



# Deep Learning-Enhanced Hybrid Fruit Fly Optimization for Intelligent Traffic Control in Smart Urban Communities



Ghamya Kotapati<sup>1</sup>, Mohd Anwar Ali<sup>2</sup>, Ramesh Vatambeti<sup>3\*</sup>

<sup>1</sup> Department of AI & ML, School of Computing, Mohan Babu University, 517102 Tirupati, India

<sup>2</sup> Department of Information Technology, MLR Institute of Technology, 500043 Hyderabad, India

<sup>3</sup> School of Computer Science and Engineering, VIT-AP University, 522237 Vijayawada, India

\* Correspondence: Ramesh Vatambeti (ramesh.v@vitap.ac.in)

Received: 05-01-2023

Revised: 06-07-2023

Accepted: 06-12-2023

**Citation:** G. Kotapati, M. A. Ali and R. Vatambeti, “Deep learning-enhanced hybrid fruit fly optimization for intelligent traffic control in smart urban communities,” *Mechatron. Intell Transp. Syst.*, vol. 2, no. 2, pp. 89–101, 2023. <https://doi.org/10.56578/mits020204>.



© 2023 by the authors. Licensee Acadlore Publishing Services Limited, Hong Kong. This article can be downloaded for free, and reused and quoted with a citation of the original published version, under the CC BY 4.0 license.

**Abstract:** The rapid urbanization accompanying the evolution into “smart” communities presents numerous challenges, not least of which is the significant increase in road vehicles. This proliferation exacerbates congestion and accident rates, posing major barriers to the successful implementation of innovative technologies such as Wireless Sensor networks (WSNs), surveillance cameras, and the Internet of Things (IoT). Accurate traffic flow prediction, a crucial component of these technological initiatives, requires a reliable and efficient methodology. This research explores the implementation of an intelligent traffic control system that employs a Transferable Texture Convolutional Neural Network (TTCNN). The design of this system eschews the traditional pooling layer, instead incorporating three convolutional layers and a single Energy Layer (EL). This configuration facilitates the provision of real-time traffic updates, which can enhance the utility and efficiency of the smart city infrastructure. A model inspired by the Hybrid Fruit Fly (HFFO) optimizes the system’s hyperparameters. The application of HFFO to the TTCNN showcases the potential for improved accuracy in traffic flow prediction. Simulation results suggest that the HFFO provides superior organizational boundaries for the TTCNN, enhancing the overall accuracy of the model’s predictions. The hybrid forecasting method discussed herein demonstrates its potential to outperform other established techniques. This investigation sheds light on the potential benefits of applying deep learning algorithms and hybrid models in the context of traffic flow prediction and control, contributing to the ongoing development of smart urban communities.

**Keywords:** Hybrid fruit fly optimization; Wireless sensor networks; Traffic flow prediction; Convolutional neural network; Vehicles; Surveillance camera

## 1 Introduction

Smart vehicles are now a practical choice for urban residents because to the widespread availability of autonomous driving technologies that have been shrunk in recent years. By performing tasks like collision avoidance, identification, autonomous vehicles lighten the load on human drivers. And because of their enhanced fuel efficiency and reduced emissions, vehicles equipped with autonomous driving technology can better manage traffic flow and minimize congestion [1]. Providing safe and dependable transportation for the elderly and the disabled, solving parking issues, and reducing the number of accidents due by human mistake are all ways in which autonomous vehicles help people in their everyday lives [2].

People today spend the vast majority of their waking hours away from home [3]. This includes commutes to and from work, visits to entertainment venues and retail establishments, and trips to and from the city centre. This undoubtedly threw a wrench into daily mobility and prompted the creation of parking services so that people wouldn’t have to waste time and gas driving aimlessly about downtown in quest of a parking spot. On the one hand, this results in more greenhouse gas emissions and harms, it exacerbates driver annoyance and urban traffic, both of which are major contributors to increased vehicle collisions [4].

Due to the evolving nature of the global economy and contemporary lifestyle, urban areas have been expanding at a dizzying clip recently. When it comes to city planning and urban development, information and communication

technologies are essential [5]. The goal of these new knowledges and networked smart devices is to create smart cities and maximize the efficiency of urban services that are directly related to inhabitants [6]. Every aspect of daily living benefits from the technological advancements made possible by smart cities [7, 8]. Increases in automobile ownership over the previous five years have been alarming, leading to gridlock, accidents, and even driver disease as a result of the stress and irritation they generate [9]. Poor management, and the need for drivers to move around aimlessly in search of parking spaces in already-crowded areas are to blame [10].

Major technological advancements, shifts in corporate practices, and global environmental issues have all contributed to the emergence of the study of “smart cities and communities” (SCC). (ICT) that link infrastructure, to data monitoring and asset management systems are among the technologies that make SCC possible [11]. Internet of Things (IoT) is another technology that allows for even the tiniest devices to attach to the Internet and report on their status [12]. IoT technology enables the interconnection of devices from many sectors, including the transportation network, power plants, and private homes [13].

Processing and analyzing the obtained data is necessary for realizing the possible of SCC in any request area. Finding connections, causes, and patterns in data is made possible by AI. The updated AI coaching and user behaviour improvement suggestions might be more specific and useful [6, 7]. Many problems are still to be solved in terms of SCC, and some of them are detailed below. Over the past 60 years, there have been enormous shifts in the way businesses function. The decline of manufacturing and the expansion of services to businesses and individuals are seen in the GDP trends from 1947 to 2009. About half of the United States’ industrial GDP has been lost. Conversely, GDP growth for commercial and professional services is 400% [14]. The percentage of GDP contributed by services has climbed from 72.8% to 77.4% during the past 17 years. The study’s most significant findings are as follows.

- For the purpose of traffic flow prognostication, we present a novel (TTCNN).
- For our TTCNN architecture, we employ the EL (energy layer). We can keep texture information, keep the productivity vector small, and improve the replica’s learning capacity in this way.
- Improve the quality of the solutions generated by the original fruit-fly optimization (FFO) procedure by creating a hybrid version of the algorithm that uses swarm intelligence.

The remainder of the paper follows this construction: The relevant literature is presented in Section 2, shadowed by a problem statement in Section 3. Section 4 delivers a concise summary of the optional perfect. In Section 5, we display the consequences of our comparison of the projected model to the already available validation methods. Section 6 provides a conclusion.

## 2 Related Works

In order to anticipate traffic flows, Bao et al. [15] offer a new technique based on (ST-CGCN). Based on physical locations, past data records, and external interference among traffic nodes, we first build a matrix. To further enhance the joint modelling capabilities of spatial-temporal characteristics and external influences, these fused into a multifaceted matrix by incorporating self-learning dynamic weights. Next, modules for extracting spatial characteristics and temporal information are developed so that dynamic spatial-temporal aspects may be described. The spatial feature extraction component is made up of a residual unit and a graph convolution operator using a suggested make up the temporal feature extraction component.

Djenouri et al. [16] have explored a novel convolutional system for predicting urban traffic flows in an edge IoT setting by combining graph prediction in a single pipeline. First, the raw data set of urban traffic road networks is pre-processed using a connected graph pre-processing procedure to eliminate unwanted noise. The road network is effectively explored by employing an outlier identification approach to filter out unnecessary designs and noise. The generated graph is then used to train an network, which is ultimately used to provide traffic predictions for the city. A novel branch-and-bound-based optimization approach is created to fine-tune the values of the planned framework. Multiple datasets and reference methodologies are used in a thorough review for contrast. The findings demonstrate that, especially with a high sum of nodes in the graph, the suggested framework performs solutions.

To deal with the novel crown epidemic’s extremely discontinuous and irregular character, Li et al. [17] offer a deep-space temporal traffic discrete wavelet transform (DSTM-DWT). First, DSTM-DWT dissects flow information into components such as trend, amplitude, and baseline. Second, we use a graph-based approach to designing the transportation network’s spatial connection, including newly available data on the crown pneumonia outbreak into the attributes of each node. The geographical correlation of each node is then determined using the graph convolutional network, while the temporal correlation is determined using the temporal convolutional network. This paper offers a graph memory network (GMN) for converting discrete magnitudes split by discrete wavelet transform into high-dimensional discrete features, which is a solution to the flow data epidemic. Once the traffic data has been forecasted using DWT, the traffic trend and discrete baseline may be separated out, and the GMN-predicted discrete model can be compared to the outcome of the inverse DWT.

A unique attention-based learning model has been proposed by Jia and Cai [18] to accurately flow at road crossings throughout an entire city. To begin, we look at how turning traffic is distributed over space and time.

Then, to predict the reversal of traffic, a four-part, end-to-end deep learning framework is developed. To learn spatial dependencies and sparseness, we adapt a graph convolutional network, and to learn temporal dependencies and fluctuations, we design a gate recurrent unit mechanism. Our model was trained and tested using trajectories collected from taxi rides in Wuhan, China. According to the outcomes, our model provides more accurate estimates of turning traffic flow than the present state-of-the-art representations.

Henry gas is a new method developed by Escorcia-Gutierrez et al. [19] for 6G-enabled vehicular networks. The primary goal of the described HGSODL-TFF method is to foretell the volume of traffic in the 6G equipped VANET. Furthermore, the traffic data is initially preprocessed in the HGSODL-TFF model with a z-score normalization technique. In addition, a deep belief network (DBN) perfect is used to predict traffic volumes with impressive accuracy. The DBN model's forecasting performance may be optimized by adjusting hyperparameters like the count, and batch size using the HSGO method. The HGSODL-TFF model is experimentally validated using test data, and the findings are analyzed in detail. According to the simulation findings, the HGSODL-TFF model is superior than the other contemporary methods.

Air traffic flow may be predicted using a convolution network (TAaDGCN) proposed by Cai et al. [20], which takes into explanation both the airspace structure and the flow paths. To begin, we build a spatial capture the interdependencies between neighbouring and OD sectors. As a next step, we employ a (SE) block to symbolize potentially connected flight sectors in order to incorporate long route information. In addition, a module is used to retrieve historical aspects of the input sequence in order to define the temporal evolution pattern. A spatio-temporal block, including several geographical and temporal relationships, is built from the aforementioned blocks. The experimental findings from using real-world flight data show that methods in terms of prediction performance, especially those that disregard the sector's spatial structure.

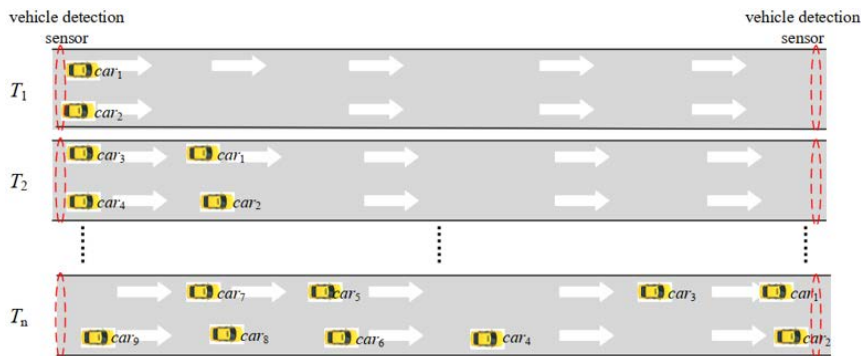
### 3 Challenges Addressed by SCC

Over the past 60 years, there have been enormous shifts in the way business's function. The decline of manufacturing and the expansion of services to businesses and individuals are seen in the GDP trends from 1947 to 2009. About half of the United States' industrial GDP has been lost. Conversely, GDP growth for commercial and professional services is 400% [14]. The percentage of GDP contributed by services has climbed from 72.8% to 77.4% during the past 17 years [21], while the percentage contributed by industry has declined from 22.5% to 18.6%.

This tendency is mostly attributable to digitalization with the aid of ICT, and with the most recent developments in computation power, AI is emerging as a crucial technology to exploit the data and advance the services. Consumption of energy, especially from nonrenewable sources like oil, is another difficulty. In 2017, primary energy consumption in Europe was 1561 Mtoe, 5.3% more than the EU goal for 2020. In 2018, crude oil accounted for 36% of all energy production, followed by natural gas (21%), renewables (15%), solid fossil fuel (15%), and nuclear (13%). Industry accounts for 31% of final energy consumption in the EU, followed by transport (28%), homes (25%), services (2%), according to a recent study [22].

## 4 Proposed Methodology

### 4.1 Mathematical Perfect of Traffic Flow in Road Sections



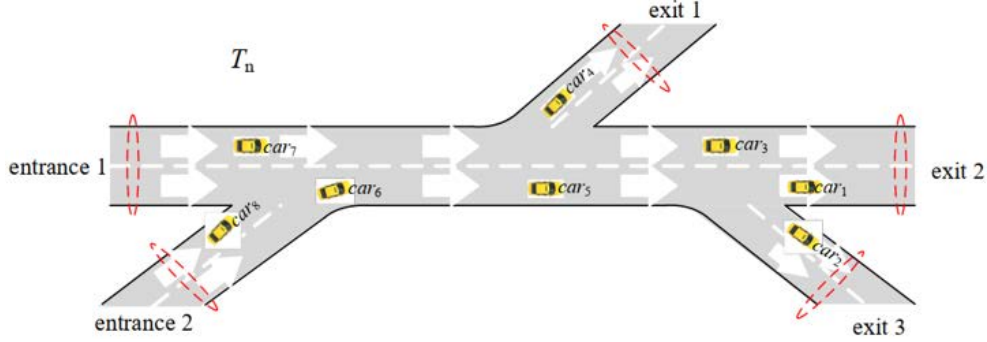
**Figure 1.** Extraction of the sum of vehicles on a road section

Vehicle volume, average speed, and vehicle density are all indicators of a road segment's traffic flow. The flow shift and the threshold value of the index [23] are both unique to individual stretches of road. One-way streets have only one oncoming and one going lane. Figure 1 illustrates how the sum of cars arriving and departing the road segment during 5(n 1) 5n minutes, as well as the sum of vehicles already present in the road segment, affect the

traffic flow along that section of road. This allows us to derive the following formula for determining the number of cars currently using this stretch of road at time  $k$ .

$$Q(k) = \sum_{i=1}^k q_{in}(i) - \sum_{i=1}^k q_{out}(i) + q_0 \quad (1)$$

where,  $q_0$  is the sum of vehicles comprised in the early period; and  $q_{in}(i)$  and  $q_{out}(i)$  are the number of vehicles incoming and section, correspondingly.



**Figure 2.** Extraction of the sum of vehicles on a road section

Figure 2 illustrates the process of vehicle count extraction along a multi-channel road stretch. Setting up several detection intersections and using the data obtained from these sensors to anticipate the sum of cars on the complicated road segments is required to estimate the volume of traffic on the multi-channel road segment. Here is how we may define  $k$ , the current number of cars:

$$\bar{Q}(k) = \sum_{i=1}^N \sum_{s=0}^k \bar{q}_{in}(i, s) - \sum_{j=1}^M \sum_{s=0}^k \bar{q}_{out}(j, s) + \bar{q}_0 \quad (2)$$

where,  $\bar{Q}(k)$  is the entire vehicle capacity on the multifaceted road segment;  $\bar{q}_0$  is the sum of vehicles at the early instant; and  $\bar{q}_{in}(i, s)$  and  $\bar{q}_{out}(j, s)$  are the sum of vehicles incoming and exiting the unit at the  $s$ th instant from the  $i$ th on-ramp and  $j$ th off-ramp, correspondingly.

## 4.2 Network Architecture

For the purpose of traffic forecasting, the TTCNN framework is described. The suggested deep CNN takes into account these three picture characteristics: First of all, certain description patterns are considerably smaller than the source picture, but the convolution filter can still locate the pattern if its size is equivalent to the size of the convolution filter mask. Second, certain areas of the picture might make use of particular forms or designs. Convoluting the full source picture is another way to define these models. In addition, the downsampled pixels play a crucial role in the max-pooling layer without altering the overall form of the original mammography.

Two pooling layers, a third convolution layer, and an output layer (EL) are all part of the proposed TTCNN. The fully connected (FC) layer is then shadowed by a softmax layer. By averaging the corrected activation output, EL préçises the feature maps of the previous convolutional layer. A value, representing the energy response of a filter bank, is returned for each feature map. This design not only improves efficiency in learning texture functions but also uses less memory and compute. This compromise between speed and processing time is made possible by EL. The primary motivation for implementing this layer is to maintain the original layer's data flow. The output of EL is flattened and delivered to the concatenation layer after the final pooling layer. Through this link, information about the image's contours and textures is flattened into a new vector and sent across the completely linked layer. Table 1 provides a comprehensive explanation of the projected network, including input and output dimensions. Eq. (3) provides a mathematical formula for calculating the convolution layer's output size.

$$\text{output} = \frac{I_a - I_b + 2S}{\varrho + 1} \quad (3)$$

where,  $I_a$  and  $I_b$  signifies the filter size correspondingly,  $S$  denotes the  $\varrho$  is the stride value.

**Table 1.** Projected texture CNN construction layers

Layers	Types	Padding	Input size	Kernel extent to form each feature map	Stride	Output size
1	Convolutional layer 1	[1 1 1 1]	64×64×1	5×5	[1 1]	62×62×16
2	Max pooling layer 1	[1 1 1 1]	62×62×16	2×2	[2 2]	32×32×16
3	Convolutional layer 2	[1 1 1 1]	32×32×16	5×5	[1 1]	30×30×32
4	ReLU					
5	Max pooling layer 2	[1 1 1 1]	30×30×32	2×2	[2 2]	16×16×32
6	Softmax layer	-	-	-	-	-
7	Classification layer	-	-	-	-	-
8	Convolutional layer 3	[1 1 1 1]	16×16×32	3×3	[1 1]	16×16×64
9	ReLU					
10	EL	-	16×16×64	-	-	128×1
11	Dropout	-	128×1	-	-	128×1
12	FC2	-	1024×1	-	-	2×1
13	FC1	-	128×1	-	-	1024×1
14	Dropout	-	1024×1	-	-	1024×1

After that, we use three convolution 16 and 32 channels, respectively, for their output. The third convolution layer, with a kernel size of 3×3, and 64 output channels, is investigated as an intermediary texture attributes. The convolution layer can only be used to learn a maximum of 31,744 parameters, which are then computed using the methods in Eqns. (4) and (5):

$$\xi_v = \zeta_v \times (I_k \times \varrho + 1) \quad (4)$$

$$\xi_v = \zeta_v + X_k \times \varrho \times \zeta_v \quad (5)$$

where,  $\xi_v$  signifies the CNN layer limits,  $I_k$  signifies the kernel size, and  $\zeta_v$  denotes the channel amount.

At each convolution layer, the output of the neuron coupled to the input is calculated. The answer is the dot product of its mass and the lowest input field connected with it. A 16-kernel 32x32x16 output is created by the first convolution layer. The production of the neurons in the first convolution layer may be calculated using Eq. (6):

$$S_{\vartheta} = \sum_{\vartheta} C_{\vartheta} \times T_{\vartheta} + P_{\vartheta} \quad (6)$$

where,  $S_{\vartheta}$  is the feature maps used to generate an output,  $C_{\vartheta}$  is the feature maps used to generate input, and  $T_{\vartheta}$  is the weighted map. After that, the output of the last convolution layer is mapped onto an energy descriptor. After the third convolutional layer, energy layers are merged according to the specifications of the energy descriptor. It performs similarly to a texture description for a cluttered, thick surface. Eq. (7) explains the relationship:

$$EL(\xi, \vartheta) = \rho \left[ \sum_{i=1}^j T_i^{\omega} \vartheta_i + P \right] \quad (7)$$

where,  $EL(\xi, \vartheta)$  represents the EL weighted vector,  $j$  stands for the EL input influences, and represents the EL output layer. The link between the EL and FC layers is substantially smaller compared to the final traditional convolution layer, which reduces the number of trainable parameters. Furthermore, EL learns during both forwards and backwards propagation by retaining energy information from the previous layer. Additionally, EL enhances the network's general learning ability and simplifies the proposed system by decreasing the vector size of the subsequent FC layer. Learnable EL parameters can be determined by solving Eq. (8):

$$\xi_{EL} = \eta^m \times \eta^{m-1} \quad (8)$$

where,  $\xi_{EL}$  is the EL learnable limits,  $\eta^m$  is the current FC neuron, and  $\eta^{m-1}$  is the previous FC layer neuron.

Between the rectified linear unit (ReLU) layer, a batch normalization and activation function is utilized to expedite the training process. The internal covariance shift can be eliminated by employing batch normalization. The mean and standard deviation can be normalized to achieve this goal. Mean and Variance are determined using Eqns. (9) and (10), respectively, in the bulk normalization procedure.

$$\tau_Q = \frac{1}{n} \sum_i^n l_i \quad (9)$$

$$v_Q = \frac{1}{n} \times \sum_i^n (l_i - \tau_Q)^2 \quad (10)$$

where,  $\tau_Q$  and  $v_Q$  indicates the average and standard deviation, and  $n$  is the smallest possible batch size for the features in the  $1_i$  dimension. The number  $n$  equals 64 in our study. Eq. (11) describes how to determine the batch normalisation:

$$\lambda_i = \frac{\vartheta_i - \tau}{\sqrt{v^2 + \phi}} \alpha + A \quad (11)$$

where,  $a$  and  $A$  are the possible starting values for each output layer's learnable parameters. Eq. (12) computes the ReLU activation function, and Eq. (13) determines the ReLU layer's output:

$$\lambda_{i,j,k} = \max 0, \vartheta_{i,j,k} \quad (12)$$

$$\lambda_{\text{ReLU}} = \text{ReLU}(\text{Bnorm}(\text{Conv}(w, x))) \quad (13)$$

where,  $\lambda_{i,j,k}$  means characteristics of the final element, while  $(i, j, k)$  means characteristics of the first element. The subsequent shrinking of feature maps, weights, and calculations due to the pooling layer demonstrates that the control network has been overfit. Eq. (14) is a mathematical formula used to determine the max pooling layer:

$$M_{\text{pool}} = \max \left( 0, \sum_Q \vartheta^{k-1} T_{\vartheta} \right) \quad (14)$$

where,  $M_{\text{pool}}$  signifies the maps,  $Q$  indicates the input feature maps, and  $T$  signifies the vector. The work utilizes two max pooling layers, each with a kernel size of  $2 \times 2$ .

To avoid overfitting training data, the dropout layer is used throughout the weighted update phase to continuously delete a subset of random parameters. Drop editing is used throughout the weighted update process to delete a selection of random parameters to prevent overfitting of the training data. Because FC layers contain the most network-wide features, they are particularly sensitive to over-compatibility difficulties when training data is used. As a result, the dropout layer is established after the FC layer. To do classification, the softmax layer employs the loss function. For softmax, the valid probability values are between zero and one. Eq. (15) provides the mathematical formulation of the loss function.

$$k_l = \delta_j + \log \sum_i \exp(\delta_i) \quad (15)$$

where,  $k_l$  signifies the total loss and  $\delta_j$  having the class  $\delta$  which is  $i$ -th element. The goal is to minimise the probability dissimilarity among the true label and the assessed label computed by the softmax purpose in Eq. (16):

$$\lambda_i = \frac{\exp^{\delta_j}}{\sum_i \exp(\delta_i)} \quad (16)$$

Here, the proposed models' hyper-parameter tuning is carried out by improved fruit fly optimization procedure.

#### 4.2.1 Original fruit-fly optimization procedure

The original FFO algorithm takes its name and inspiration from the foraging habits of the fruit fly. The four-step original fruit-fly optimisation technique:

- initialization,
- population evaluation,
- osphresis foraging,
- vision.

As shown in Eq. (17), initial solutions (fruit flies) are created at random inside the provided lower and upper boundaries, with  $x_{i,j}$  denoting the i-th solution and the j subscript indicating the element's location within the i-th solution. Lower bound (lb) and upper bound (ub) stand for limits. rand is a uniformly distributed random sum generator.

$$x_{i,j} = \text{rand}(ub_j - lb_j) + lb_j \quad (17)$$

Following the initialisation of the population, the positions of the solutions are updated according to the phase, during which the solutions migrate at random with respect to their previous positions, as shown in Eq. (18), where the solutions' new positions are indicated by  $x_{i,j}^{(t+1)}$ , the current solution is  $x_{i,j}^{(t)}$ , rand() 2 [-1, 1]. The iteration counter is signified by t.

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} \pm \text{rand}() \quad (18)$$

After the position is updated, the fitness value of each solution is determined, and the greedy selection process decides whether the previous position or the new one should be maintained. The vision foraging phase is the next step in the algorithm. A new answer will replace an older one if its fitness value is higher; otherwise, the older solution will continue in the populace while the newer one is eliminated. Upon meeting the stop condition, the algorithm exits and delivers the optimal solution.

#### 4.2.2 Proposed hybrid FFO

The FFO is easy to customize because of its uncomplicated structure, few parameters, and flexibility in solving a wide variety of problems. Despite its benefits, however, this approach is not without its flaws. The method has the potential to become stuck in local minima, has a static position update technique, and is not very good at exploiting opportunities.

Experiments on unbounded functions revealed the algorithm's flaws. Because of its superiority in intensification, the firefly algorithm (FA) search mechanism has been added into the algorithm to improve exploitation. In addition, opposition-based learning (OBL) is presented as a means of enhancing the search space exploration process.

The acronym HEFFF stands for "hybrid enhanced fruit-fly firefly" to describe the name of the hybrid algorithm. At first, Eq. (17) is used to produce a random population. If the random number is less than 0.5, then the FFO search mechanism is used in even iterations; otherwise, opposition-based learning is performed; and the firefly search mechanism is used in odd iterations.

The definition of the inverse number,  $X'$  is given by Eq. (19).

$$x_i'^{(t+1)} = lb + ub - x_i^{(t)} \quad (19)$$

To determine how far apart two solutions are, the firefly search process uses Eq. (20), where  $r_{i,j}$  signifies the distance  $x_j$ . At zero distance,  $\gamma$  represents how alluring the healthiest firefly (the optimal option) is. The random number k is drawn from a Gaussian distribution with control parameters and alpha.

$$x_i^{(t+1)} = x_i^{(t)} + \frac{\beta_0}{1 + \gamma r_{i,j}^2} (x_j - x_i) + a(k - 0.5) \quad (20)$$

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^D (x_{i,k} - x_{j,k})^2} \quad (21)$$

According to reference [24], a dynamic stage is used for a control limit to further enhance the algorithm's efficiency, where a value continuously decreases from the beginning value ( $a_0$ ), is reached through the course of



the iterations. Eq. (22) defines the update of a value at each repetition as follows: where  $t$  is the current iteration,  $\text{MaxIter}$  is the extreme sum of repetitions,  $a(t)$  is the value of  $a$  at the present iteration, and  $u$  is the updated value  $a(t + 1)$ .

$$a^{(t+1)} = a^{(t)} \cdot \left(1 - \frac{t}{\text{MaxIter}}\right) \quad (22)$$

Algorithm 1 exemplifies the main phases of the projected method [25].

---

**Algorithm 1 Pseudo-code of projected model**

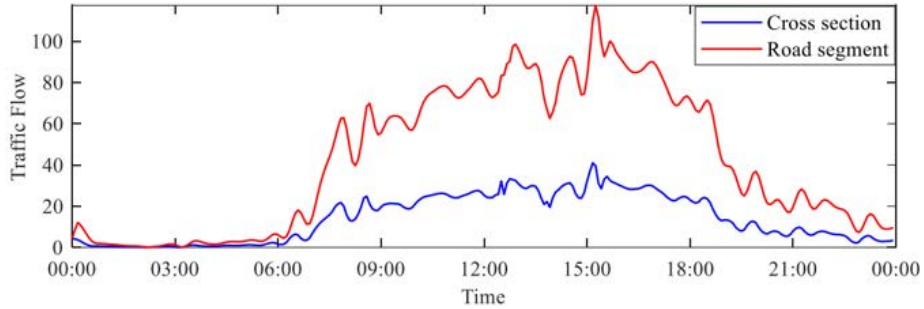
---

```

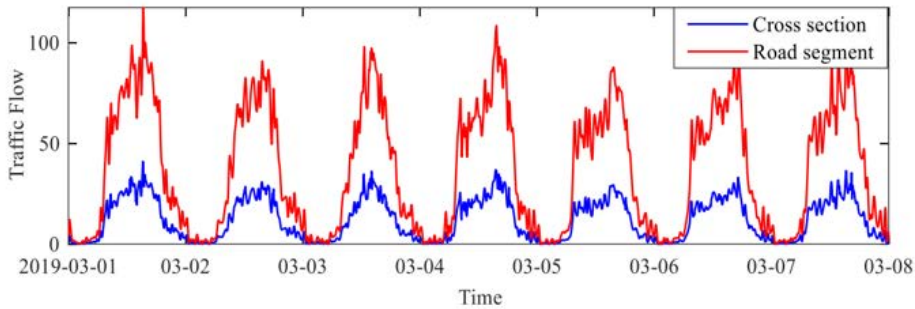
Initialize the population randomly by Eq. (17)
Initialize the FA parameters of  $\beta_0, \gamma, a$ 
Set the iteration counter  $t$  to 0 and define the termination criteria
Evaluate the fitness of each individuals
while termination criteria is not satisfied do
  for  $i = 1$  to  $N$  do
    if  $t$  is even then
      if  $\text{rand} < 0.5$  then
        Update the position according to FFO updating mechanism by Eq. (18)
      else
        Use OBL procedure by Eq. (19)
      end if
    else
      Perform FA search using Eq. (20)
    end if
  end for
  Evaluate the solutions, and if the new solution has better fitness than the old one, replace the old solution
  Sort the population according to the fitness value of the solutions and save the best solution
  Replace the worst solutions (30% in the population) by new random solutions
end while
Return the best solution

```

---



(a) A daylong examination of traffic volume comparison



(b) In a week, we'll compare traffic flows

**Figure 3.** The cross section of the road and its traffic flow are compared



## 5 Simulation Experiments

### 5.1 Dataset and Pre-Processing

Our experimental dataset consists of traffic counts recorded along California State Route 1 during the course of 31 days, beginning on March 1 and ending on March 31 [26]. The ratio of training data to validation data to test data is 6:2:2. We use data on traffic flows from March 1st through March 19th for our training set, March 20th through March 25th for our validation set, and March 26th through March 31st for our test set. The traffic data in the road segment scenario is computed using the mathematical model discussed in Section 2 because this dataset is a cross section dataset. Figure 3 depicts a comparison of daily and weekly traffic volumes on the road lane. It is clear that the amplitude, temporal correlation, and time lag of the road segment's traffic flow are all higher than those of the overall traffic flow.

### 5.2 Dataset and Pre-Processing

The suggested model's predictive power is graphically represented in Table 2 by calculating the bias among the anticipated value and the actual traffic flow of ten randomly selected samples. It demonstrates that the suggested model has better predictive accuracy than the state-of-the-art methods.

**Table 2.** The foretold value and real traffic of the projected model with existing models

Sample	Actual data	Proposed		LSTM		CNN	
		Bias (%)	Predicted value	Predicted value	Bias (%)	Predicted value	Bias (%)
5	14.01	3.26%	14.47	15.27	9.02%	16.10	14.92%
65	5.61	4.15%	5.84	5.47	3.27%	5.18	7.68%
99	39.84	4.97%	41.82	46.23	16.04%	39.67	0.44%
123	55.07	0.80%	54.63	54.76	0.56%	52.39	4.86%
138	55.85	0.03%	55.83	54.70	2.05%	59.80	7.07%
174	77.12	4.12%	80.3	76.06	1.37%	80.9	4.90%
175	74.56	2.33%	76.30	76.46	2.55%	87.37	17.19%
244	30.26	0.63%	30.07	28.99	4.20%	31.88	5.35%
261	28.64	8.41%	31.05	31.28	9.22%	33.64	17.47%
281	13.77	3.20%	14.21	15.65	13.65%	13.85	0.58%

In the above Table 2 represent that the proposed model with existing models. In this analysis we have used different sample analysis. In the 5th sample the actual data is 14.01 and the proposed model reached the foretold value is 14.47 and Bias (%) as 3.26% and the LSTM model the Predicted Value as 15.27 and Bias (%) value as 9.02% and the CNN model the predicted value as 16.10 and Bias as 14.92 respectively. In the 65th sample the actual data is 5.61 and the projected model reached the predicted value is 5.84 and Bias 4.15% and the LSTM model the Predicted Value as 5.47 and Bias (%) value as 3.27% and the CNN model the predicted value as 5.18 and Bias as 7.68 respectively. In the 99th sample the actual data is 39.84 and the projected model reached the predicted value is 41.82 and Bias (%) as 4.97% and the LSTM model the Predicted Value as 46.23 and Bias value as 16.04% and the CNN model the predicted value as 39.67 and Bias as 0.44 respectively. In the 123th sample the actual data is 55.07 and the proposed model reached the predicted value is 54.63 and Bias (%) as 0.80% and the LSTM model the Predicted Value as 54.76 and Bias (%) value as 0.56% and the CNN model the predicted value as 52.39 and Bias as 4.86% respectively. In the 138th sample the actual data is 55.85 and the projected model reached the predicted value is 55.83 and Bias (%) as 0.03% and the LSTM model the Foretold Value as 54.70 and Bias value as 2.05% and the CNN model the predicted value as 59.80 and Bias as 7.07 respectively. In the 174th sample the actual data is 77.12 and the projected model reached the foretold value is 80.03 and Bias as 4.12% and the LSTM model the Foretold Value as 76.06 and Bias (%) value as 1.37% and the CNN model the foretold value as 80.09 and Bias as 4.90 respectively. In the 175th sample the actual data is 74.56 and the proposed model reached the predicted value is 76.30 and Bias (%) as 2.33% and the LSTM model the Foretold Value as 76.46 and Bias (%) value as 2.55% and the CNN model the foretold value as 87.37 and Bias as 7.19 respectively. In the 244th sample the actual data is 30.26 and the proposed model reached the foretold value is 30.07 and Bias (%) as 0.63% and the LSTM model the Predicted Value as 28.99 and Bias value as 4.20% and the CNN perfect the foretold value as 31.88 and Bias as 5.35 respectively. In the 261th sample the actual data is 28.64 and the proposed model reached the foretold value is 31.05 and Bias (%) as 8.41% and the LSTM model the Foretold Value as 31.28 and Bias value as 9.22% and the CNN model the foretold value as 33.64 and Bias as 17.47 respectively. In the 281th sample the actual data is 13.77 and the proposed model reached the foretold value is 14.21 and Bias (%) as 3.20% and the LSTM perfect the Predicted Value as 15.65 and Bias (%) value as 13.65% and the CNN model the foretold value as 13.85 and Bias as 0.58 respectively.

### 5.3 Performances Metrics

Based on the aforementioned criteria, the following measures of effectiveness are analysed.

**Accuracy:** It is hypothesised what fraction of the test dataset contains of connection records with a predicted ratio. Accuracy, as stated, is a suitable measure for use on an experimental dataset with evenly distributed classes.

**Precision:** It takes as input the total number of attachment logs and makes an educated judgement as to what percentage of those logs were successfully detected. Higher precision (Precision [0,1]) in an ML model is preferable.

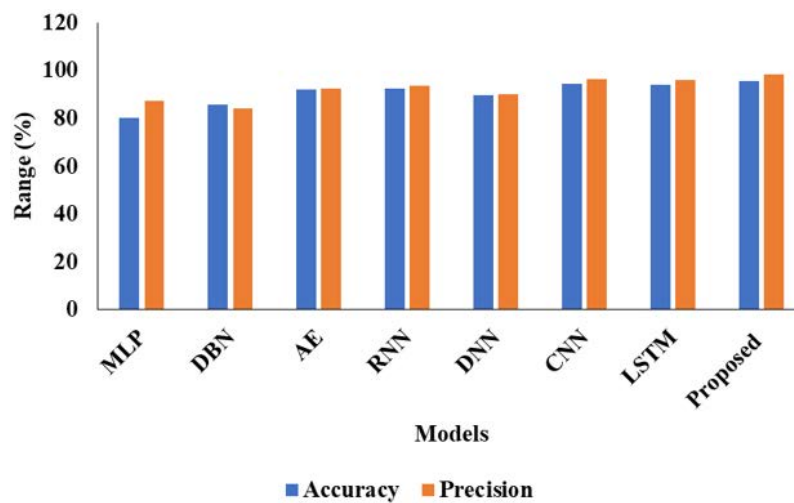
**F1-Score:** The harmonic mean is accuracy and memory. A higher F1-score (F1-score for [0,1]) is preferable.

**False Positive Rate (FPR):** Traffic is determined by dividing the sum of regular connecting records by the total sum of standard connection records. The ML model is enhanced with a reduced FPR (FPR [0,1]).

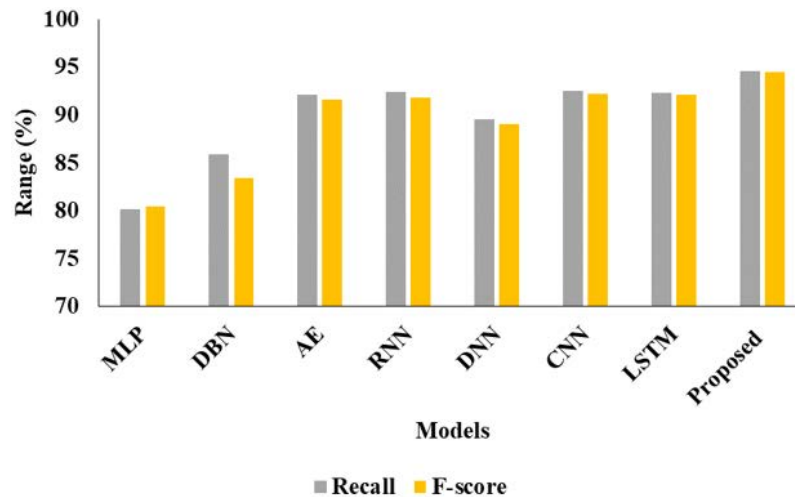
**Table 3.** Prediction analysis of various DL classifiers

Algorithm	Precision	Recall	F-score	Accuracy
LSTM	96.17	92.32	92.10	94.16
Proposed model	98.32	94.62	94.53	95.62
MLP	87.21	80.15	80.43	80.10
DBN	84.32	85.93	83.45	85.71
AE	92.43	92.15	91.68	92.10
RNN	93.48	92.44	91.81	92.46
DNN	90.21	89.54	89.03	89.52
CNN	96.61	92.52	92.24	94.53

In the above Table 3 signify that the prediction analysis of various DL Classifiers. In this analysis we have used different classifiers. In the initial analysis of MLP model reached the accuracy of 80.10 and the precision value as 87.21 and the recall value of 80.15 and finally, F-score value as 80.43 respectively. DBN model reached the accuracy of 85.71 and the precision value as 84.32 and the recall value of 85.93 and finally, F-score value as 83.45 respectively. AE model reached the accuracy of 92.10 and the precision value as 92.43 and the recall value of 92.15 and finally, F-score value as 91.68 respectively. In the next model of RNN model reached the accuracy of 92.46 and the precision value as 93.48 and the recall value of 92.44 and finally, F-score value as 91.81 respectively. In the next model of DNN model reached the accuracy of 89.52 and the precision value as 90.21 and the recall value of 89.54 and finally, F-score value as 89.03 respectively. In the next model of CNN model reached the accuracy of 94.53 and the precision value as 96.61 and the recall value of 92.52 and finally, F-score value as 92.24 respectively. In the next model of LSTM model reached the accuracy of 94.16 and the precision value as 96.17 and the recall value of 92.32 and finally, F-score value as 92.10 correspondingly. In the next suggested model of Proposed model reached the accuracy of 95.62 and the precision value as 98.32 and the recall value of 94.62 and finally, F-score value as 94.53 correspondingly. In this comparisons investigation the projected model reached the better consequences than other compared models (See Figure 4 and Figure 5).



**Figure 4.** Validation comparison of various models



**Figure 5.** Graphical illustration of proposed DL model

## 6 Conclusions

Using the traffic flow data from the cross section, this research first establishes a mathematical model of the road segments, and then uses this model to derive the traffic flow information for the road segments. After that, we put the enhanced TTCNN to use to complete the traffic flow prediction process. The TTCNN architecture relies on an EL to analyze texture characteristics, extract the broad shape information, cap the output vector's size, and fine-tune the model's receptiveness to new input. TTCNN is sensitive to the starting weight, thus the HFFO is built to optimize the TTCNN's structural parameters using a combination of a parallel search method and a group cooperation technique. The experimental consequences validate the superiority of our projected HFFO over the HFFO -TTCNN hybrid perfect improves prediction accuracy and convergence speed by using high-quality structural parameters found by the proposed HFFO for the TTCNN. In the next suggested model of Proposed model reached the accuracy of 95.62 and the precision value as 98.32 and the recall value of 94.62 and finally, F-score value as 94.53 correspondingly. In this comparisons investigation the projected model reached the better consequences than other compared models. The geographical aspect of traffic flow is ignored in favour of its temporal counterpart for the purposes of this article. The prediction performance may be greatly enhanced by including the spatio-temporal characteristic into the input matrix in future research.

## Data Availability

The data supporting our research results are included within the article or supplementary material.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

- [1] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "A deep-learning model for urban traffic flow prediction with traffic events mined from twitter," *World Wide Web*, vol. 24, no. 4, pp. 1345–1368, 2021. <https://doi.org/10.1007/s11280-020-00800-3>
- [2] Y. Li, S. Chai, Z. Ma, and G. Wang, "A hybrid deep learning framework for long-term traffic flow prediction," *IEEE Access*, vol. 9, pp. 11 264–11 271, 2021. <https://doi.org/10.1109/ACCESS.2021.3050836>
- [3] X. Chen, H. Chen, Y. Yang, H. Wu, W. Zhang, J. Zhao, and Y. Xiong, "Traffic flow prediction by an ensemble framework with data denoising and deep learning model," *Physica A*, vol. 565, p. 125574, 2021. <https://doi.org/10.1016/j.physa.2020.125574>
- [4] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Comput. Appl.*, vol. 33, pp. 613–624, 2021. <https://doi.org/10.1007/s00521-020-05002-6>
- [5] A. Navarro-Espinoza, O. López-Bonilla, E. García-Guerrero, E. Tlelo-Cuautle, D. López-Mancilla, C. Hernández-Mejía, and E. Inzunza-González, "Traffic flow prediction for smart traffic lights using machine learning algorithms," *Technologies*, vol. 10, no. 1, p. 5, 2022. <https://doi.org/10.3390/technologies10010005>

- [6] K. Wang, C. Ma, Y. Qiao, X. Lu, W. Hao, and S. Dong, "A hybrid deep learning model with 1DCNN-LSTM-attention networks for short-term traffic flow prediction," *Physica A*, vol. 583, p. 126293, 2021. <https://doi.org/10.1016/j.physa.2021.126293>
- [7] B. Vijayalakshmi, K. Ramar, N. Jhanjhi, S. Verma, M. Kaliappan, K. Vijayalakshmi, S. Vimal, Kavita, and U. Ghosh, "An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city," *Int. J. Commun. Syst.*, vol. 34, no. 3, p. e4609, 2021. <https://doi.org/10.1002/dac.4609>
- [8] W. Fang, W. Zhuo, J. Yan, Y. Song, D. Jiang, and T. Zhou, "Attention meets long short-term memory: A deep learning network for traffic flow forecasting," *Physica A*, vol. 587, p. 126485, 2022. <https://doi.org/10.1016/j.physa.2021.126485>
- [9] Y. Hou, Z. Deng, and H. Cui, "Short-term traffic flow prediction with weather conditions: Based on deep learning algorithms and data fusion," *Complexity*, vol. 2021, p. 6662959, 2021. <https://doi.org/10.1155/2021/6662959>
- [10] Z. Cheng, J. Lu, H. Zhou, Y. Zhang, and L. Zhang, "Short-term traffic flow prediction: An integrated method of econometrics and hybrid deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5231–5244, 2021. <https://doi.org/10.1109/TITS.2021.3052796>
- [11] H. Zeng, Z. Peng, X. Huang, Y. Yang, and R. Hu, "Deep spatio-temporal neural network based on interactive attention for traffic flow prediction," *Appl. Intell.*, vol. 52, pp. 10 285–10 296, 2022. <https://doi.org/10.1007/s10489-021-02879-1>
- [12] T. Zhang, W. Ding, T. Chen, Z. Wang, and J. Chen, "A graph convolutional method for traffic flow prediction in highway network," *Wireless Commun. Mobile Comput.*, p. 1997212, 2021. <https://doi.org/10.1155/2021/1997212>
- [13] W. Li, X. Wang, Y. Zhang, and Q. Wu, "Traffic flow prediction over multi-sensor data correlation with graph convolution network," *Neurocomputing*, vol. 427, pp. 50–63, 2021. <https://doi.org/10.1016/j.neucom.2020.11.032>
- [14] Wikimedia Commons, "Sectors of US economy as percent of GDP 1947-2009," 2021. [https://commons.wikimedia.org/wiki/File:Sectors\\_of\\_US\\_Economy\\_as\\_Percent\\_of\\_GDP\\_1947-2009.png](https://commons.wikimedia.org/wiki/File:Sectors_of_US_Economy_as_Percent_of_GDP_1947-2009.png)
- [15] Y. Bao, J. Huang, Q. Shen, Y. Cao, W. Ding, Z. Shi, and Q. Shi, "Spatial-temporal complex graph convolution network for traffic flow prediction," *Eng. Appl. Artif. Intell.*, vol. 121, p. 106044, 2023. <https://doi.org/10.1016/j.engappai.2023.106044>
- [16] Y. Djenouri, A. Belhadi, G. Srivastava, and J. Lin, "Hybrid graph convolution neural network and branch-and-bound optimization for traffic flow forecasting," *Future Gener. Comput. Syst.*, vol. 139, pp. 100–108, 2023. <https://doi.org/10.1016/j.future.2022.09.018>
- [17] H. Li, Z. Lv, J. Li, Z. Xu, Y. Wang, H. Sun, and Z. Sheng, "Traffic flow forecasting in the COVID-19: A deep spatial-temporal model based on discrete wavelet transformation," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 5, pp. 1–28, 2023. <https://doi.org/10.1145/3564753>
- [18] T. Jia and C. Cai, "Forecasting citywide short-term turning traffic flow at intersections using an attention-based spatiotemporal deep learning model," *Transportmetrica B*, vol. 11, no. 1, pp. 683–705, 2023. <https://doi.org/10.1080/21680566.2022.2116125>
- [19] J. Escorcia-Gutierrez, M. Torres-Torres, N. Madera, and C. Soto, "Henry gas solubility optimization with deep learning enabled traffic flow forecasting in 6G enabled vehicular networks," *AI-Enabled 6G Netw. Appl.*, pp. 35–53, 2023. <https://doi.org/10.1002/9781119812722.ch3>
- [20] K. Cai, Z. Shen, X. Luo, and Y. Li, "Temporal attention aware dual-graph convolution network for air traffic flow prediction," *J. Air Transp. Manage.*, vol. 106, p. 102301, 2023. <https://doi.org/10.1016/j.jairtraman.2022.102301>
- [21] Statista, "Value added to gross domestic product across economic sectors in the United States from 2000 to 2021," 2021. <https://www.statista.com/statistics/270001/distribution-of-gross-domestic-product-gdp-across-economic-sectors-in-the-us/>
- [22] Eurostat, "Shedding light on energy in the EU - A guided tour of energy statistics - 2021 edition," 2019. <https://ec.europa.eu/eurostat/web/products-interactive-publications/-/ks-09-21-149>
- [23] C. Ma, G. Dai, and J. Zhou, "Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM-BiLSTM method," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 5615–5624, 2021. <https://doi.org/10.1109/TITS.2021.3055258>
- [24] H. Wang, X. Zhou, H. Sun, X. Yu, J. Zhao, H. Zhang, and L. Cui, "Firefly algorithm with adaptive control parameters," *Soft Comput.*, vol. 21, no. 18, pp. 5091–5102, 2017. <https://doi.org/10.1007/s00500-016-2104-3>
- [25] T. Bezdan, C. Stoean, A. A. Naamany, N. Bacanin, T. A. Rashid, M. Zivkovic, and K. Venkatachalam, "Hybrid fruit-fly optimization algorithm with k-means for text document clustering," *Mathematics*, vol. 9, no. 16, p.

1929, 2021. <https://doi.org/10.3390/math9161929>

- [26] Q. Zhang, C. Li, C. Yin, H. Zhang, and F. Su, "A hybrid framework model based on wavelet neural network with improved fruit fly optimization algorithm for traffic flow prediction," *Symmetry*, vol. 14, no. 7, p. 1333, 2022. <https://doi.org/10.3390/sym14071333>