ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

Empirical Formulation of Highway Traffic Flow Prediction Objective Function Based on Network Topology

Arsalan Rahi[1], Soodamani Ramalingam[2]

[1,2*] Doctoral Student, [3] Senior Lecturer
[1,3] Centre for Engineering Research, School of Engineering and Technology, University of Hertfordshire (UH), Hatfield, Hertfordshire, United Kingdom, AL10 9AB.
[2] Research Assistant at University Bus Limited (UNO), Hatfield, Hertfordshire, United Kingdom, AL10 9BS.
* Corresponding authors: arsalanrahi92@gmail.com

ABSTRACT: Accurate Highway road predictions are necessary for timely decision making by the transport authorities. In this paper, we propose a traffic flow objective function for a highway road prediction model. The bi-directional flow function of individual roads is reported considering the net inflows and outflows by a topological breakdown of the highway network. Further, we optimise and compare the proposed objective function for constraints involved using stacked long short-term memory (LSTM) based recurrent neural network machine learning model considering different loss functions and training optimisation strategies. Finally, we report the best fitting machine learning model parameters for the proposed flow objective function for better prediction accuracy.

KEY WORDS: Intelligent Transportation Systems, Machine Learning, LSTM, Flow Estimation, Hyper Parameter Optimisation.

I.INTRODUCTION

With the understanding of how intelligent transport systems (ITS) operate in a modern city, their reliance on an accurately predicted regional traffic flow and congestions changes have become inevitable. This gives rise to the quest for finding the better formula to forecast traffic parameters for as close as possible to the real world observed parameters [1]. But for ITS and transport operators to rely on traffic parametric forecasts, systems must be reliable, and this is only possible when the forecasting systems represent the traffic network on a smallest unit as offered by the network which consists of junction and the inter road links. Based on this criterion we set out the flow of this paper. We report the unique significance of the proposed system in section II, section III sheds a detailed light on what has already been done in the relevant subject in response to the advancements in machine learning technique and traffic flow predictions. Section IV list the proposed strategy along with the subsequent subsections detailing the dataset and pre-processing involved along with the system design and performance metrics are considered. Sections V and VI deal with the experimental results and their conclusion with future suggestions respectively.

II. SIGNIFICANCE OF THE SYSTEM

The paper mainly focuses on predicting the real traffic flow based on retaining the traffic network topology in the form of a dynamic objective function and using data driven time series spatiotemporal machine learning model to optimise it for more accurate highway network individual road flow predictions.

III. LITERATURE SURVEY

Traffic flow forecasting has been in research discussions for quite some time. Traffic flow forecasting can be broadly classified into two distinct categories which are as follows:
Parametric: Conventional approaches that use statistical methods for time series forecasting are normally termed as parametric model approaches. The prior knowledge of data distribution is assumed in parametric approaches. Most notable of these approaches are auto regressive integrated moving average (ARIMA) and its variant seasonal auto regressive integrated moving average (SARIMA) [2], Kalman filters [3] and exponential smoothing [4]. The problem

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

with most of these parametric approaches is that they can effectively be employed for only one-time interval prediction and cannot predict well enough due to the stochastic and nonlinear nature of the traffic data. This can better suit short term forecasts only which are well biased towards the most recent observations in the data, thus this makes the parametric approaches incapable of handling real world trends.

Non-Parametric: A few years ago machine learning (ML) strategy based traffic parameter prediction algorithms have been utilised [5]. These data driven approaches are also termed as non-parametric approaches. The most commonly tested non-parametric approaches for spatiotemporal traffic forecasting includes the k-nearest neighbours (KNN) [6][7] and support vector regression (SVR)[4]. However, these shallow ML algorithms work in a supervised manner which makes their performance dependent upon the dataset manual feature selection criteria.

With the advancement in the ML algorithms, a bit more sophisticated dense supervised learning approach is applied for traffic predictions by using back propagation techniques in artificially connected neural networks (ANN) [8][9]. Although ANN out performs conventional linear parametric models but struggles with simple time series data learning and finding global minimum. Recently, deep recurrent neural networks (RNN) have shown some great promises for dynamic sequential modelling especially in the field of speech recognition [10][11]. Simple RNNs however suffer from gradient explosion for extra-long sequence training which results in information loss and reduced performance [12]. *Fu R et al* [13], have used the RNN variants called long short term memory (LSTM)[14] and gated recurrent units (GRU) for the traffic forecasting because of their ability to retain and pass on the information that is necessary and forget what is redundant using the output and forget gates. *Haiyang Yu et al.* proposed the spatiotemporal traffic feature learning utilising the deep convolutional LSTM network where LSTM network learns the temporal dependent patterns in the data. This makes the LSTM vanishing gradient problem during back propagation problem to fade off during error training with the usage of LSTM memory blocks and makes it able to predict with much accuracy for longer sequences [15]. For the very reason we employ LSTM in our proposed methodology to learn the temporal features whereas to keep the training and the model architecture simple we incorporate the feed forward connected ANN layer at the end for the spatial feature learning and then we train the whole architecture in a back-propagation manner. This is further discussed in the system design section.

IV. METHODOLOGY

In this Section, we represent a traffic model as consisting of a set of nodes and input-output links. The traffic flow of a set of input links will have an influence on the traffic flow of the output links. This model acts as a black box interpreting and manipulating the system inputs. A system is governed by a set of rules associated with a combination of the inputs fixed and dynamic states mapped to outputs and represented in mathematical terms [16]. Such a system can be modelled as an objective function consisting of variable parameters is shown in figure 1.

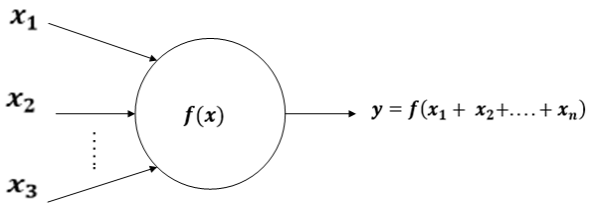


Figure 1. A general function definition.

A) Definitions

We consider a highway junction spatially with inflows and outflows to be an independent system and designate each junction system as a node denoted by N. The links L serves as both the inputs as well as outputs of a node in bidirectional highway links. As an example, consider a single sample node of an actual highway junction in Hertfordshire, UK, shown in figure 2.a and its equivalent representation using the nodes and links configuration is given in figure 2.b. Further, the bidirectional arrows indicate bidirectional traffic flow of the node. Here outflow implies traffic flow moving away from the node and inflows to those moving into the node.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

a) b)

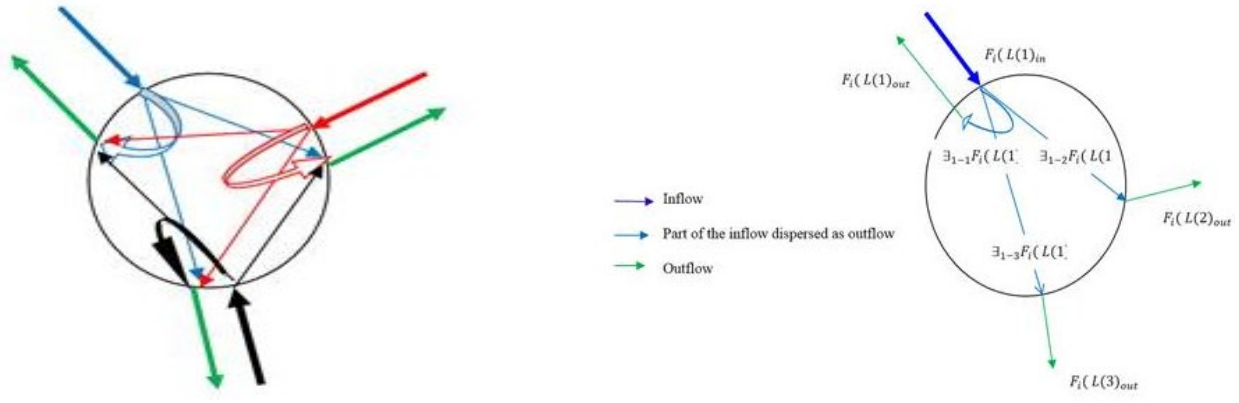
Figure 2. a) Highway junction under consideration (Google Maps, 2018). b) Node illustration retaining junction original topology.

B) Flow Estimation Function

To predict the outflow of traffic for each individual link on a single node, all the incoming link flows are to be considered for the output flow forecast objective function. The outflow of a node's link is determined by the summation of inflows of individual links of the node. Figure 2.b shows that the output flow associated with a link is dependent on the inflows of every other link in the same node. The estimated traffic outflow for link $L1$ is given by equation (1) showing the dependency of the objective function on the inflows associated with the rest of the links of the same node. Equation (2) is a more general objective function mathematical representation which describes the conservation of flow with a node system where x is the link for which the flow is being calculated and n is the total number of links on the same N. This makes the objective function retain the correlations in the flow characteristics for each individual node link when the single node is considered as a basic unit level in the traffic network.

$$L1_{out} = f(L2_{in} + L3_{in} + L1_{in}) \quad \{ L1, L2, L3\ \varepsilon\ (same\ N) \} \tag{1}$$

$$L(x)_{out} = f(L(n-x)_{in}) \quad \begin{cases} x, n\ \varepsilon\ (same\ N) \\ and\ x < n \end{cases} \tag{2}$$



a) b)

Figure 3. a) Extension of traffic network at node *i* showing three links and their associated inflows and outflows. b) A simple traffic network at a node *i* with 3 links. It shows the distribution of incoming traffic dispersed as outgoing traffic at the node.

With reference to figure 2.b, let us consider a node *i* consisting of a set of links $L(j)$ that are associated with bi-directional traffic flow F_i. Each link $L(j)$ at the node *i* is associated with traffic inflow $F_i(L(j)_{in})$ and a corresponding outflow

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

indicated by $F_i(L(j)_{out})$. The function for the traffic flow of links $L(j)$ considers the fact that the traffic inflow of every link contributes partially (to a certain degree) to the outflow of each of the other links at the same node. In other words, the traffic outflow of a link is a function of the traffic inflow of all the other links including its own at the node. This notion is modelled as follows:

$$F_i(L(j)_{out}) = \sum_{j=1}^{n} {F'(L(j)_{in})}/{F_i(L(j)_{in})} \quad (3)$$

where ${F'(L(j)_{in})}/{F_i(L(j)_{in})}$ represents a fraction of the traffic inflow that contributes to an outflow of a specific link. As an example, consider figure 3.b, in which the circle represents a node *i* with three links. The thick blue arrow indicates the traffic inflow of link $L(1)_{in}$ that gets dispersed into the node and flows through the rest of the links. They contribute to the outflows of the rest of links including itself. This dispersion is indicated by thin blue arrows in Fig 3.b. The outflow of each of the links in shown in green arrows. The symbol $\exists_{1-j}$ indicates that part of the inflow of link $F(L(1)_{in})$ contributes to the outflow of the links $L(j)_{out}$. The sum of the traffic flow of $F(L(1)_{in})$ inside the node represented by thin blue arrows is equal to the traffic inflow of $L(1)$ represented by a thick blue arrow, at a time instant. This applies to the traffic inflow of all other links at the node as shown in figure 3.b.

We will show in the sub-section E, the use of the above model in the proposed system design.

C) Dataset Description

We perform all the experiments on the traffic flow dataset for the chosen Hatfield Hertfordshire UK area junction as shown in figure 1. The dataset is obtained from Gov.uk open datasets which contains public sector information licensed under the Open Government Licence v3.0 [17]. The used dataset contains traffic flow information for two-hour timed aggregated intervals from start of 1st April 2015 to the end of 31st Dec 2015 for the highway roads. First three and last three raw dataset plots for links are shown in figure 4. The data is collected for the number of passing vehicles using the loop detectors installed on both the ends of the selected highway links.

D) Dataset Pre-processing

Figure 4. First and last three days of pre-processed data.

The raw dataset is taken through a series of data preprocessing steps:

Data Cleaning: As with every real world gathered data the links flow raw dataset had approximately 15% of values that were missing. Due to the ongoing trends comprising of seasonality and other environmental factors it is very important to retain the inherit trends in the traffic data. So, these values are imputed using the backward fill approach. The backward filling approach takes the value from next interval logged value and make an imputation for the previous interval. This imputing process continues until all the missing imputes are done through which all the inconsistencies are resolved.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

- **Data Integration:** A total of 3252 data samples are used for each considered link. Using equation (1) they are reshaped to form an array of dimensions 3252x4. Where 4 corresponds to the links considered as given by equation (1). The sample plot from dataset containing the newly shaped $L1_{in}$ and two outflows i.e.$(F(L(1)_{out}, L(2)_{out}))$ for first and last three days of the gathered dataset are shown with twelve two-hour intervals as shown in figure 4.
- **Data Transformation:** After the data aggregation and reshaping is done it is further generalized and normalized by scaling for the minimum and maximum values among each data column. i.e. intra flow links normalization. Further the reshaped dataset is lagged by one-time interval to make it suitable for supervised training.
- **Data Reduction:** With the aim to generate the training and validation sets to train and validate the ML model we consider 20% of the original dataset as the validation set. Since it's a time series consecutive interval data the order of training and validation ensemble is very important. Therefore, we consider the tail end 20% for the validation of trained model after each training iteration.
- **Data Discretization:** Among the originally reported dataset there are twelve intervals in a twenty-four-hour time window we consider only the twelve intervals which are two hours apart each to make the ML model training not only fast but a more generalized representation of the sequential data throughout the day

E) System Design

In this section the machine learning model used to fit the pre-processed data is discussed. We discuss the architecture of LSTM and the proposed architecture based on the combination of LSTM and the NN architectures.

- **Feed Forward-Long Short-Term Memory (LSTM):** As the first part we just consider the recurrent neural network (RNN) variants called long short-term memory (LSTM) units in training for feed forward data iteration as the main time series data learners of our ML architecture along with conventional connected feed forward neural networks (NN). The hybrid LSTM-NN architecture is shown in figure 6. This part of the architecture consists of two layers of LSTM units and one layer of densely connected NN. In between each layer is an activation function. The LSTM model is defined [12] by the following set of equations:

$$f_t = \sigma(w_f.[h_{t-1}, x_t] + b_f) \quad (4), \qquad i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i) \quad (5),$$

$$\bar{C}_t = \tanh(w_c.[h_{t-1}, x_t] + b_c) \quad (6), \qquad C_t = f_t \otimes C_{t-1} + i_t \otimes \bar{C}_t \quad (7),$$

$$o_t = \sigma(w_0[h_{t-1}, x_t] + b_0) \quad (8), \qquad h_t = o_t \otimes \tanh(C_t) \quad (9).$$

LSTM's general purpose can be defined as the estimation of the conditional probability $p\left(y_1, y_2, \ldots y_{T'} \mid x_1, x_2, \ldots x_T\right)$ given that $(x_1, x_2, \ldots x_T)$ is an input sequence and $(y_1, y_2, \ldots y_{T'})$ is the corresponding output sequence. The lengths of T' and T may differ. The deep LSTM computes the conditional probability by first computing the fixed dimensional input representations v, of the input sequence, from the last hidden memory state of the LSTM layer [18]. The hidden states h_t for each individual LSTM unit is calculated as given by the equation (9). Accordingly, for the proposed objective function in

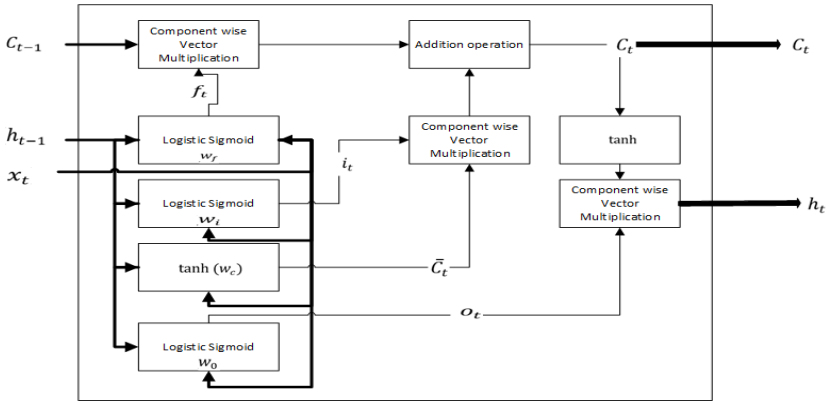


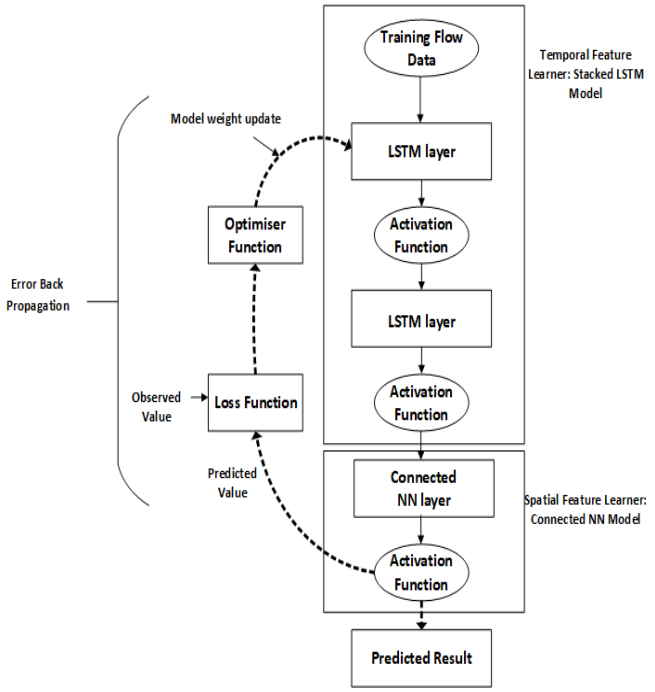
Figure 5. Structural data flow in a Long Short-Term Memory (LSTM) unit [11].

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

(3), standard LSTM network for the i^{th} node with internal hidden states v of corresponding inputs $\left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)$ is given by equation (10) :

$$\left(F_i(L(k)_{out})_1,\ F_i(L(k)_{out})_2, \ldots,\ F_i(L(k)_{out})_{T'} \mid \left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)\right)$$

$$= \prod_{t=1}^{T'} p\ (F_i(L(k)_{out})_t \mid v,\ F_i(L(k)_{out})_{t-1}) \qquad (10)$$



where k in the $F_i(L(k)_{out})$ represents the output, link being considered by the LSTM for the output flow conditional probability estimations. As shown in the system design (refer figure 6.) the LSTM layers are cascaded with NN layers. Equation (10) can now be interpreted for our flow problem as given by equation (11) which forms the model for traffic flow. Note that in equation (13), f_o and f_h represent the output and hidden layer activation functions respectively. H_j in equation (12) and O_k in equation (13) define the hidden layer and output layer outputs.

It is to be noted that there are two LSTM layers stacked model followed by a NN model in this architecture. We later show that the choice of the number of nodes of the hidden layers in each of these models can impact the system performance. Both models try to learn spatial and temporal features respectively.

Figure 6. Proposed System Architecture

From (10), we have the input $X_j =$

$$p\left(F_i(L(k)_{out})_1,\ F_i(L(k)_{out})_2, \ldots,\ F_i(L(k)_{out})_{T'} \middle| \left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)\right) \qquad (11)$$

$$H_j = f(I_j);\quad I_j = \sum_{j=1}^{n} W_{kj} X_j \qquad (12) \qquad\qquad O_k = f(I_k);\quad I_k = \sum_{j=1}^{n} W_{kj} H_j \qquad (13)$$

Substituting (11) and (12) in (13), we get:

$$O_k = f_o\left(\sum_{j=1}^{n} W_{kj}\, f_h\left(\sum_{j=1}^{n} W_{kj}\, p\left(\begin{matrix} F_i(L(k)_{out})_1,\ F_i(L(k)_{out})_2, \ldots,\ F_i(L(k)_{out})_{T'} \mid \\ \left(\sum_{j=1}^{n}(F_i(L(j)_{in})_1, \sum_{j=1}^{n}(F_i(L(j)_{in})_2 \ldots, \right. \\ \left. \sum_{j=1}^{n}(F_i(L(j)_{in})_T\right)\end{matrix}\right)\right)\right) \qquad (14)$$

The activation functions F_i tested for the scope of this paper are given in table 1 along with their mathematical representation. In our model the pre-processed data of shape (2602, 1, 4) with three inflows and one outflow according to equation (1) is be fed into the model and the respective link inflow and outflow values for the next time interval can be generated through the LSTM-NN. The shape dimensional values in (2602,1,4) represents the number of samples, batch number, variable features or corresponding link values, respectively. For each model iteration a separate validation set of similar shape (650, 1, 4) as of training data is used for the performance analysis measures. The final model parameters including the number of LSTMs and NNs chosen along with activation function are further discussed in the experiments section.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

	Activation Function (*g*)	Mathematical Implementation
1.	sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}};\ \sigma(x) \in [0,1]$
2.	softmax	$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}};\ j = 1,2,\dots,K;\ \sigma(x)_j \in [0,1]$
3.	tanh	$\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}};\ \tanh(x) \in [-1,+1]$
4.	relu	$f(x) = \max(0,x);\ f(x) \in [0,\infty)$

Nomenclature: softmax represents the normalised exponential function for multiclass logistic function flow values in our case, that makes K-dimensional vector x to have values in range [0, 1] that all add up to 1.

Table 1. Layer activation functions considered.

- **Feed Backward-Loss and Optimiser Function:** The second part of the system design considers the optimisation function and the loss function while updating the feed forward model weights before the next iteration. The iterative back-propagation allows the LSTM architecture to learn the temporal correlations amongst the intra node links whereas as the connected NN layer help learns the spatial dependencies. A set of optimisation strategies and loss functions considered in the experiments are given in table 2 & 3, respectively whose relative performances are evaluated in the process.

	Optimisation Function (*X*)	Mathematical Representation
1.	Stochastic Gradient Descent (SGD)	$w_{t+1} = w - \eta \left[\sum_{i=1}^{N} \frac{\nabla Q(w_i)_t}{N}\right] + \alpha \Delta w;$
2.	Adaptive Gradient Algorithm (Adagrad)	$w_{t+1} = w_t - \frac{\eta}{\sqrt{G_t+\varepsilon}} \odot g_t$
3.	Root Mean Squared Propagated Gradient Descent (RMSprop)	$w_{t+1} = w_t - \frac{\eta}{\sqrt{E(G_t)+\epsilon}} \odot g_t$

Nomenclature: $\boldsymbol{w_i} = (\bar{\boldsymbol{y}}_i - \boldsymbol{y}_i)^2$ *,* η *is the learning rate,* α *is the learning momentum factor,* g_t *is the iteration gradient,* $G_t = \sum_{i=1}^{N} g^2{}_{t,i}$ *is the diagonal.*

Table 2. Optimisation Strategies considered.

	Loss Function (*J*)	Mathematical Loss Representation
1.	Mean Squared Error (L2 loss)	$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2$
2.	Mean Absolute Error (L1 loss)	$MAE = \frac{1}{N}\sum_{i=1}^{N} abs\,(y_i - \lambda(x_i))$
3.	Mean Squared Logarithmic Error	$MSLE = \frac{1}{N}\sum_{i=1}^{n}(\log(\bar{y}_i + 1) - \log(y_i + 1))^2$
4.	Poisson	$poisson = \frac{1}{N}\sum_{i=1}^{N}(\bar{y}_i - y_i) * log\,(\bar{y}_i + \varepsilon)$
5.	Cosine	$cosine = cosine(\bar{y}_i - y_i)$
6.	Cosine Proximity or Cosine Distance	$cp = -\frac{\sum_{i=1}^{N}(\bar{y}_i * y_i)}{\sqrt{\sum_{i=1}^{N}(y_i)^2} * \sqrt{\sum_{i=1}^{N}(\bar{y}_i)^2}}$
7.	Logarithmic Hyperbolic Cosine	$logcosh = \sum_{i=1}^{N}\log(\cosh(\bar{y}_i - y_i))$
8.	Hinge	$hinge = \frac{1}{N}\sum_{i=1}^{N}\max(0, m - \bar{y}_i * y_i)$
9.	Kullback Leibler Divergence	$kl = \frac{1}{N}\sum_{i=1}^{N}(y_i * \log(y_i)) - \frac{1}{N}\sum_{i=1}^{N}(y_i * \log(\bar{y}_i))$

Nomenclature: $\bar{\boldsymbol{y}}_i$ *is the model last layer predicted value,* $\boldsymbol{y_i}$ *is the actual value,* $\boldsymbol{\lambda}$ *is the rate of absolute change set initially* $\boldsymbol{m}$ *is the threshold margin value already set for the hinge cost function.*

Table 3. Cost / loss estimation functions considered.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

A shallow LSTM-NN architecture is effective in capturing the spatio-temporal dependencies on node level with defined topological link order and this can be extended to further inter connected nodes and links. Thus, in the next section we perform experiments with varying parameters including loss function and activations which are given in table 1 and 2 respectively. The experimental run involves searching for the best parameters for both the two defined stages from that we hope to analyse the performance measures for best data driven objective function determination.

F) Performance Metrics

For the performance measure for the proposed model, we consider the root mean square error (RMSE) as widely used by researcher's community in the field of machine learning. We consider validation RMSE as our major model performance indicator. The formula given in equation (15) is the mathematical representation of RMSE.

$$RMSE = \left\{\frac{1}{N}\sum_{n=1}^{N}(|\overline{y_n} - y_n|)^2\right\}^{1/2} \quad (15)$$

where in equation (15), N represents the number of validation samples used for the error calculation, $\overline{y_n}$ is the predicted output and y_n is the original value observed by model.

V. EXPERIMENTAL RESULTS

In this section we show how the hyper-parameters of the proposed LSTM-NN network are optimised based on the network's performance using the Hatfield node junction data. The following notation is observed. Let $g \rightarrow$ Activation Function, $X \rightarrow$ Optimisation Function, $J \rightarrow$ Loss Function, $n \rightarrow$ Number of nodes in hidden layer, $\boldsymbol{J_{opt}}$ and $\boldsymbol{X_{opt}}$ are the optimised output values of J and X respectively.

Hyper-parameters optimisation is carried as a three-stage process whereby we first determine optimal values of J and X using Algorithm A. These optimal parameters are in turn used by Algorithm B to determine the optimal parameters of n. It is worth noting that n takes only 2 sets of values in Algorithm A to determine $\boldsymbol{J_{opt}}$ and $\boldsymbol{X_{opt}}$ whereas in principle several other combinations exist, and they are not considered at this point; instead they are optimised in the second stage using Algorithm B.

A) Finding Best Fitting Loss and Optimisation Functions

Firstly, we compare the performance measure by changing the loss functions Jalong with the optimisation techniques X. We compare nine different loss functions for our data model including the most common ones majorly used in data regression problems like mean square error, mean absolute error, mean squared logarithmic error, Poisson, cosine and the probability based logarithmic hyperbolic cosine, cosine proximity, hinge and lastly the cross entropy based Kullback-Leibler divergence. The best performing loss function $\boldsymbol{J_{opt}}$ is declared based on the minimum RMSE error.

The hybrid LSTM-NN model training is carried out by two different layer configurations of n = (35, 5, and 5) and (45, 20, 20) at different instances each with three different optimisers used. Each layer configuration corresponds to the (LSTM-layer1, LSTM-layer2, and NN-layer) respectively. But for each of them the activation function g for the respective layers was taken as constant i.e. (sigmoid, sigmoid, sigmoid) for the loss function versus the optimiser function performance test. The optimiser we used are the simple stochastic gradient descent (SGD), to the adaptive gradient algorithm (Adagrad) and running average-based root mean squared propagated gradient descent (RMSprop). Performance bar graphs in figure 7 shows that the minimum validation RMSE is achieved by the RMSprop among all the three optimiser which indeed is true in our case as the learning rate of the optimiser better adapts to the running average of time series then just simply considering the previous time interval. And the least RMSE is achieved by the

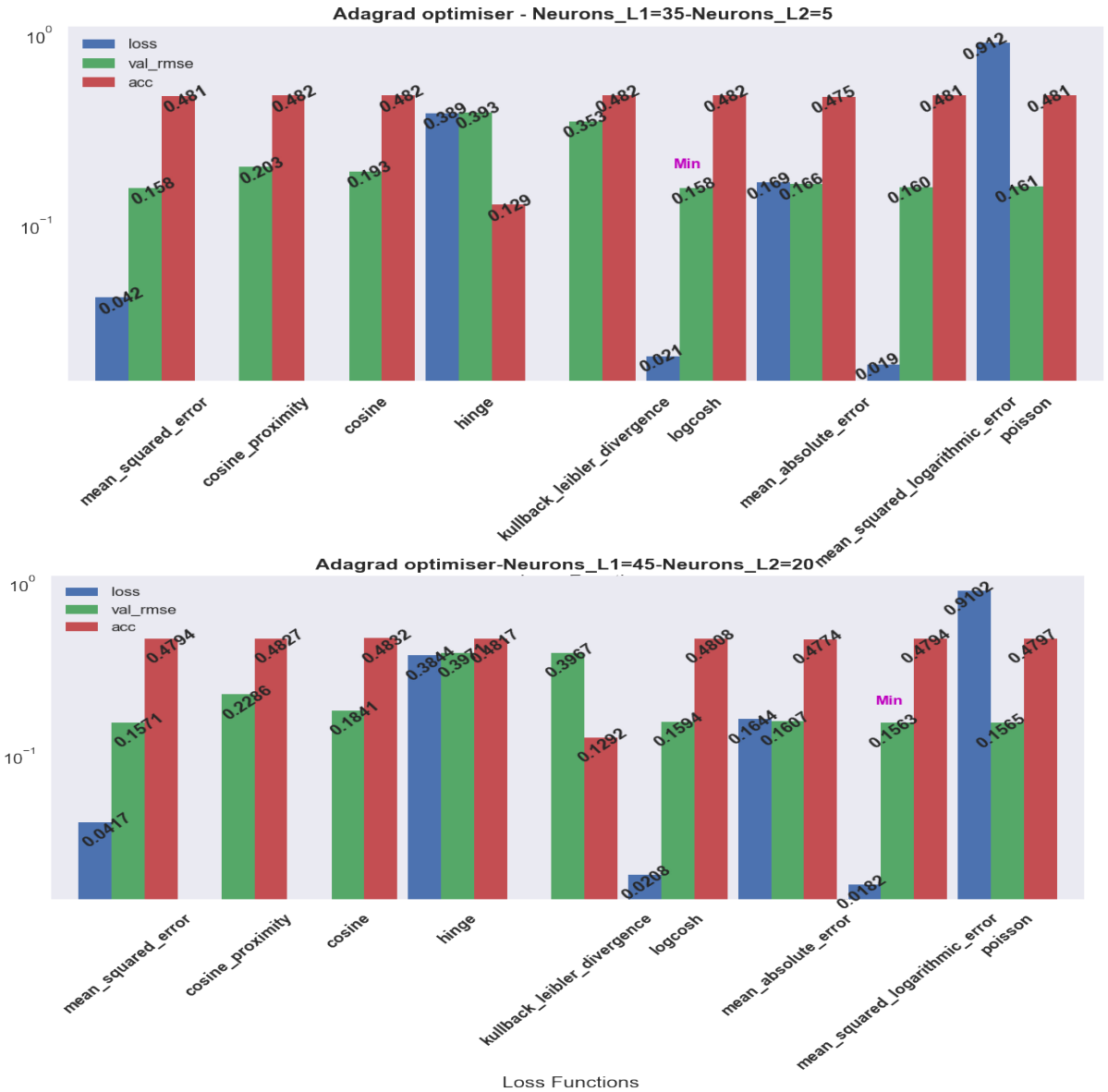
ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

(45, 20, 20) layer configuration. The training loss, accuracy and validation RMSE for each of the instances are shown in figure 7. All three metrics reflect one and the same result.

Algorithm A: Hyper parameter Optimization - Loss (J_{opt},) and Optimisation Functions (X_{opt})

function Hyper parameter Optimisation $(J, X, g, n, J_{opt}, X_{opt})$

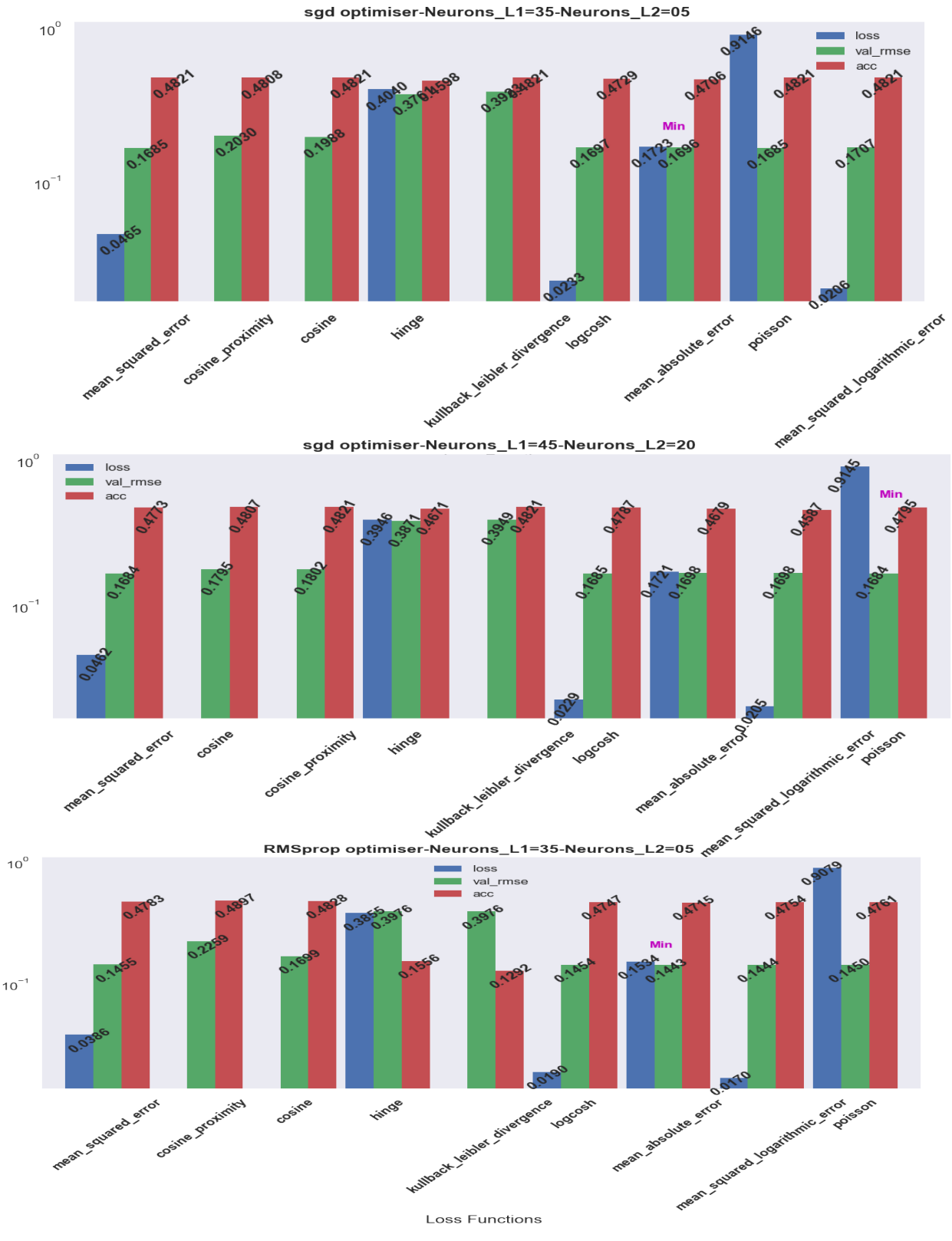
1. **Input:** *Performance evaluate loss functions* $\boldsymbol{J}$ ***(dimensionality=9)***
2. **Compute** RMSE
3. **Output:** J_{opt}
4. **Input:** $J_{opt}, g\ (dimensionality = 4), n\ (dinesionality = 2), X(dimensionality = 3)$



ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 5, Issue 10, October 2018



ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 5, Issue 10, October 2018

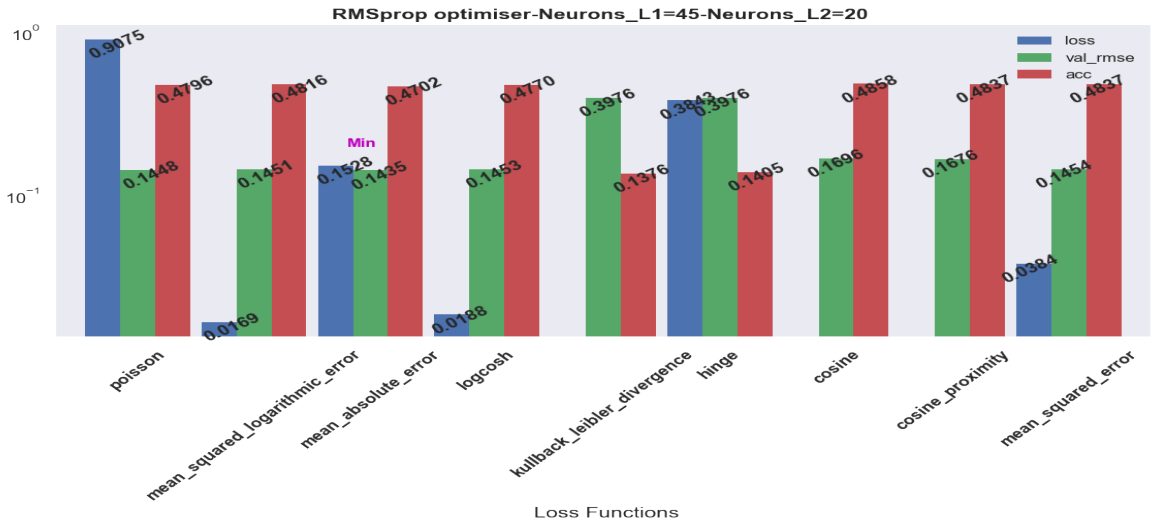


Figure 7. Performance report comparing three different optimisation techniques versus the loss functions with two different layer configurations.

B) Layers LSTM Units

Algorithm B: Hyper parameter Optimisation – Number of Hidden Layer Nodes, n_i of LSTM Layers

function Hyper parameter Optimisation ($J_{opt}, X_{opt}, n_{opt}$)

5. **Input:** *Performance evaluate Number of Hidden Layer Nodes,* ***n (dimensionality =20)***
6. **Compute RMSE**
7. **Output:** n_{opt}

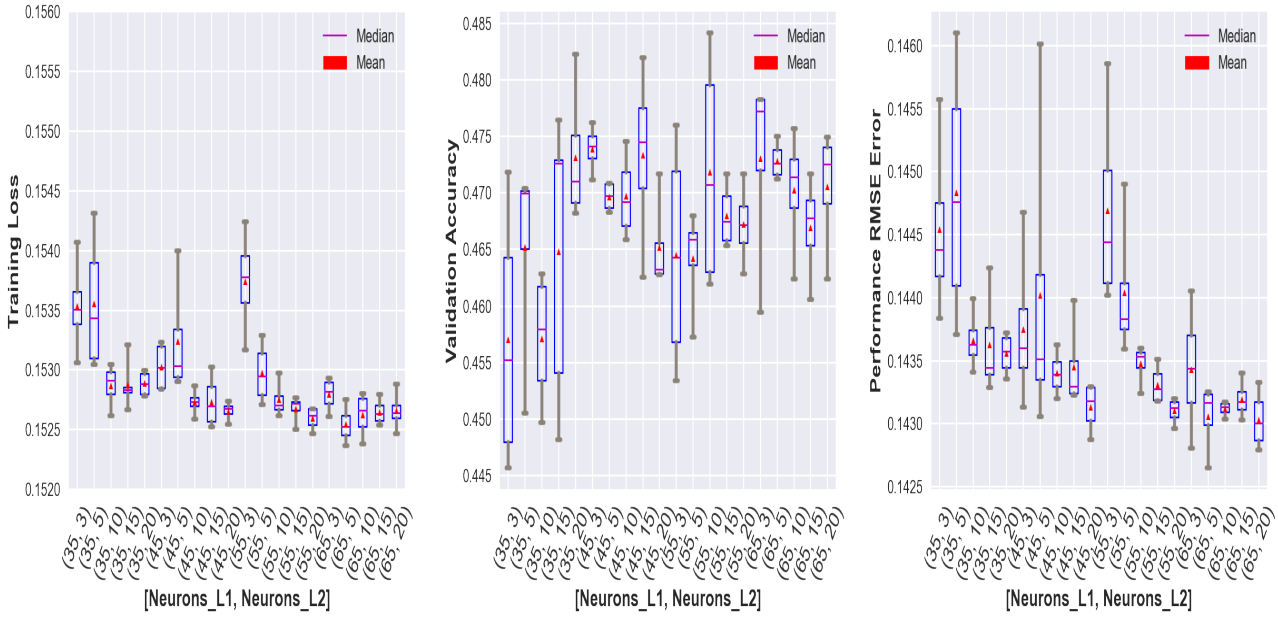


Figure 8. Performance Evaluation of Hyper-parameter *n* (Layer1, Layer2)

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 5, Issue 10, October 2018

In the second stage, we consider experimenting with n , the varied number of layer one, layer two LSTM units and NN layers. Using the optimal performing (X_{opt}) RMSprop optimisation technique and the best performing (J_{opt}) Mean Absolute Error (MAE) as a loss function we quest for the best suited LSTM layer unit numbers (n_{opt}) that minimises the validation RMSE exhibited by the model. The final performance plot in the form of boxplot with mean and median of four iteration runs made with each configuration is shown in figure 8. As before, any of the performance metrics may be used, but we show all three metrics for better clarity. Figure 8 has a notation [Neurons_L1, Neurons_l2] in which Neurons_L1 refers to the number of units in the hidden layers of both LSTM layers and Neurons_L2 refers to that of the NN layer as shown in the system design in figure 6.

C) Effect of Layer Activation functions

In the third stage, we analyse the architecture based on the choice of different layer activation functions, g. From Algorithms A and B, we consider the determined optimum performing RMSprop optimiser (X_{opt}), MAE as a loss function (J_{opt}) and n_{opt} = (65,65, 5) as the chosen final layer LSTM unit configuration. This is because the (65,65,5) combination exhibits the lowest mean validation RMSE out of all the configurations tested as shown in figure 8. Algorithm C tests all the combination of layer activation functions from table 1. We find that the least validation RMSE of 0.1398 is exhibited by the relu-tanh-relu configuration as shown in figure 9. The experimental result heat map in figure 8 shows that tanh does generalise the objective function well enough compared to softmax and sigmoid. This is because tanh as given in table 1 has a range of [-1, 1] and the negative first derivative is not a constant which is the property common to both sigmoid and softmax activation functions.

Algorithm C: Hyper parameter Optimisation – Activation Function (g)

function Hyper parameter Optimisation ($J, X, n, J_{opt}, X_{opt}, g_{opt}$)

1. **Input:** *Performance evaluate activation functions* g *(dimensionality=4)*
2. **Compute** RMSE
3. **Output:** g_{opt}

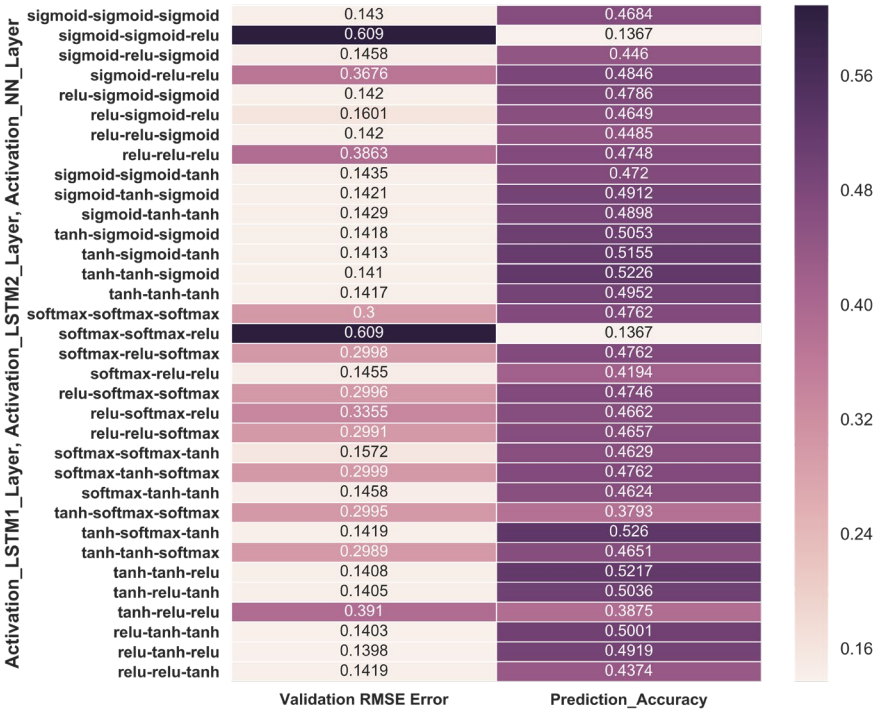


Figure 9. Performance Comparison of activation function combinations.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 5, Issue 10, October 2018

VI. CONCLUSION AND FUTURE WORK

To forecast the traffic flow in transportation networks several methods have been proposed by many researchers. During the survey it is seen that the flow prediction using conventional statistical and latest machine learning techniques starting from simple KNN to the latest deep ANN and time series LSTMs are highly effective in determining the spatiotemporal features which are crucial to traffic flow forecasting. In this paper we showed the spatiotemporal flow data remodelling in the form of topological objective function and exhibited the performance comparison of LSTM-NN with architecture parameter tunings. LSTM and ANN learns the temporal and spatial features respectively. The network is simple and fast enough for online data learning with dedicated geographical junction weight matrices for future training models. Future recommendations might include the local weather and incident data in combination with the objective function.

REFERENCES

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 3–19, 2014.
[2] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *Eur. Transp. Res. Rev.*, vol. 7, no. 3, pp. 1–9, 2015.
[3] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 50–64, 2014.
[4] M. T. Asif *et al.*, "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, 2014.
[5] J. Xin and S. Chen, "Bus Dwell Time Prediction Based on KNN," *Procedia Eng.*, vol. 137, pp. 283–288, 2016.
[6] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. Part C Emerg. Technol.*, vol. 62, pp. 21–34, 2016.
[7] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting," *Neurocomputing*, vol. 179, pp. 246–26, 2016.
[8] S. Oh, Y. Kim, and J. Hong, "Urban Traffic Flow Prediction System Using a Multifactor Pattern Recognition Model," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2744–2755, 2015.
[9] J. Amita, S. S. Jain, and P. K. Garg, "Prediction of Bus Travel Time Using ANN: A Case Study in Delhi," *Transp. Res. Procedia*, vol. 17, no. December 2014, pp. 263–272, 2016.
[10] Q. V. Le I.lSutskever, OV.inyals, "Sequence to Sequence Learning with Neural Networks," in *Neural Information Processing Systems Conference*, 2016, pp. 1–9.
[11] L. Deng and N. Jaitly, "Deep Discriminative and Generative Models for Pattern Recognition," pp. 1–26, 2015.
[12] G. B. Zhou, J. Wu, C. L. Zhang, and Z. H. Zhou, "Minimal gated unit for recurrent neural networks," *Int. J. Autom. Comput.*, vol. 13, no. 3, pp. 226–234, 2016.
[13] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *Proc. - 2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2016*, pp. 324–328, 2017.
[14] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.
[15] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. Part C Emerg. Technol.*, vol. 54, pp. 187–197, 2015.
[16] W. Feng, Wei Feng, W. Feng, and Wei Feng, "PDXScholar Analyses of Bus Travel Time Reliability and Transit Signal Priority at the Stop-To-Stop Segment Level," 2014.
[17] Highways England, "Highways England – Data.gov.uk – Journey Time and Traffic Flow Data April 2015 onwards – User Guide," no. April, pp. 1–14, 2015.
[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," 2014.

AUTHOR'S BIOGRAPHY

Arsalan Rahi graduated in Electronics Engineering from Ghulam Ishaq Khan Institute (GIKI) Institute Pakistan in 2014.He finished his MSc in Embedded Intelligent Systems from Hertfordshire University (UH) United Kingdom in 2015. He is now a PhD candidate in Biometrics and Media Processing department in UH since 2016 and working as a data scientist at University Bus Limited (UNO). Field of interest includes smart transport management systems, IOT, Artificial Intelligence, data analytics with research interests lies in the latest machine learning algorithms implementations.

Dr Soodamani Ramalingam is a Senior Lecturer in the School of Engineering and Technology, University of Hertfordshire since 2006. She has several years of academic and research experience in the UK, Singapore and Melbourne. She received her PhD(CSE) award from the University of Melbourne, Australia in 1997 and her M.E.(CS) and B.E.(ECE) degrees from PSG College of Technology, Bharathiar University in India. Her research expertise is in Computer Vision and Machine Learning, Biometrics, Image Processing and Fuzzy Logic. Applications areas include Automatic Number Plate Recognition ANPR), 3D Face Recognition and Intelligent Transportation Systems and Energy. She has over 65 international conference and 30 journal publications in related areas of research. She is a member of IEEE and Biometrics Institute (UK).