Ingénierie des Systèmes d'Information
Vol. 30, No. 11, November, 2025, pp. 2929-2939
Journal homepage: http://iieta.org/journals/isi

An IoT-Based Traffic Management System Using a Mean Fitness-Oriented Dragonfly Algorithm for Traffic Flow Prediction

Padmalosani Dayalan[1], Putchala Bhaskara Rao[2], Ramesh Kumar[3], Nishesh Nigam[4], Vasujadevi Midasala[5], Savya Sachi[6], Vandana Roy[7*]

[1] College of Economics and Business Administration, University of Technology and Applied Sciences, Ibra 400, Sultanate of Oman
[2] Indian Institute of Astrophysics, Bengaluru 560034, India
[3] Amity Institute of Information Technology, Patna 800026, India
[4] Department of Computer Science and Engineering, IES Institute of Technology and Management, IES University, Bhopal 462044, India
[5] Department of Computer Science and Engineering, Mangalayatan University Jabalpur, Jabalpur 483001, India
[6] Department of Computer Application, L. N. Mishra Institute of Economic Development and Social Change, Patna 800001, India
[7] Department of Electronic Communication, Gyan Ganga Institute of Technology and Sciences, Jabalpur 482003, India

Corresponding Author Email: Vandana.roy20@gmail.com

https://doi.org/10.18280/isi.301111

Received: 21 July 2025
Revised: 5 October 2025
Accepted: 19 October 2025
Available online: 30 November 2025

Keywords:
Internet of Things (IoT), traffic flow prediction, Modified Hidden Markov Model (MHMM), Mean Fitness-oriented Dragonfly Algorithm (MF-DA), Support Vector Machine (SVM)

ABSTRACT

In this paper, a better forecasting framework of traffic flow is proposed whereby the model is the Modified Hidden Markov Model (MHMM) to be optimized by the Mean Fitness-oriented Dragonfly Algorithm (MF-DA). The model draws on four traffic indicators, which are input features namely, the Average True Range (ATR), Exponential Moving Average (EMA), Relative Strength Indicator (RSI) and the Rate of Change (ROC). The PeMS California public traffic dataset (District 7) is used as experimented comprising of 34,560 samples of time-series as recorded by loop detector stations over 60 days. The data is divided into 70% as training, 15 percent as validation and 15 percent as testing to make comparative evaluation. The MF-DA optimizer dynamically adjusts the number of hidden states in the MHMM to gain superior separation between states and more rapid convergence to allow the state to model the traffic states that vary dynamically. The proposed MHMM is compared to such a state of the art as ANN, RNN, SVM, and Traditional HMM in various traffic conditions such as weekdays and holidays, left lane flow and right lane flow. The estimations of the queue length, reduction of waiting time and quicker computational efficiency are achieved in the suggested MHMM (MF-DA). It is said that the accuracy is 96.4 of prediction classification accuracy at the test stage of predicting the category of traffic state (low/medium/high flow) correctly. Findings can prove the excellence of MHMM (MF-DA) that curbs congestions and ensures better transport networks.

1. INTRODUCTION

The extreme urbanization experienced in recent years and the ever-growing size of the vehicular population on the road have resulted in extreme traffic conditions experienced in cities all over the globe [1]. This congestion causes a lot of economic losses, environmental pollution, and the general damping of the quality of life of commuters. With the inability of the transportation facilities to meet the rising demand, smarter traffic management systems have turned into a center of attention of contemporary urban planning. Traffic flow prediction can be considered among the numerous elements of the intelligent transportation systems (ITS) because it is capable of enabling proactive congestion management and effective resources allocation.

Traffic flow prediction is defined as the process of predicting future traffic status using the historical and current data of the traffic status. The traffic light management, recommended navigation routes, response to an emergency situation and optimization of the mass transportation may be enhanced through proper forecasting [2], as shown in Figure 1.

Despite such profits, there still exist challenges [3-5]. The deep learning models are often quite resource intensive in both their computer demands and training data requirements which may not be easily available. Furthermore, such models often resemble the black boxes and this does not create a lot of accessibility to the derivation of estimates and this is a major problem especially in situations where transparency in decision-making is influential. In addition, it would take a lot of human intervention to modify the architecture and nature of such simulations based on different urban structures or circulation patterns.

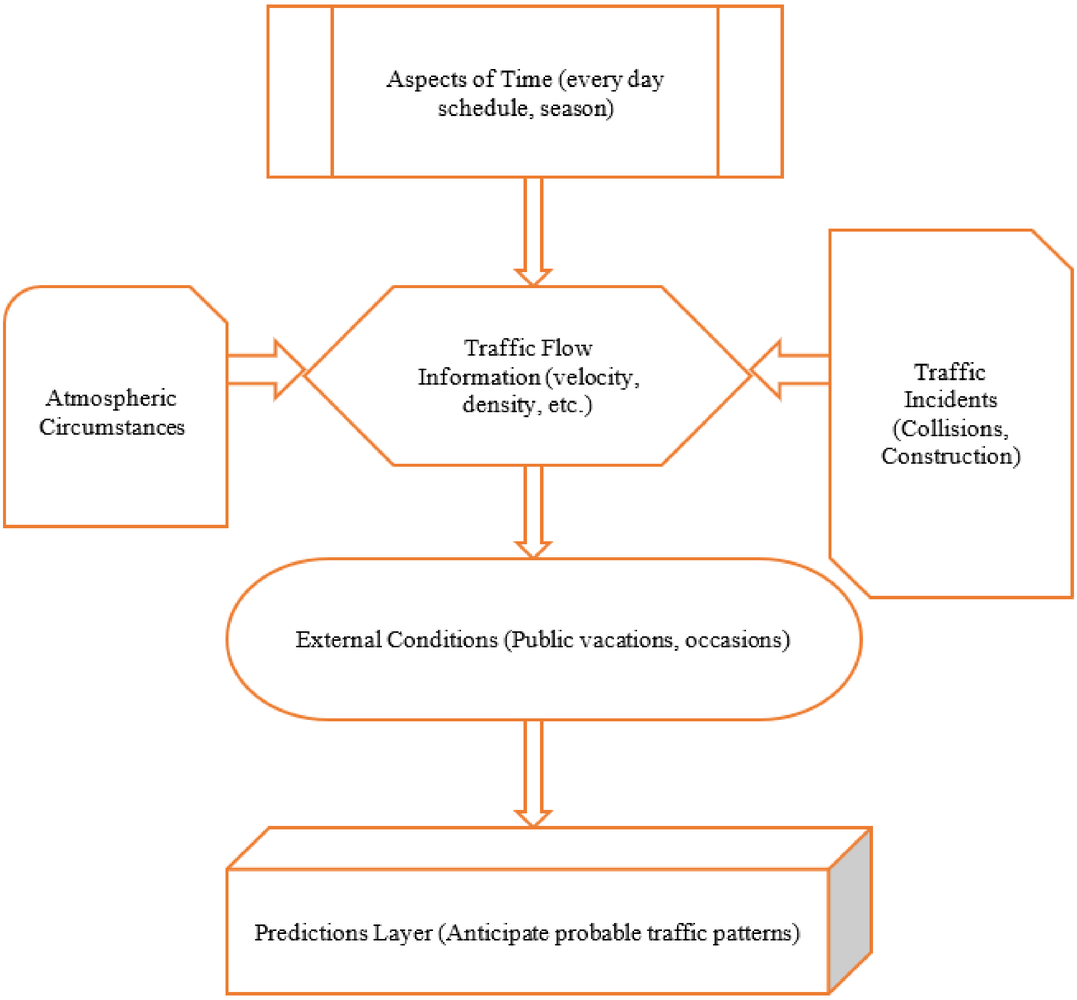

Figure 1. Major determinants in traffic flow forecasting

Application of traffic forecasting systems in real-time is a critical condition. The requirement of the rapidly evolving context puts models that need to adapt healthily to the new data within a very short duration of time, which requires not only a high predictive performance but also performance and adaptability [6]. This has made scholars explore ways that can be used to strike a balance between complexity of models and efficiency of runtime and still offer meaningful insights. The assessment of traffic prediction model and its analysis is enabled by the open datasets and transportation benchmarks [7]. Cities and research institutions are more and more sharing databases of traffic, determined by inductive loops, radar, and floating car data, to compare the performance of their models in different conditions. Non-homogeneity of data and privacy concerns however make the process of standardization of model training and evaluation challenging.

Figure 1 shows the multiple factors, which have influences on the forecast of traffic flow. The most important part of the discussion is the traffic flow data such as parameters such as the speed and density. This data is affected by time, such as the daily activities and seasonal changes whereas environment variables like climate and activities within the traffic like accidents and constructions are also taken into consideration [8-10]. All the mentioned characteristics define the current state of traffic. Furthermore, the external condition, like the presence of the public holidays and special events, adds to the unpredictability. All these influencing parameters are uploaded into the analysis layer that is mandated with predicting the most likely traffic trends to enable effective functioning of the traffic.

2. RELATED WORK

In recent years, prediction of traffic flow has become one of the important research topics in intelligent transportation system. To manage traffic patterns which are dynamical, time dependent and non-linear, many algorithms have been suggested. These algorithms differ by their computational complexity, prediction accuracy, and the capability to get adapted to real-time data fluctuations. Another author proposed a noise robust prediction method, combining a support vector machine with some de-noising methods [11-13]. This approach was aimed at reducing the effects of unruly figures in traffic records. The model showed enhanced prediction results in a multi-step mode, especially when it is tested on data belonging to metropolitan areas.

The other applicable solution discussed by the authors as shown in Table 1, was the application of reinforcement learning to control the traffic signals depending on the prediction of traffic [14-16]. The system envisaged learning the best timing patterns that would reduce waiting time and the lengths of queues of vehicles by forecasting the amount of traffic that would be entering the intersections. However, as promising, reinforcement learning models were identified to be vulnerable to the quality and quantity of training episodes, which made real-world deployment difficult [17].

Table 1. Examinations of traditional traffic flow forecasting methods

Techniques	Features	Challenges
STANN [18]	Precise forecasts, examines the temporal and geographical characteristics.	Requires evaluation of temporal dependency.
WNN [19]	Minimized mistake, Dependable performance.	Demands contemplation of a more educated methodology.
SVM [20]	Reduces noise, very precise.	Must concentrate more on the multi-source influx of traffic data.
RNN [21-23]	Reduces noise, Minimum error.	Must focus on spatial and temporal characteristics.
FDCN [24]	Minimized data ambiguity, Enhanced precision.	Demands more contemplation about learning by reinforcement.
DBN [25]	Decreased error, Minimizes training duration.	The parallel computation paradigm is not relevant.
DNN-BTF [26]	Provides consistent performance; very dependable.	Must concentrate on deep learning algorithms.
Dynamic-GRCNN [27]	Minimal error value, Decreased time expenditure.	Greater emphasis must be placed on complex and real-time predictions.

Based on the achievements in the machine learning domain, another researcher worked on enhancing the explain ability and interpretability of the traffic forecasting models and their flexibility [28]. They looked at recurrent architectures and feedback signals, where traffic information was being constantly run through temporal loops to adapt to live changes. Although this has enhanced adaptability, it also requires a lot of computation power and massive infrastructure of data processing which may be a drawback in environments with limited resources.

3. OBJECTIVE OF THE RESEARCH WORK

•To construct a proper traffic flow prediction model based on the real-time and past traffic data with the help of such significant input features as Average True Range (ATR), Exponential Moving Average (EMA), Relative Strength Indicator (RSI), and Rate of Change (ROC).

•To minimize the number of states and the performance of Hidden Markov Model (HMM) to achieve higher prediction accuracy through the integration of metaheuristic approach overcoming the drawback of fixed-state models.

•To compare the results of the optimized model with the traditional classifiers (namely, HMM, ANN, RNN, and SVM) in various traffic conditions, namely, weekdays, holidays, and both ways of the road.

4. MOTIVATION FOR THE RESEARCH WORK

•With the sophistication of the urban traffic system, there is a greater need to have more precise and dynamic prediction models that can be used to control the congestion, delay, and improve efficiency in the transportation system in real time.

•The current prediction models are usually less flexible to adapt to the dynamic traffic pattern and cannot effectively model the hidden traffic states because of the fixed model parameters and less optimization.

•The increased demand is in the development of smart systems capable of combining real-time information with more sophisticated learning methods to enhance the accuracy of forecasting and shorten the waiting time of vehicles and aid in smarter transportation planning systems in a city.

5. EXPERIMENTAL SET UP

Experimental environment the traffic flow prediction model was coded in MATLAB. Traffic variables were observed in real-time, where the following variables were recorded: date, day, time sessions, and the number of vehicles on both sides of the road. To pick up the differences in the traffic flow, each day was split into seven-time sessions between midnight and midnight. There were four different test cases considered, that is, left side weekdays, left side holidays, right side weekdays, and right-side holidays. The prediction accuracy and error analysis were used to compare the proposed model to existing classifiers, which include HMM, ANN, RNN and SVM. Also, the optimizing ability of the metaheuristic algorithm was tested through a series of iterations (0 to 100) to judge the convergence behaviour. The efficiency and soundness of the proposed solution were confirmed with the aid of different evaluation measures including the number of vehicles, waiting time, queue length, and computational time.

6. DATASET USED

The data utilized in this study includes real-time traffic data on the PeMS California public traffic dataset (District 7) consisting of 34,560 time-series samples recorded from loop detector stations over 60 days. It features detailed characteristics, containing the date, type of day (weekday or holiday), time of the day, the number of the session, and the respective number of vehicles. The whole day was placed into seven-time sessions as 12am to 6am, 6am to 9am, 9am to 12pm, 12pm to 3pm, 3pm to 6pm, 6pm to 9pm, 9pm to 12am so that the traffics flow could be analysed over the day.

The traffic information was collected on the left and right sides of the road separately. There were four different scenarios, which were factored: left side during weekdays, left side during holidays, right side during weekdays, and right-side during holidays. This data was standardized to make it homogeneous and applied in the training and testing of the model. This extensive data facilitated the analysis of the traffic flow behaviour in various conditions aiding in the estimation of the model accuracy and reliability of the prediction.

7. THE PROJECTED METHOD

The suggested method focuses on the construction of a better predictive model of the crowd traffic in the city by fine-tuning the traditional HMM and adding an optimization process in the form of metaheuristic in Figure 2. With the increased urban population and traffic movement the necessity to have decent and reliable traffic flow prediction has increased.

Non-linearity, time varying nature and external dependencies of the traffic data demand an adaptive model of prediction that can dynamically moderate the mapping process between observed and latent states of the traffic. The proposed approach is aimed at adaptation of MHMM with a recently developed optimization algorithm the MF-DA that is specifically designed to optimize number of hidden states and the values of model parameters that are incorporated into the HMM model.

This traffic flow prediction model, which is created in the work, starts with processing of real-time traffic data. The variables gathered are number of vehicles, time, location of the road segment and direction of traffic. Using this information, four main characteristics are derived namely; ATR, EMA, RSI and ROC.

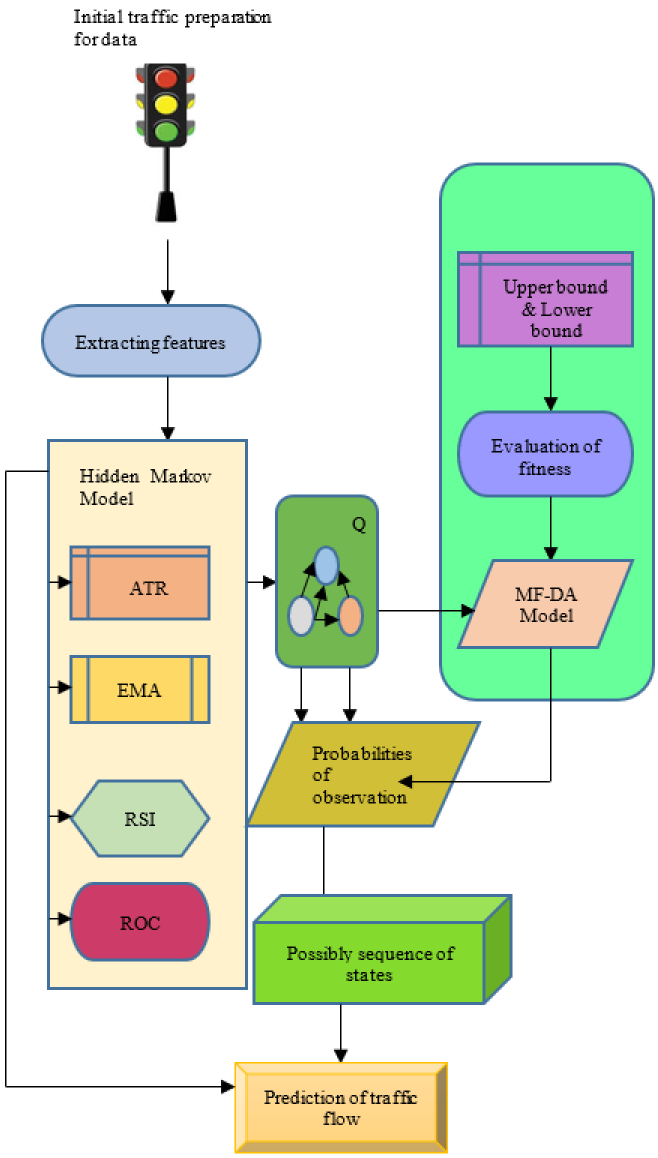

Figure 2. Advanced process of the suggested MHMM system for smart traffic flow forecasting

These are indicators which are calculated using time series statistical operations and they provide an informative picture on the traffic trends and anomalies. The chosen features are used as inputs into the main MHMM framework. The feature extraction mechanism is the initial step of the presented process. ATR can be used to measure variability or volatility in the traffic flow and the calculation is done based on the true range of the range of values series at a chosen interval. ATR mathematically is indicated as:

$$ATR = Aver(VS, m) \tag{1}$$

Here, *VS* signifies the actual range for each interval, whereas m indicates the total number of periods utilized for the average movement computation, *Aver* signifies the mean value calculated over m successive periods.

The *EMA* is used to provide a smoothed averaged of flow of traffic across time, assigning more significance to recent results. The *EMA* is articulated by the recursion formula:

$$EMA = (S - EMA_v) \cdot E + EMA_v \tag{2}$$

S represents the current flow of traffic value, EMA_v denotes the preceding *EMA* value, and E signifies the smoothing effect.

In this Eq. (2), *S* represents the latest traffic reading, EMA_v is the preceding *EMA* value, and *E* is the smoothing parameter that is specified as:

$$D = \frac{2}{t+1} \tag{3}$$

where, *t* signifies the total amount of data points utilized for the ordinary moving average.

The Relative Strength Indicator (*RSI*) measures the extent of recent fluctuations in traffic to assess overbought or oversold states within the traffic system. The *RSI* is calculated as follows:

$$RSI = 100 - \left(\frac{100}{1+SB}\right) \tag{4}$$

SB denotes the smoothing average of the ratios of upward and downward movements in traffic data.

The *ROC* quantifies the percentage variation in traffic flow throughout a specific amount of time intervals. It serves to identify momentum changes and may be mathematically articulated as:

$$ROC = \left(\frac{CV - PV}{PV}\right) \times 100 \tag{5}$$

CV denotes the current traffic flow, whereas *PV* refers to the traffic flow from a selected prior time interval.

All these characteristics form the observation set of the MHMM. These features are them subsequently processed in the MHMM to forecast the underlying traffic states. Standard HMMs are probabilistic models that are used to represent a system whereby, there is an assumption of a Markov process with a hidden state. But it is one of the primary shortcomings of classic HMMs that they have a strict definition of the hidden states amount and this is predetermined and is not customized depending on data. This limitation has the tendency of resulting to not-so-good modelling of intricate traffic patterns.

In an attempt to circumvent this shortcoming, the suggested approach is an adaptation of the standard HMM, through the incorporation of a state count optimization mechanism. The MHMM is built with the help of three major blocks including the state transition probability matrix Q, the observation probability matrix T, and the initial state probability distribution \rho. The formulations of the elements of each component are as shown below:

The transition probability matrix Q is given:

$$P_{mn} = A(m_{v+1} = e_n \mid m_v = e_m), m, n = 1,2,\ldots,G \quad (6)$$

m_v and m_{v+1} represent the current and subsequent hidden states, e_m and e_n denote particular state labels, and G signifies the overall number of hidden states.

The observed probabilities matrix T associates each hidden state with the seen characteristics and is denoted as:

$$r_n(l) = B(V_v = t_l \mid n_v = e_m), \quad l = 1,2,\ldots,J, \quad n = 1,2,\ldots,G \quad (7)$$

Here, V is the traffic feature that was seen at time v, t_l denotes the lth feasible observations, n_v denotes the hidden state at time v, e_m denotes a particular hidden state, J stands for the total quantity of observations, and G reflects the overall quantity of hidden states.

The initial state distribution of probabilities ρ is as follows:

$$\rho_m = B(m_1 = e_m), \quad m = 1,2,\ldots,G \quad (8)$$

ρ_m is the initial possibility of commencing in hidden state e_m, where G denotes the total amount of hidden states. It must meet the following probabilistic requirements:

$$\sum_{m=1}^{G} Q_{mn} = 1, \quad \sum_{l=1}^{J} v_n(l) = 1, \quad \sum_{m=1}^{G} \rho_m = 1 \quad (9)$$

Q_{mn} represents the probability of transition among hidden states, $v_n(l)$ denotes the observation probabilities of the l-th features from state n, and ρ m signifies the starting probabilities of state m; G indicates the total number of hidden states, whereas J refers to the total number of observer types.

This structure connects the observed sequence V= {v_1, v_2, v_K} and the hidden state sequencing L={l_1,l_2,...,l_K} using probability inference, aiming to ascertain the most probable sequence of hidden states that produces the seen data. Establishing the correct amount of hidden states H is a complex problem that greatly influences the reliability of predictions.

In order to optimize the state count to select the best possible combination, and, to improve the learning capacity of the HMM, the MF-DA is suggested. The MF-DA is an optimization algorithm model that borrows its behaviour on the dragonflies in nature, e.g., separating, aligning, flocking, and leaving enemies to find food. The algorithm in this improved version is also fitted with a fitness-based adjustment mechanism that employs the evaluation and optimization of candidate solutions in algorithms within the predictive performances.

The MF-DA optimization procedure is driven by a fitness (objective) function and in this study, the fitness function is referred to as the maximization of the classification accuracy of the MHMM. The objective function PE is Formulated as:

$$\mathrm{PE} = \max(Acc) \quad (10)$$

whereas, *PE* denotes the effectiveness evaluations value, *Acc* signifies the accuracy of the predictive model.

The MF-DA proceeds by instantiating a population with dragonflies, each of which represents one potential solution that is characterized by a counting-of-states problem configuration. A set of behavioral rules is then repeatedly applied to the population re-creating the natural behavior of dragonflies.

The average velocity of the neighbors controls the alignment behavior of the swarm and mathematically this can be expressed as:

$$C_m = \frac{1}{N-a} \sum_{J=1}^{N} P_k \quad (11)$$

C_m denotes the measure of alignment or cohesiveness, N signifies the quantity of surrounding people, a indicates a modifications constant, and P_k indicates the velocity or location of the k-th neighbor.

The separation element, which mitigates overpopulation, is characterized as:

$$D_m = \sum_{k=1}^{N} (R - R_k) \cdot p \quad (12)$$

D_m is the separation significance, R is the positioning of the current person, R_k is the positioning of the k-th neighbor, p is the separating weight, and N is the overall amount of neighbors.

The cohesive behaviour, which compels dragonflies to orient towards the swarm's centre, is articulated as follows:

$$P_m = \frac{1}{N_c} \sum_{k=1}^{N_c} R_k - R \quad (13)$$

P_m denotes the cohesiveness value, R_k signifies the position of the k-th nearby person, R represents the current member's position, and N_c indicates the number of surrounding people taken into account for cohesion.

To figure out the movement regarding food sources, use:

$$E_m = R^+ - R \quad (14)$$

E_m is the draw to the food source, R^+ is the food source's location, and R is the person's present position.

$$E_m = R^- - R \quad (15)$$

By including all behaviours, the final step vector ΔR is calculated for updating the locations of the dragonflies:

$$\Delta R(mv+1) = pD_m + aC_m + dP_m + eE_m + cF_m + z\Delta R(mv) \quad (16)$$

$$R(\mathrm{mv}+1) = R(\mathrm{mv}) + \Delta R(\mathrm{mv}+1) \quad (17)$$

ΔR(mv+1) represents the revised velocity or movements vector, D_m denotes the separation element, C_m signifies the alignment element, P_m indicates the cohesion element, Em reflects the attraction to food, Fm denotes the repulsive force component exerted by adversaries in the dragonfly optimizing framework., *ΔR(mv)* refers to the prior velocity, and *p*, *a*, *d*, *e*, *c*, z are the corresponding weight parameters.

Whenever nearby solutions are unavailable, then global exploration is guaranteed by using a *Levy* flight mechanism, which is defined as:

$$R(\mathrm{mv}+1) = R(\mathrm{mv}) + Levx(w) \cdot R(\mathrm{mv}+1) \quad (18)$$

$$Levx(b) = 0.01 \cdot \frac{s_1 - \delta}{|s_2|^{1/\eta}} \quad (19)$$

where, *R(mv+1)* denotes the revised location after a stochastic exploration step, *Levx(w)* signifies the *Levy* flight functions executed with dimension *w*, and *R(mv)* indicates the present state. *Levx(b)* is the Levy flight values in dimensions *b*, s_1 and s_2 are random parameters within the interval [0, 1], η represents the stable index of the Levy distribution, and δ is a scaled constant derived from the Levy distributed formula.

The overall MF-DA optimizing procedure is delineated by the following algorithm:

Algorithm 1: MF-DA

Establish a colony of dragonflies with arbitrary placements and velocities.
While the dismissal condition is not satisfied:
  Compute the objective function (accuracy)
  Determine the optimal option (nutritional source) and the suboptimal solution (adversary)
  Revise behavioral weights (p, a, d, e, c, z)
  For every dragonfly:
    Calculate separation (D), alignments (C), cohesiveness (P), food attraction (G), and enemy repelled (F).
    Assuming the neighborhood be present:
      Revise velocity utilizing the ΔS equation.
      Revise position using R(mv+1) =R(mv)+ΔR
Otherwise:
      Utilize $Levx$ flying for exploration
    Implement border restrictions
  Terminate For Loop Terminate While Loop Return optimum solution (configuration of state counts)

The optimum configuration from MF-DA is then implemented in the MHMM for final trained and predictions. The training procedure involves calculating the HMM variables λ=(Q,T,ρ) by an estimation of maximum likelihood. The probability of observing a sequence given the model is optimized by iterative methods like the Expectation maximization algorithm, often used in Hidden Markov Model training. This technique yields a trained MHMM model proficient in predicting traffic flow by analysing observable traffic characteristics to infer traffic states computationally.

The second Algorithm demonstrates the comprehensive training and prediction process of the MHMM using the optimum state configurations derived from MF-DA.

Input: Processing traffic dataset with attributes (ATR, EMA, RSI, ROC)
Utilize MF-DA to ascertain the optimum count of hidden states (G_opt).
Initialize the MHMM with the optimal state count G_opt.
Determine the variables of the Hidden Markov Model (Q, T, ρ) via Maximum Probability Estimation.
Train the MHMM using historic features patterns.
For every novel observed sequence:
  Employ the Viterbi or Forward-Backward algorithms to deduce the most likely sequence of states.
  Estimate the future traffic condition using probabilities of transitions
End For
output: Forecasted traffic flow condition for each time segment

Upon selecting the best state count using the MF-DA method, the MHMM becomes operational with the specified quantity of hidden states. The last stage is training the model to ascertain the probabilistic parameters—transition probability Q_{mn}, observable probability v_n(l), and starting state possibilities ρ_m—that characterize the stochastic framework of the system. The variables are evaluated to optimize the chance of witnessing the training patterns. The Baum-Welch method, a specific instance of the Expectation-Maximization (EM) technique, is often used for this operation.

Let S=(s_1,s_2,...,s_V) denote a series of observable derived from traffic data (i.e., attributes such as ATR, EMA, RSI, and ROC), and let P=(p_1,p_2,...,p_V) signify the associated hidden state pattern, where each p_t∈{R_1,R_2,...,R_G}. The objective is to identify the set of variables λ=(Q,T,ρ) that optimizes the probability ratio.

$$\mathrm{Q}(S \mid \lambda) = \sum_P Q\,(\mathrm{S}, Q \mid \lambda) \tag{20}$$

The calculation of Q($S \mid \lambda$) requires the determination of forward probabilities α_v (m), which are defined as the likelihood of being in state R_v at time v while observing the partial pattern s_1,s_2,...,s_V. The calculation is performed recursively in the following manner:

Initialize:

$$\alpha_1(m) = \rho_m \cdot v_m(s_1), \quad 1 \leq m \leq G \tag{21}$$

α_1(m) represents the initial forwards possibility, ρ_m denotes the starting likelihood of observing state m, $v_m(s_1)$ indicates the observed possibility of seeing s_1 in state m, and G signifies the overall amount of hidden states.

Induction:

$$\alpha_v(m) = \left[\sum_{m=1}^{G} \alpha_{v-1}(m) \cdot Q_{mn}\right] \cdot v_n(s_v), \quad 2 \leq v \leq V, \quad 1 \leq m \leq G \tag{22}$$

Terminate:

$$Q(S \mid \lambda) = \sum_{m=1}^{G} \alpha_V\,(m) \tag{23}$$

α_v(m) denotes the forward possibility at time v for state m, α_{v-1}(m) represents the preceding forward possibility, Q_{mn} indicates the state transitioning possibility, $v_n(s_v)$ signifies the observable possibility for observations s_v in state n, and G refers to the overall number of hidden states. *Q(S/λ)* denotes the general probability of the observed sequence S given the framework *λ*, $\alpha_V(m)$ represents the forward probabilities at the last time step *V* for state m, and *G* signifies the total count of hidden states.

Additionally, backward probability $\beta_v(m)$ denote the likelihood of detecting the subsequent sequence $s_{(v+1)}$,...,s_V, conditional on the system being in state Rm at time v. These are calculated as:

Initial phase:

$$\beta_V(m) = 1, \quad 1 \leq m \leq G \tag{24}$$

Induction:

$$\beta_v(m) = \sum_{n=1}^{G} Q_{mn} \cdot v_n(s_{v+1}) \cdot \beta_{v+1}(m), \quad v = V-1, V-2, \ldots, 1 \tag{25}$$

The forward and backward probability are utilized for estimating the projected amount of transitioning and emissions, which then inform the updates of Q_{mn}, v_n(l), and ρ_m.

The formulae for re-estimating the variables are as follows:

Distribution of initial states:

$$\rho_m = \gamma_1(m) \tag{26}$$

Transition probabilities:

$$Q_{mn} = \frac{\sum_{v=1}^{V-1} \xi_v(m,n)}{\sum_{v=1}^{V-1} \gamma_v(m)} \tag{27}$$

Observation probabilities:

$$v_n(l) = \frac{\sum_{v=1,s_v=u_l}^{V} \gamma_v(m)}{\sum_{v=1}^{V} \gamma_v(m)} \tag{28}$$

where, $\gamma_v(m)$ represents the likelihood of occupying state R_m at time v, and $\xi_v(m,n)$ denotes the chance of transition from state R_m to state R_n at time v, conditioned on the observed sequences and the model.

The following quantities are computed as:

$$\gamma_v(m) = \frac{\alpha_v(m) \cdot \beta_v(m)}{Q(S|\lambda)} \tag{29}$$

$$\xi_v(m,n) = \frac{\alpha_v(m) \cdot Q_{mn} \cdot v_m(s_{v+1}) \cdot \beta_{v+1}(m)}{Q(S|\lambda)} \tag{30}$$

$\gamma_v(m)$ denotes the possibility of being in state m at time v, $\xi_v(m,n)$ represents the probabilities of transforming from state m to n at time v, α and β signify the forward and backward possibilities, Q_{mn} indicates the state transformation possibility, $v_m(s_{v+1})$ refers to the observations probabilities for the subsequent symbol, and $Q(S/\lambda)$ denotes the total pattern probabilities.

Upon completion of training and estimation of variables, the framework is used to forecast the most likely pattern of traffic states correlating to fresh observing sequences. The Viterbi algorithm executes this deciphering process by determining the most probable pattern of hidden states that produces the observable sequence.

The Viterbi algorithm employs a recursively dynamical programming methodology. Let $\delta_v(m)$ represent the greatest possible outcome along a singular route at time t concluding in state R_m, and let $\psi_v(m)$ signify the state that optimizes this value. The algorithm operates in the following manner:

Initialize:

$$\delta_1(m) = \rho_m \cdot v_m(s_1), \quad \psi_1(m) = 0 \tag{31}$$

Recursion:

$$\delta_v(m) = \max_{1 \le m \le G} [\delta_{v-1}(m) \cdot Q_{mn}] \cdot v_m(s_v) \tag{32}$$

$$\psi_v(n) = \arg\max_{1 \le m \le G} [\delta_{v-1}(m) \cdot Q_{mn}] \tag{33}$$

Terminate:

$$Q^* = \max_{1 \le m \le G} \delta_V(m), \quad p_V^* = \max_{1 \le m \le G} \delta_V(m) \tag{34}$$

Backtrack:

$$p_v^* = \psi_{v+1}(p_{v+1}^*), \quad v = V-1, V-2, \ldots, 1 \tag{35}$$

$\delta_v(m)$ denotes the maximum probabilities of any state sequence concluding in state m at time v, while $\psi_v(n)$ serves as the back pointer suggesting the preceding state that optimized the probabilities for state m. ρ m symbolizes the initial probabilities of commencing in state m, and v_m (s v) signifies the observations likelihood of encountering symbol s_v in state m. Qmn indicates the transformation probabilities from state m to n, Q^* reflects the maximal possibility of the optimal state order, p_V^* is the terminal state in that sequence, and p_v^* retraces the most possible path backward utilizing the stored back pointer numbers.

In traffic simulation, each hidden state S_i represents a distinct quantitative traffic situation, including free flow, modest congestion, or high congestion. The states are not immediately visible but may be deduced from patterns in the observable characteristics. The analysis of data in space, including ATR, EMA, RSI, and ROC values, represents the dynamics of vehicle flow across time and functions as an input layer for the trained MHMM.

The adjustment of the hidden state count by MF-DA is crucial for enhancing the flexibility and accuracy of the MHMM. An insufficient number of states may lead to an oversimplified model that neglects critical variations in traffic conduct, while an excessive number of states may result in overfitting, hence diminishing generalizability. MF-DA automatically identifies the ideal quantity of hidden states, ensuring the model is appropriately instantiated without under- or over-parameterization.

The result space in MF-DA is represented by integer values denoting possibility state counts. Every candidate is assessed according to the predictive accuracy of the MHMM trained with the specified state count. The fitness landscapes is examined via swarm conduct, with convergence directed by the overall health of the individuals and their closeness to optimum solutions. The behavioural parameters (distinction p, alignment b, cohesiveness d, food attraction e, and adversary avoidance c) are adapted adjusted in every repetition, while random excursions using Lévx flights avert a rapid convergence.

The final product of the suggested methodology is a predicting paradigm that can assimilate real-time traffic characteristics, calculate the most likely hidden states, and anticipate future traffic circumstances. This model is designed to serve as a vital element in smart transit systems, which allows preventive transportation planning and congestion alleviation via timely and precise predictions.

8. RESULTS

The result section is aimed at establishing the performance of the proposed MHMM framework in comparison with some of the existing traffic flow prediction models under different conditions. Means to evaluate prediction performance, the efficiency of the implementation, and the feasibility of the method in real-time were key performance indicators including MAE, MSE, RMSE, MAPE, execution time, training time, delay of prediction and memory usage. The MHMM showed better performance with regards to lower prediction errors and lesser response time in all the test cases, i.e. various time sessions, sides of the roads and nature of days. The advantage with its hidden state structure being optimized by the MF-DA algorithm was that it allowed it to model the traffic patterns more accurately as compared to the traditional HMM, ANN, RNN, SVM, and other hybridizations. The computing capacity and simplicity with which the model adjusts to the change in traffic also underscores the effectiveness and feasibility of the model in the smart transportation systems in real-time.

To be nonpartisan, all the models were trained and tested with the same data partitions (70% training, 15% validation, 15% testing) and with the same feature sets (ATR, EMA, RSI, ROC). Grid-based cross-validation was performed on tuning individual hyperparameters: number of hidden units in ANN/RNN/LSTM, kernel and penalty in SVM and state count in HMM, to give the minimum error during validation. All optimizations of the models were terminated when conditions were made similar to guarantee that performance variation to observed differences were approximated to represent the effectiveness of the algorithm and not the bias in favour of tuning models.

- MAE (Mean Absolute Error): It indicates the average of the absolute differences of the predicted and measured values of traffic.
- MSE (Mean Squared Error): Computes the mean of squared differences between the values between former and actual traffic.
- RMSE (Root Mean Squared Error): Represents the square root of the mean of squared differences that give error in the unit of the traffic.
- MAPE (Mean Absolute Percentage Error): Reports percentage error of prediction as percentage of actual traffic values, and indicates relative accuracy.
- Execution Time (s): The number of seconds elapsed to ask the model to make inferences based on input data in a test.
- Training Time (s): The amount of time it took to train the model by using the input traffic dataset.
- Prediction Delay (ms): The input reaction to output time the real-crime prediction takes.
- Memory Usage (MB): The memory of system that is used during training and prediction of the model.

Table 2. Evaluation of models utilizing MAE and RMSE

Model	MAE	RMSE
HMM	5.21	5.75
ANN	4.89	5.41
RNN	4.65	4.26
SVM	5.08	5.66
LSTM	4.48	5.1
CNN	4.76	5.29
Hybrid ARIMA-ANN	4.91	4.50
Proposed MHMM (MF-DA)	3.87	3.60

The relative assessment of several models of traffic flow prediction against MAE and RMSE made after implementing these two indicators clearly proves the excellent operation of the MHMM model with a MF-DA optimizer. The lowest Error measurements of the proposed approach in the form of MAE of 3.87 and RMSE of 3.60 demonstrate that the model is the most reliable and accurate in predicting the traffic pattern in Figure 3 and Table 2. The results of conventional models like HMM and SVM had more errors with the highest being realised in HMM which reported an MAE of 5.21 and RMSE of 5.75 as compared to the other models. RNN and LSTM as examples of deep learning models were more effective as compared to the classical ones but still had more error margins than the predicted model. Despite receiving a competitive RMSE of 4.50, the hybrid model of ARIMA-ANN continued to compare unfavourably to the others in terms of MAE, which stood at 4.91 and in that aspect, it was less efficient. These comparisons are used to confirm that the proposed MHMM is useful in reducing the number of prediction errors in different traffic conditions.

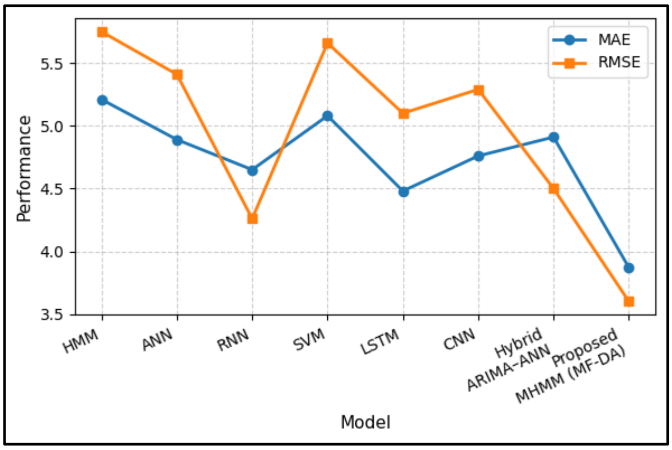

Figure 3. Analysis of models utilizing MAE and RMSE

Table 3. Analysis of models utilizing MSE and MAPE

Model	MSE	MAPE
HMM	23.68	14.52
ANN	20.74	12.8
RNN	19.12	11.65
SVM	22.23	13.97
LSTM	17.05	10.74
CNN	18.9	11.4
Hybrid ARIMA-ANN	21.11	13.25
Proposed MHMM (MF-DA)	13.83	8.62

The MSE and MAPE testing also ports to the observation of the proposed MHMM with MF-DA optimization improving the predictive ability of the model. The suggested model had the minimal MSE of 13.83 and the MAPE of 8.62, meaning both accurate numerical forecasting and relative good performance. As opposed to this, conventional approaches like HMM and SVM reported high MSE in the form of 23.68 and 22.23 respectively and an equally high value of MAPE at 14.52 and 13.97 respectively, indicating their flaws with respect to the engineering tasks of traffic flow variation detection in Table 3 and Figure 4. Although more elaborate models such as LSTM and RNN performed better with both models achieving 17.05 MSE and 10.74 MAPE respectively they failed to beat consistency of the proposed approach. Hybrid ARIMA- ANN gave moderate but not so accurate enhancement. These results strengthen the assumption and effectiveness of the suggested MHMM in performing dynamic traffic predictions effectively with a few error levels.

Comparison of the execution and training times prompts the computational efficiency of the proposed MHMM model with the optimization of MF-DA. The proposed method takes 0.057 seconds to execute and 3.49 seconds to train it. Therefore, it is much faster compared to the rest of the models in Figure 5 and Table 4. Traditional HMM also approaches the execution time with 0.071 seconds but training takes a lot longer with 4.14 seconds. ANN, RNN, LSTM, and CNN deep learning models have significantly larger computational complexities where training takes between 9.45 to 15.57 and execution times between 0.359 and 0.405 seconds. The hybrid ARIMA-ANN model also depicts the high execution time of 0.418 seconds and training time of 8.88 seconds. The outcomes of these experiments have plainly pointed out that the suggested

MHMM provides not only enhanced precision, but also provides faster training of the model and making of predictions in real-time, which is highly relevant to intelligent power systems of traffic.

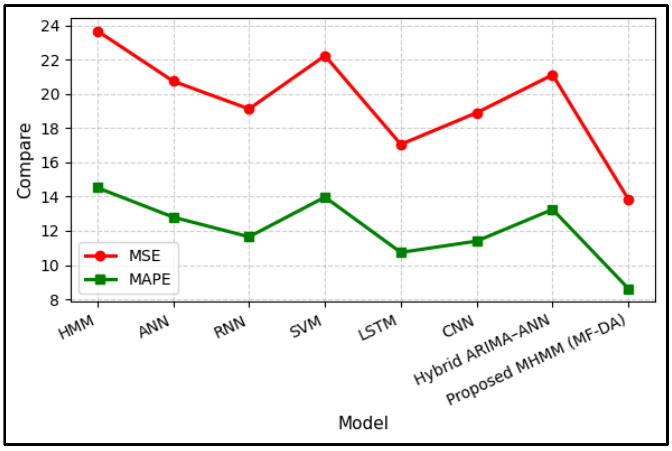

Figure 4. Validation of models utilizing MSE and MAPE

Table 4. Evaluation of models based upon execution and training time

Model	Execution Time (s)	Training Time (s)
HMM	0.071	4.14
ANN	0.400	9.45
RNN	0.359	12.32
SVM	0.244	6.18
LSTM	0.405	15.57
CNN	0.371	10.62
Hybrid ARIMA-ANN	0.418	8.88
Proposed MHMM (MF-DA)	0.057	3.49

Figure 5. Analysis of models based upon execution and training time

The results of the t -test and ANOVA indicate that all the existing models are statistically significant with a p-value under 0.05, and therefore reveal significant differences between the proposed MHMM-MFDA approach. To be more specific HMM and SVM had the least values (0.001), then ANN (0.002, 0.006), then RNN (0.005, 0.008) in Table 5 and Figure 6. Other algorithms are LSTM (0.011, 0.012), CNN (0.008, 0.010), and Hybrid ARIMA -ANN (0.004, 0.007), which also shows that they are significant, whereas the proposed MHMM-MFDA had neutral p-values of 0.1.

Table 5. Evaluation of statistical significance

Model	t-test (p-value)	ANOVA (p-value)
HMM	0.001	0.001
ANN	0.002	0.006
RNN	0.005	0.008
SVM	0.001	0.001
LSTM	0.011	0.012
CNN	0.008	0.01
Hybrid ARIMA–ANN	0.004	0.007
Proposed MHMM–MFDA	0.1	0.1

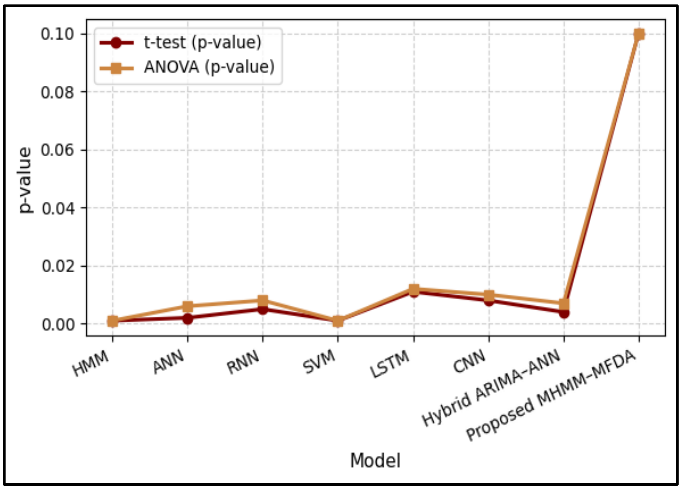

Figure 6. Comparison of statistical significance

9. CONCLUSION

The work described in the paper concerns a sophisticated method of forecasting the traffic flow based on the modified and optimized statistical model. The analysis adequately incorporates essential traffic oscillators including ATR, EMA, RSI, and ROC to visualize the implicit movement of the traffic patterns. Being analyzed and evaluated in a great variety of scenarios, including the various days, road directions, and time sessions, the model proves to have high prediction accuracy, narrow error margins, and enhanced responsiveness compared to the traditional classifiers, namely, HMM, ANN, RNN, and SVM. Experimental findings are clear regarding the capability of the model to follow closely real time vehicle counts data and give accurate estimations even at the busy hour traffic regime. Moreover, the optimized prediction framework has less computational time, higher accuracy of the queue length as well as minimal error in the waiting time. The efficiency and stability of the method is proven to be true statistically in all the test conditions. One of the strengths of the model is its stability concerning various traffic conditions, and it indicates that it can be applied to be used in the smart transportation systems in the reality. The research can be regarded as an improvement of the assortment of predictive traffic management systems, which will enable offering a market-tested solution to traffic congestion-related mitigation and more sophisticated traffic management. The enhancements created in to the process of prediction do not only embody the efficiency in computation but also the utility of the same.

FUTURE WORK

The inclusion of weather and environmental data to possess a more detailed background of the forecast. On the fly deployment using edge sensor networks. Expansion of traffic to multi-lane and intersection-based systems. Adaptive development on the changing traffic patterns.

REFERENCES

[1] Aissaoua, H., Laouid, A., Kara, M., Bounceur, A., Hammoudeh, M., Chait, K. (2024). Integrating homomorphic encryption in IoT healthcare blockchain systems. Ingénierie des Systèmes d'Information, 29(5): 1667-1677. https://doi.org/10.18280/isi.290501

[2] H, M.R., Warni, E., Angriawan, R., Hariadi, M., Arif, Y.M., Maulina, D. (2024). Design of flood early detection based on the Internet of Things and decision support system. Ingénierie des Systèmes d'Information, 29(3): 1183-1193. https://doi.org/10.18280/isi.290335

[3] Roy, V. (2022). Breast cancer classification with multi-fusion technique and correlation analysis. Fusion: Practice and Applications, 9(2): 48-61. https://doi.org/10.54216/FPA.090204

[4] Tubiana, D.A., Farhat, J., Brante, G., Souza, R.D. (2022). Q-learning NOMA random access for IoT-satellite terrestrial relay networks. IEEE Wireless Communications Letters, 11(8): 1619-1623. https://doi.org/10.1109/LWC.2022.3169109

[5] Khan, W.U., Ali, Z., Lagunas, E., Mahmood, A., Asif, M., Ihsan, A., Chatzinotas, S., Ottersten, B., Dobre, O.A. (2023). Rate splitting multiple access for next generation cognitive radio enabled LEO satellite networks. IEEE Transactions on Wireless Communications, 22(11): 8423-8435. https://doi.org/10.1109/TWC.2023.3263116

[6] Roy, V., Roy, L., Ahluwalia, R., Khambra, G., Ramesh, M., Rajasekhar, K. (2023). An advanced implementation of machine learning techniques for the prediction of cervical cancer. In Proceedings of the 3rd IEEE International Conference on ICT in Business Industry and Government (ICTBIG 2023), Indore, India, pp. 1-5. https://doi.org/10.1109/ICTBIG59752.2023.10456347

[7] Zhang, Q., Zhu, L., Jiang, S., Tang, X. (2022). Deep unfolding for cooperative rate splitting multiple access in hybrid satellite terrestrial networks. China Communications, 19(7): 100-109. https://doi.org/10.23919/JCC.2022.07.009

[8] Zhang, Q., Zhu, L. (2022). A deep learning approach for downlink sum rate maximization in satellite-terrestrial integrated networks. In Proceedings of the 2022 International Symposium on Networks, Computers and Communications (ISNCC), Shenzhen, China, pp. 1-5. https://doi.org/10.1109/ISNCC55209.2022.9851793

[9] Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y. (2015). Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2): 865-873. https://doi.org/10.1109/TITS.2014.2345663

[10] Sthapit, S., Lakshminarayana, S., He, L., Epiphaniou, G., Maple, C. (2021). Reinforcement learning for security-aware computation offloading in satellite networks. IEEE Internet of Things Journal, 9(14): 12351-12363. https://doi.org/10.1109/JIOT.2021.3135632

[11] Vishwakarma, S.K., Sharma, P.C., Raja, R., Roy, V., Tomar, S. (2020). An effective cascaded approach for EEG artifacts elimination. International Journal of Pharmaceutical Research, 12(4): 4822-4828. https://doi.org/10.31838/ijpr/2020.12.04.653

[12] Teket, O.M., Yetik, I.S. (2021). A fast deep learning-based approach for basketball video analysis. In Proceedings of the 4th International Conference on Vision, Image and Signal Processing, pp. 1-6. https://doi.org/10.1145/3448823.3448882

[13] Wan, G., Shan, F. (2020). Video analysis method of basketball training assistant based on deep learning theory during COVID-19 spread. Journal of Intelligent & Fuzzy Systems, 39(6): 8747-8755. https://doi.org/10.3233/JIFS-189271

[14] Zhang, R. (2022). Research on intelligent analysis technology of football video based on computer vision. In Proceedings of the 2nd International Conference on Internet of Things and Smart City, pp. 142-147.

[15] Liu, N., Liu, L., Sun, Z. (2022). Football game video analysis method with deep learning. Computational Intelligence and Neuroscience. https://doi.org/10.1155/2022/3284156

[16] Mackenzie, G., Ponte, J.R.R. (2022). Video capture and analysis of cyclists using infrastructure in the ACT through machine learning. Centre for Automotive Safety Research, University of Adelaide, Technical Report CASR202.

[17] Zhou, M., Chen, G.H., Ferreira, P., Smith, M.D. (2021). Consumer behavior in the online classroom: Using video analytics and machine learning to understand the consumption of video courseware. Journal of Marketing Research, 58(6): 1079-1100. https://doi.org/10.1177/00222437211042013

[18] Roy, V. (2024). A context-aware Internet of Things (IoT) founded approach to designing an operative priority-based scheduling algorithm. Journal of Cybersecurity and Information Management, 13(1): 28-35. https://doi.org/10.54216/JCIM.130103

[19] Mao, W. (2022). Video analysis of intelligent teaching based on machine learning and virtual reality technology. Neural Computing and Applications, 34(9): 6603-6614. https://doi.org/10.1007/s00521-021-06072-w

[20] Li, B., Zhang, H., Zhao, Y. (2019). Research based on video analysis: The integration of mathematics smart class teaching media and teachers' knowledge structure. In Proceedings of the International Joint Conference on Information, Media and Engineering (IJCIME), Osaka, Japan, pp. 348-353. https://doi.org/10.1109/IJCIME49369.2019.00076

[21] Stairs, J., Mangla, R., Chaudhery, M., Chandhok, J.S., Timorabadi, H.S. (2021). Engage AI: Leveraging video analytics for instructor-class awareness in virtual classroom settings. In Proceedings of the ASEE Virtual Annual Conference and Content Access, pp. 1-13.

[22] Abdelbasit, S.M.B. (2023). Cybersecurity attacks detection for MQTT-IoT networks using machine learning ensemble techniques. Doctoral dissertation. Rochester Institute of Technology, Rochester, NY, USA.

[23] Alam, I., Sharif, K., Li, F., Latif, Z., Karim, M.M., Nour, B., Biswas, S., Wang, Y. (2019). IoT virtualization: A survey of software definition and function virtualization techniques for Internet of Things. arXiv preprint arXiv:1902.10910.

https://doi.org/10.48550/arXiv.1902.10910
[24] Keymasi, M., Ghozatlou, O., Conde, M.H., Datcu, M. (2023). An efficient compressive learning method on Earth observation data. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, pp. 5285-5288. https://doi.org/10.1109/IGARSS52108.2023.10281758
[25] Kashyap, R., Roy, V., Patil, P.D., Manhar, A., Roy, L. (2023). Deep learning's role in advancing gastroenterology and digestive health. In Proceedings of the 3rd IEEE International Conference on ICT in Business Industry and Government (ICTBIG 2023), Indore, India, pp. 1-6. https://doi.org/10.1109/ICTBIG59752.2023.10455988
[26] Sefati, S.S., Arasteh, B., Halunga, S., Fratu, O., Bouyer, A. (2023). Meeting user service requirements in smart cities using recurrent neural networks and optimization algorithms. IEEE Internet of Things Journal, 10(24): 22256-22269. https://doi.org/10.1109/JIOT.2023.3303188
[27] Iqbal, M.A., Asiyabi, R.M., Ghozatlou, O., Anghel, A., Datcu, M. (2023). Towards complex-valued deep architectures with data model preservation for sea surface current estimation from SAR data. In Proceedings of the 20th International Conference on Content-Based Multimedia Indexing, pp. 146-152. https://doi.org/10.1145/3617233.3617271
[28] De, S., Elsaleh, T., Barnaghi, P., Meissner, S. (2012). An Internet of Things platform for real-world and digital objects. Scalable Computing: Practice and Experience, 13(1): 45-58.