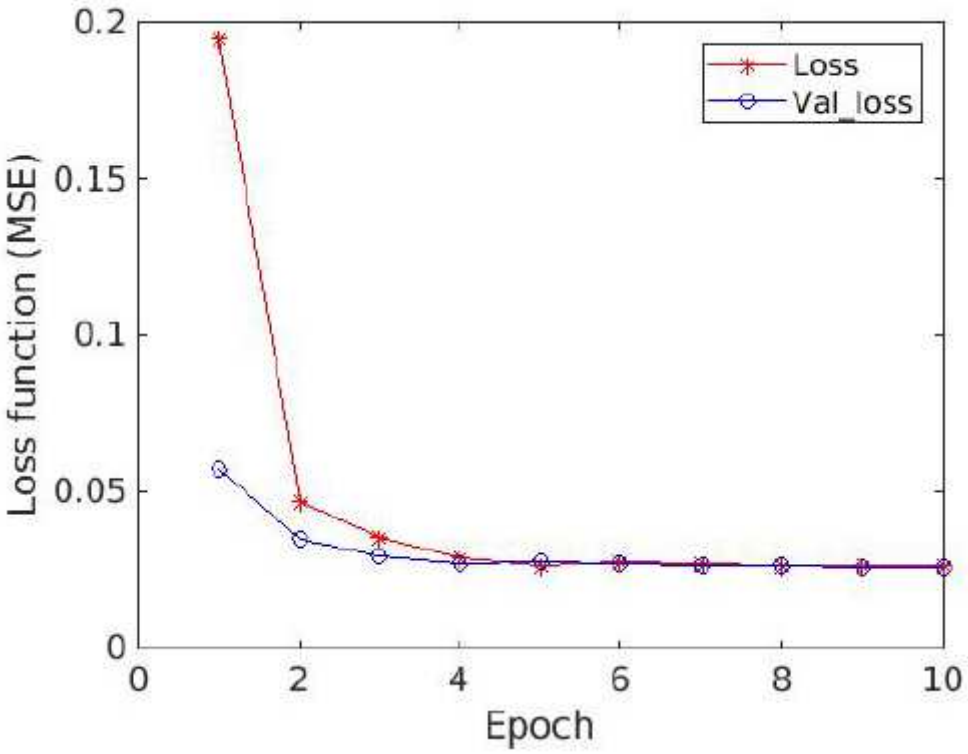

Figure 7

Loss function when training the model (Time step: 50 and Units: 200)