Alexandria Engineering Journal (2022) 61, 8325–8334

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

Design and analysis of logistic agent-based swarm-neural network for intelligent transportation system

Monagi H. Alkinani [a], Abdulwahab Ali Almazroi [b], Mainak Adhikari [c], Varun G. Menon [d,*]

[a] *Department of Computer Science and Artificial Intelligence, College of Computer Sciences and Engineering, University of Jeddah, Saudi Arabia*
[b] *Department of Information Technology, College of Computing and Information Technology at Khulais, University of Jeddah, Saudi Arabia*
[c] *Department of Computer Science, Indian Institute of Information Technology Lucknow, India*
[d] *Department of Computer Science and Engineering, SCMS School of Engineering and Technology, India*

Received 7 December 2021; revised 26 December 2021; accepted 17 January 2022
Available online 04 February 2022

KEYWORDS

Intelligent transportation system;
Logistic framework;
Edge networks;
Swarm-neural network;
Agent-based model

Abstract Driven by the massive number of connected vehicles and the stringent requirements of data-intensive applications, logistics transportation systems have evolved to fully comprehend its effectiveness and quality to meet public transportation needs. As a result, to meet public transportation requirements and analyze the data efficiently at the edge of the networks, an advanced artificial intelligent technique needs to be introduced to make the transportation system intelligent by supporting efficient decision making, intelligent traffic control, and intrusion and misuse detection. Motivated by the challenges mentioned above, in this paper, we develop a logistic agent-based model for analyzing public transports such as cars, bus or trains in the intelligent transportation system. The intelligent logistic framework is built on a parallel neural network structure, known as a Swarm-Neural Network (SWNN). The proposed SWNN model analyzes the sensory data and recognizes the public transportation at the edge of the networks. The SWNN model is constructed so that it fits within the intelligent logistic transportation framework, and the proposed model shortens the transit time of every small-scale logistics delivery to its destination. The performance of the proposed SWNN model is evaluated using a standard TMD dataset, where the SWNN model is trained using data, retrieved multiple sensors such as accelerometer, gyroscope, magnetometer, and audio sensors. The features of the sensory data are extracted based on a 5-s time interval. The performance of the proposed SWNN model is studied over various standard machine learning techniques such as Random Forest, XGBoost, and Decision Tree. As per the simulation results, the proposed technique achieves 78–98% accuracy over a real-time dataset's different sets of features.

* Corresponding author.
E-mail addresses: malkinani@uj.edu.sa (M.H. Alkinani), aalmazroi@uj.edu.sa (A.A. Almazroi), varunmenon@scmsgroup.org (V.G. Menon).
Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2022.01.046
1110-0168

1. Introduction

Recent advancements in technology, together with the exponential rise in the number of vehicles and rapid urbanization, have paved the way to numerous developments and increased research in Intelligent Transportation Systems (ITS) [1–4]. ITS involves the interconnection of vehicles and infrastructures and provides opportunities for continuous communications, interactions, and resource sharing among vehicles with different applications for ubiquitous data analysis and processing and maximizes transportation infrastructure's efficiency with improved service quality. Recently, there has been a massive demand in supply chain management for goods to be efficiently transported from warehouses to final destinations, leading to the rapid development of ITS-enabled logistic transportation systems [5,6].

ITS-enabled logistic transportation systems and their gradual adoption of Autonomous Vehicles (AVs) are bringing massive advancements to smart cities [7]. But this has also led to the generation of unprecedented amounts of data, posing major challenges to the applications deploying ITS. With the deployment of numerous sensors in vehicles for various applications, there is a sharp increase in the data produced in ITS, rising from the Trillion-byte level to Petabyte [8,1]. Analyzing the sensory data and recognizing the public transportation systems with minimum delay is a significant concern that requires efficient and robust solutions to meet the ever-increasing demand for efficiency and quality of service [9–11].

1.1. Related works

Over these years, many methods have been proposed for analyzing and processing the data for intelligent transportation systems. Most of the methods have focused on the management and processing of data in cloud servers. The authors in [12] discuss a technique for real-time active, safe driving with a three-tier cloud computing framework. The technique helps predict and analyze the high threat of backward shock waves using the data that has been collected from vehicles' state information. An intelligent technique using cloud and big data for systematic traffic control is presented in [13]. The system uses artificial intelligence and deep learning to predict traffic flow and congestion. Managing different types of data, including video, is a significant challenge in ITS, and authors in [14] discuss an interesting technique for effective video management with the cloud. The authors also try to solve the challenges in balancing the load and storage issues using a novel parallel computing model. A recommendation system for vehicle speed using a cloud/fog computing service infrastructure is discussed in [15]. The system uses a game approach to satisfy the objectives, where the drivers are the players and the vehicle's speed is their strategy.

With the advent of edge/fog computing, many applications in ITS have tried to reduce the latency in data storage and retrieval and processing that was a significant issue with the cloud-based systems. A comprehensive survey on edge cloud computing for ITS and connected vehicles is presented in [16]. The paper presents exciting discussions and future research perspectives on how edge cloud computing can be efficiently used in ITS. In [17], a technique for identification of traffic flows on the edge node with the help of deep learning is discussed. The authors discuss a real-time vehicle tracking counter that combines identification and tracking algorithms for vehicles to detect the traffic flow. An edge-enabled distributed trustworthy storage framework with reinforcement learning in ITS is discussed in [18]. The system adopts an intelligent technique for dynamic allocation for storage using reinforcement learning based on trustworthiness and popularity. A technique that uses multiple fog servers to detect the identity of vehicles is proposed in [19]. The system uses a voting mechanism to detect the most suitable fog server, determining the real identity and the trajectory. Despite all the aforementioned works, analysis and management of data at the edge for an efficient public transportation system is still a significant concern.

1.2. Motivation and contributions

Due to the increasing growth of the connected vehicles and the long communication distance between the vehicles and cloud servers, analyzing the vehicular data in the centralized cloud server may increase the noise on the monitoring dataset and deteriorate the prediction accuracy. Thus, one of the critical challenges is to design an intelligent transportation framework that can analyze the monitoring vehicular data locally while meeting the resource requirements during data analytics. In this paper, to address the challenge mentioned above, we design a new edge-centric framework with a set of resource-constraint edge devices and a centralized cloud server for efficiently analyzing vehicular sensory data.

Another important research aspect in transportation systems is to process the vehicular sensory data with an advanced machine learning model that can process the data on resource-constraint edge devices and increase the prediction accuracy with minimum delay. One of the potential solutions in this aspect is to integrate a lightweight machine learning technique with a set of distributed edge devices to develop an intellectual and logistic framework for the transportation system that can analyze the data at the edge of the network with higher accuracy and lower precision.

Motivated by the challenges mentioned above, in this work, we develop a logistic agent-based intelligent transportation framework in edge networks for identifications of public transportation using an agent-based Swarm-Neural Network (SWNN) model. In the proposed edge framework, data scheduling and preprocessing have been performed on the distributed edge devices while analyzing the modified dataset on the centralized cloud servers using the proposed SWNN model with higher accuracy. Thus, the main novelty of the proposed method is to integrate edge networks in transportation systems for making correct decisions and introduce a new SWNN model for analyzing vehicular data with higher prediction accuracy.The significant contributions of the work are highlighted as follows.

- Design an agent-based intelligent transportation framework to assist the delivery person of any small size logistic to find in public transportation. Thus, if the smart logistic movement framework correctly identifies the vehicle such as bus, train, or taxi, then logistic movement is feasible for that vendor through those vehicles. This technique shortens the transit time of every small-scale logistics that delivers the products to its destination.

- Design an intelligent logistic framework using a parallel neural network structure to analyze vehicular data, known as the SWNN technique. The proposed strategy improves the performance of the discussed system. The SWNN model is used to recognize a vehicle, and the sensor attached to this vehicle sends data to the distributed edge server for detecting the vehicles. The SWNN model is constructed so that it fits within the intelligent logistic transportation framework and analyzes the sensory data with higher prediction accuracy
- The performance of the proposed SWNN model in the intelligent logistic transport framework is evaluated using the TMD dataset, on which the SWNN model is trained using data from multiple sensors. The sensors data used in this experiment include accelerometer, gyroscope, magnetometer, and audio sensors. Further, the sensor data are analyzed to choose the best characteristics of the proposed model. The same characteristics are then taken from additional sensor data and utilized to build the patterns. The features of the sensory data are extracted based on a 5-s time interval. The performance of the proposed SWNN model is studied over various simulation matrices with various standard machine learning techniques.

The remaining sections of the paper are organized as follows. The system model of the proposed logistic transportation system and the proposed SWNN model for analyzing the sensory data of the public transportation system in edge networks are described in Section 2. The empirical analysis of the work over various standard machine learning algorithms is described in Section 3. Finally, the conclusion and future direction of the work are highlighted in Section 4.

2. Swarm-neural network for intelligent transportation system

This section describes various components of the proposed logistic transportation system followed by the proposed SWNN model with a suitable algorithm.

2.1. System model of logistic transportation system

The intelligent transport system framework incorporates intelligent agent-based swarm-neural network approaches to give the capacity to recognize the logistic carrying vehicle. This method is meant to be easily compatible with the edge-enabled transport framework. This component of the system is implemented in the edge server to handle data processing and analysis. Different components of the proposed SWNN model of the logistic transportation system are shown in Fig. 1. As per Fig. 1, the sensory vehicular data is collected by the set of distributed edge devices, and the data is temporarily held in the data-gathering phase in the resource-constrained edge devices before being transmitted to the data processing phase. Data is sampled in this method, and the collected data is then provided to the feature selection algorithm, which calculates features from the sampled data. The selected feature is now ready to be sent into the SWNN for logistic type categorization.

The proposed technique considers the input from the vehicle's various sensors as well as a large number of vehicles submitting data to the system at the same time. This massive work of gathering data and processing it before feeding it to SWNN is broken down into four stages. The data scheduler is introduced into the system in the first phase to schedule the raw data from the sensor for data processing. The scheduler takes each sensor's data and places it in a queue to be processed by the data processing phase. The second phase is data processing, which is used to process the sensor signal using window-based sampling techniques. The features are extracted from each sample data in the third phase. The fourth phase is where the created sample is classified based on previous experience. The fifth phase's final step incorporates a rule-based decision support system to determine SWNN performance and govern overall system confidence based on that performance. The data is then stored in clod servers for any further processing and analysis in the future.

2.2. Data scheduler

In the proposed approach, the data scheduler is responsible for managing the incoming data from remote IoT devices deployed in the logistic vehicle. The scheduler is connected to several watchdogs to receive data from various IoT devices, and when the data is received, the lightweight watchdog procedures send it to the scheduler for first-come, first-served processing. In the data processing step, the data is then scheduled for processing. The data scheduler assists the system in receiving sensor data in parallel from multiple IoT devices in a single time quantum and so plays a critical role in network traffic management for the proposed method.

2.3. Data processing technique

The data processing component initially assembled the data by temporarily processing the data for each vehicle. One of the first jobs in this section is to check the correctness of each incoming data from various sensors, and only legitimate data may be sent on to the next phase. For each vehicle, sensor data is pooled over a defined time period. Following that, each vehicle's data is evaluated for feature extraction. The extraction of significant features is one of the most crucial things in sensor data processing. The data collected from numerous sensors are frequently complex and non-linear. Because of the changing traffic environment, these sensor signals with different frequencies may be seen, making them unpredictable. The sensor signals collected from various IoT devices are not stationary. As a result, the signal's enormous array is separated into N portions, with each window containing a predetermined window of size S for extracting features. The following are the features retrieved from the various windows in the proposed technique.

Correlation: This process of feature selection incorporates Correlation-based selection. The correlation between two features X_1 and X_2 is calculated

$$CO(X_1, X_2) = \frac{\sum_{r=1}^{m}(X_1 - \mu_{X_1})(X_2 - \mu_{X_2})}{\sqrt{\sum_{r=1}^{m}(X_1 - \mu_{X_1})}.\sqrt{\sum_{r=1}^{m}(X_2 - \mu_{X_2})}} \tag{1}$$

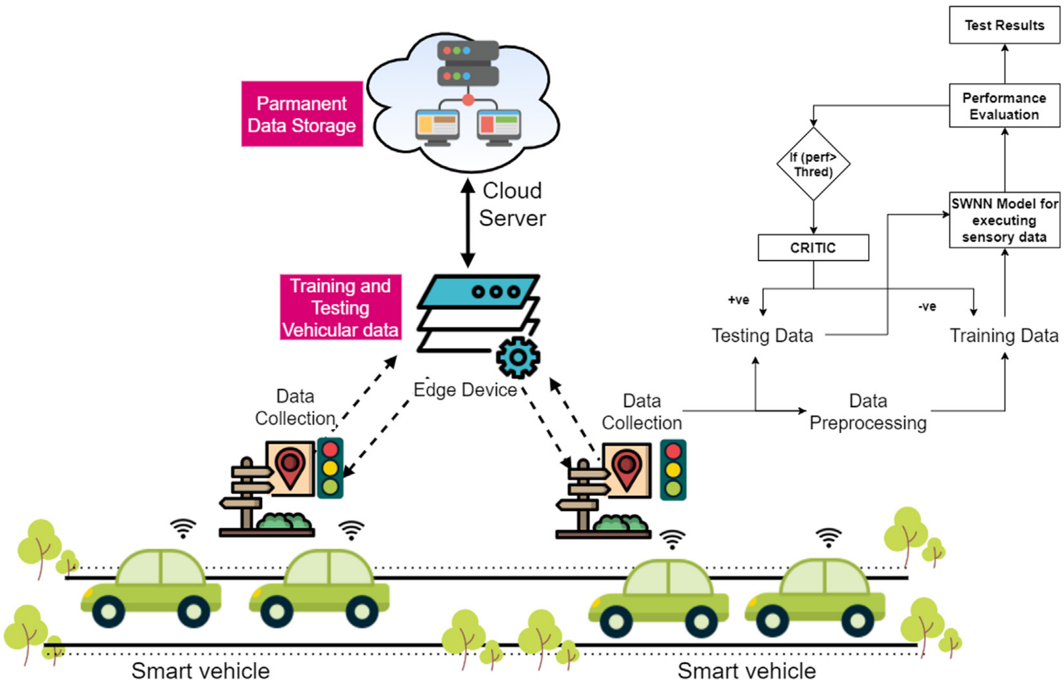

Fig. 1 System Model of Logistic Transportation System.

where the mean values of $X_1$and X_2 signal is performed as $\mu_{X_1} = \sum_{m=1}^{m} X_1$ and $\mu_{X_2} = \sum_{m=1}^{m} X_2$. The output value of $CO(X_1, X_2) \in (+1, -1)$ where the output value close to $+1$ indicates that X_1 and X_2 are similar and then one considered in the pattern and if it is close -1 proves the uniqueness in between $X_1$and X_2. Each signal's characteristic is retrieved by examining a five-second window on it. The mean, minimum, maximum, and standard deviation values are derived from the signals of several sensors. The formula for calculating these is given below.

Mean: The means of the five seconds signal data is calculated as $Mean(SI)_i^t = (\frac{1}{n}\sum_{i=0}^{i} SI_i)$, where i^{th} signal value and t represent the window of the current current iteration and N is the total number of signal values in the current window.

Minimum Value: The minimum value of each windows of signal is calculated as $Minimum(SI)_t^i = Minimum\{SI_1^t, SI_2^t, SI_3^t, \ldots, SI_i^t\}$.

Maximum Value: The maximum value of each symptom is defined as follows. $Maximum(SI)_t^i = Maximum\{SI_1^t, SI_2^t, SI_3^t, \ldots, SI_i^t\}$. In the obtained features from the vehicle sensor data, the fundamental statistical characteristics are used. These are mean, standard deviation, skewness, kurtosis, and correlation computed across a small time window of signals.

2.4. Intelligent agent-based model

The intelligent critic in the suggested system maximizes the SWNN classification method's confidence by continually training the system with newly observed data. This is what the logistics system's pre-trained SWNN-based detection is built on. This intelligent critic aids SWNN classification systems in continuous assessment and switches between training and testing phases using a rule-based mechanism.

Rule 1: $SNN_{accuracy}^{best} \geqslant \theta \Rightarrow Detection_{ready}$

Rule 2: $SNN_{accuracy}^{best} < \theta \Rightarrow Learning_{ready}$where $\forall(NN_{accuracy}^{best}) \in BoNN_{accuracy}$,

$SNN_{accuracy}^{best}$ is the best accuracy produced by neurons in the SWNN population, θ is the threshold value. Thus, when Rule 1 of the intelligent critic is met, the test data is provided to SWNN as an input. In the instance of Rule 2, the intelligent critic guarantees that the SWNN classifier is allowed to continue training.The intelligent critic notifies the gateway and edge servers if any reply is observed in the test data.

2.5. Proposed Swarm-Neural Network (SWNN) model

The SWNN classification approach is used in this proposed strategy of the intelligent logistic transportation system for the detection of different types of logistic models. The SWNN classification method is used here to detect the transport mode efficiently. This approach is empirical, making the SWNN more fit for testing different sensor data, which varies due to dynamic traffic conditions. The SWNN is designed to complete its detection task through three different phases. In the first phase, a pre-defined set of neural networks of the same architecture is generated to create the population. The initial rounds of weights and bias matrix are also generated during this phase. In the second phase, the training is started initially, and this training phase is started for the first time; then, the switching between the training and testing cycle depends on the intelligent critic. The weight and bias matrix of the neural networks in the populations is performed two times during the training phase. In the training phase, the weight and bias change based on the backpropagation algorithm, and then lastly, it is modified based on the population's best performing neural network structure.

In the proposed method, the neural network for the population is created and stored in the queue. The creation of the neural network involves the generation of the weight and bias matrix for each layer of the neural network. The weight matrix and bias matrix generation require generating the random numbers using the following equations.

$$W_i = R_i + w_C \tag{2}$$

$$B_i = R_i + b_C \tag{3}$$

where W_i and B_i is the weight and bias of i^{th} neural network in the population of the size m and $R_i \in (0,1)$ and w_c and b_c is constant. Now once the weight and bias for all the neural networks are created the population, then the SWNN classifier is ready to start the training phase. The actual output of every neural network in the population is presented as follows.

$$Y_i = \sum_{l=1}^{n} W_i P_K^T + B_i \tag{4}$$

where Y_i is the actual output of the i^{th} neuron in the neural network population. The patterns are fed into every neural network of the neural population and it is done in every iteration. Now the evaluation of neurons is performed in parallel with others, so at the end of each iteration the performance is kept in the list YO_r, and to represent the parallel the $\prod$ symbol is used.

$$YO_r = (Y_1, Y_2, \ldots\ldots Y_i)^r = \prod_{r=1}^{r} (Y_i)_r \tag{5}$$

Each transport pattern is labeled with the type of transport and the label of the pattern is the target TR_K for pattern P_k. Now, these patterns are considered to be neural network populations and feed into the population by considering one at a time. So, after every iteration, the calculation of the error EO_r for the r^{th} iteration is performed. The calculation of the Error is performed as follows.

$$EO_r = |TR_k^i - Y_k^i|_r \tag{6}$$

where TR_K^i is the target of pattern P_k The sum of square error or SSE is calculated for i^{th} neuron in the SWNN population as follows.

$$SSE_i = EO_r^T EO_r = |TR_i - Y^i|_r^T |TR_i - Y^i|_r \tag{7}$$

The matrix format for calculating the SSE is presented in Eqn. (7). Now, the first phase of the training of SWNN is performed with the back-propagation method.In this method the error is back-propagated and the sensitivity or S_l of l^{th} layer is calculated from the $(l+1)^{th}$ layer. The calculation of the sensitivity is calculated for all the neural layers of a neural network and it is calculated for every neural network in the population. The calculation of the sensitivity, SE_l of the single 3-layer feed-forward neural network is calculated as follows.

$$SE_l = \frac{\delta net_{l+1}}{\delta net_l} \frac{\delta F}{\delta net_{l+1}} = F_l(net_l)(W_{l+1})^T SE_{l+1} \tag{8}$$

where $F_l(net_l)(W_{l+1})$ is the Jacobin matrix. In this method for a neural network in the SWNN population, the sensitivity is calculated over 3-layer feed-forward network structure. So, the calculation of SE_l is started from the last layer and moves back to the first layer where the sensitivity is calculated as follows.

$$\begin{aligned} SE_1 &= \tfrac{\delta F}{\delta net_l} \\ &= \tfrac{\delta (TR_k - Y)^T (TR_k - Y)}{\delta net_l} \\ &= -2(TR_k - Y) F_l(net_l) \end{aligned} \tag{9}$$

Now, in the back-propagation phase, the weight and bias of all the neural networks in the neural population are calculated as follows.

$$W_{i+1}^l = W_i^l - \alpha SE_l (Y_{l-1})^T \tag{10}$$

$$B(i+1)^l = B_i^l - \alpha SE_l \tag{11}$$

Then Eqs. (10) and (11) is helped us to modify the weight and bias of the neural network of the neural population. After a single iteration the neural population NN_pop is evaluated and the performance of the best neuron NN_best is selected based on the following rule.

If $(Accuracy(NN_{pop}) \geqslant \theta)$ Then.

$Max(Accuracy(NN_{pop})) = NN_{best}$

Now, when this NN_{best} is decided then the last round of the weight and bias modification is performed. This equation of weight and bias modification considers the W_{best} and B_{best} of NN_{best} as the best weight and bias of the population for an iteration. The equation is constructed as follows.

$$W_{i+1} = W_i + R_1 e^{(DW)} (W_{best} - W_i) + R_2(1 - rand(0,1)) \tag{12}$$

$$B_{i+1} = B_i + R_1 e^{(DB)} (B_{best} - B_i) + R_2(1 - rand(0,1)) \tag{13}$$

where the DW and DB are the euclidean distance between the best and current weight and bias of the neural network. R_1 and R_2 are the two constant. The rand(0,1) is the random number in between 0,1. Based on the Eqs. (12) and (13) is the weight and bias of all neural networks in the neural population is updated. The algorithm of the proposed SWNN model for intelligent transportation system is presented in Algorithm 1.

Algorithm 1. Proposed SWNN Model

INPUT: Pattern P_k, Target TR_k, Neural population NN_{pop}, Population weight W_{pop}, Population Bias B_{pop}, Learning rate α
OUTPUT: NN_{best}, W_{i+1}, B_{i+1}
Fixed the number of iteration: EP_k
for K:1 to $Size(NN_{pop})$ **do**
  create weight and bias matrix for every NN based on Eqs. (2) and (3)
  $W_{pop} \leftarrow W_i$
  $B_{pop} \leftarrow B_i$
end for
Maximum Independent Run $= M_r$
while $(Counter \leqslant M_r)$ **do**
  for LP_i:1 to $Size(NN_{pop})$ **do**
    Create lightweight process: $LP_{process}^i$
    $LP_{process}^i \leftarrow NN_{Model}(\alpha, W_i, B_i)$
    Each neuron in NN_{pop} updates W_i and B_i using Eqs. (10) and (11)
    Error of each NN $E_i \Rightarrow E_{pop}$
    Calculate $Accuracy(NN_{pop})$ using Eq. (14)
  end for
  if $Max_{accuracy}(NN_{pop}) \geqslant \theta)$ **then**

(*continued on next page*)

Update Best neuron in NN_{pop}
$NN_{best}^{new} \Rightarrow Max_{accuracy}(NN_{pop})$
end if
Find $W_{best} \Rightarrow NN_{best}$
Find $B_{best} \Rightarrow NN_{best}$
for p:1 to $Size(NN_{pop})$ **do**
$DW \leftarrow Euclidean(W_i, W_{best})$
$DB \leftarrow Euclidean(B_i, B_{best})$
Update W_i using Eq. (12)
Update B_i using Eq. (13)
end for
$Counter = Counter + 1$
if $Accuracy(NN_{best}) \geqslant \Phi$ **then**
Break
end if
end while

3. Empirical evaluation

The proposed method's success is dependent on the SWNN-based classification approaches' performance, which aids in detecting the logistic transit type. This section evaluates and improves the suggested swarm neural network approach for the TMD dataset for optimum performance by modifying different parameters over a wide range of values. The SWNN is evaluated for its performance in terms of solution quality and stability in delivering the best solution. The simulation is performed over a dataset that is relevant to the theory presented in this work, and in this dataset, data is sent via car sensors, making it relevant for the experiment. The performance of the suggested technique is compared with the standard benchmark algorithms such as Random Forest, XGBoost, and Decision Tree.

3.1. Simulation setup and dataset

The test is carried out on an Intel i5 9*th* generation computer with 8 GB RAM and a 4 GB graphics card, as well as a 1 TB hard drive. The Numpy, Scipy, and Panda libraries are used to create the SWNN simulation, and the Matplotlib library is used to plot the studied results. The proposed method's performance is assessed in terms of solution quality, which is measured by calculating accuracy. The typical machine algorithm's classification error, precision, recall, and accuracy are evaluated.

The TMD dataset is separated into sections based on the number of features. These features are derived from the sensor data received and processed by the data processing section. The sensor signal is separated using a set-sized window in the processing stage. The features are then gathered from this window. The experiment employs an accelerometer, gyroscope, magnetometer, and audio sensors. The dataset is now sorted into three groups depending on the available features. The first dataset contains 12 features, the second contains 32 features, and the third dataset contains 36 features. For comparing the performance of the proposed approach and other machine algorithms across different dimensions of the transport dataset, the dataset is divided into these three groups. The performances of the proposed method and the standard machine algorithm are analyzed based on classification error, precision, recall, and accuracy as follows,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

where TP = true positive, TN = True negative, FP = False Positive, and FN = False Negative. In the performance measurement, the high value of precision and recall reflects the highly accurate result.

The correlation value of the characteristics in the dataset is used to analyze them and the correlation matrix is constructed based on Eqn. 1. Such matrix is shown in Fig. 2. The following rule is used during the correlation-based feature selection process.

$If\ CO(X_1, X_2) > 0.5 Then$
$Remove\ X_2$

Where X_1 and X_2 are the two different features. This rule states that the strongly correlated for the correlation matrix are eliminated. After eliminating the strongly linked features, the correlation matrix is re-plotted and shown in Fig. 3.

The correlation matrix of 32 feature dataset without and with the implementation of the Rule are shown in Figs. 4 and 5, respectively. Similarly, the correlation matrix of 36 feature dataset without and with the implementation of the Rule are shown in Figs. 6 and 7, respectively.

3.2. Parameter analysis for SWNN model

The parameter analysis for the proposed method is performed in an empirical way to adjust the parameters for the SWNN method over the transport mode dataset. The adjustment of these parameters will remain the same for any number of fea-

Fig. 2 Correlation matrix of the 12 feature dataset without the implementation of the Rule.

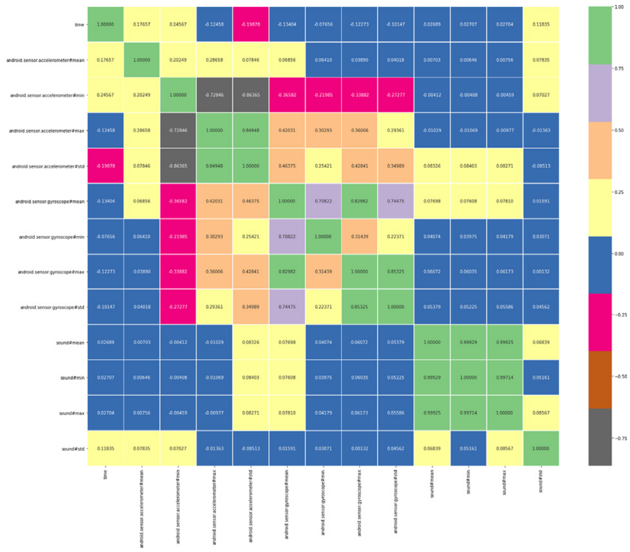

Fig. 3 Correlation matrix of the 12 feature dataset after the implementing the Rule.

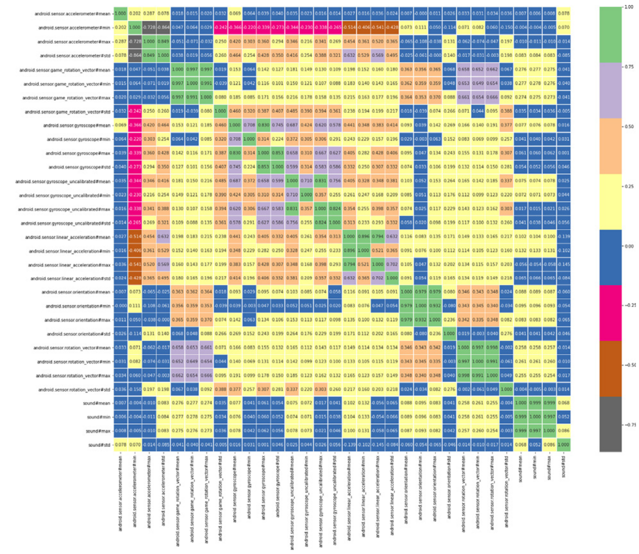

Fig. 5 Correlation matrix of the 32 feature dataset after the implementing the Rule.

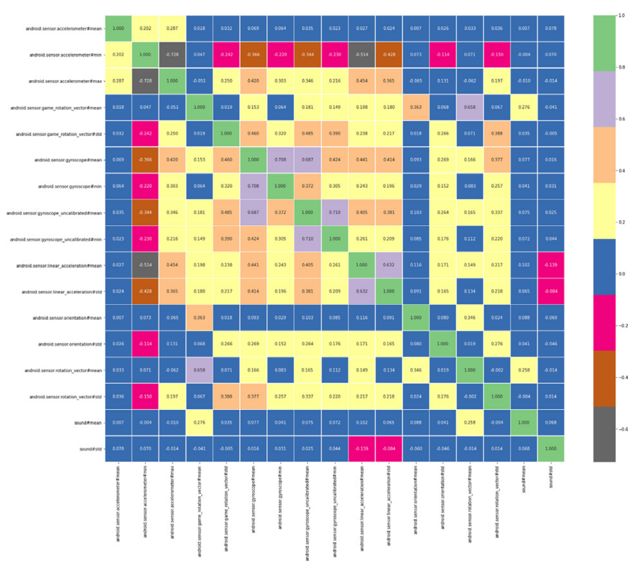

Fig. 4 Correlation matrix of the 32 feature dataset without the implementation of the Rule.

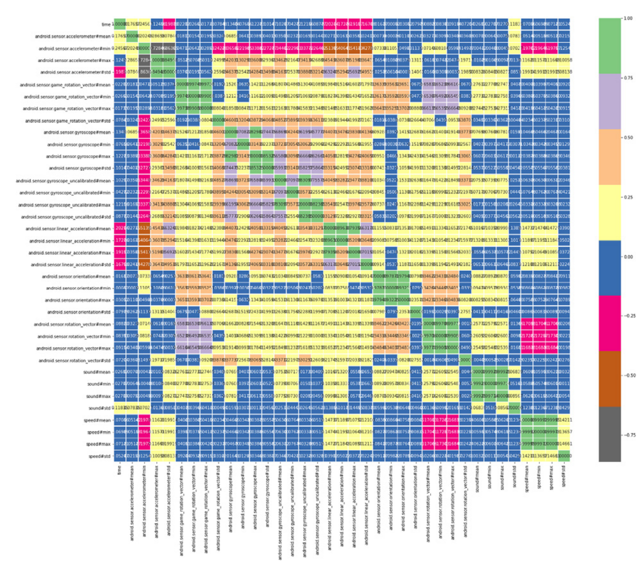

Fig. 6 Correlation matrix of the 36 feature dataset without the implementation of the Rule.

tures used for the dataset. The parameter analysis for every parameter is initiated with a wide range of values, but only a few of them are selected for the presentation purpose. The parameters that impact the performance of the proposed SWNN model are neural population size and learning rate. The population size is tested with the different values, but the 5, 10, and 20 population size is used for the presentation purpose. (See Tables 1–3).

Considering the higher population size becomes unnecessary as the SWNN achieves its maximum accuracy with a population size of 20. The error produced with population size 10 and 20 is 0.024 and 0.006. Hence, population size 20 is good with the learning rate range 0.06 to 0.125. The Learning rate ranges are used instead of the single learning rate for neural networks of the neural population. The use of different learning rates creates a variation in the performances in the neural networks of the neural population. The learning rate in which the best performance is observed is 0.06 to 0.125. The higher learning rate is not used here because the proposed learning rate produces the best accuracy over the transport mode dataset. The overall fixed values of the different parameters are described in Table 4.

3.3. Comparative result analysis and discussion

Proposed swarm neural network-based techniques are applied over the dataset consisting of different numbers of features. This variation is analyzed to understand the stability of the proposed SWNN model over different dimensions of the dataset. The solution quality is also analyzed by comparing the per-

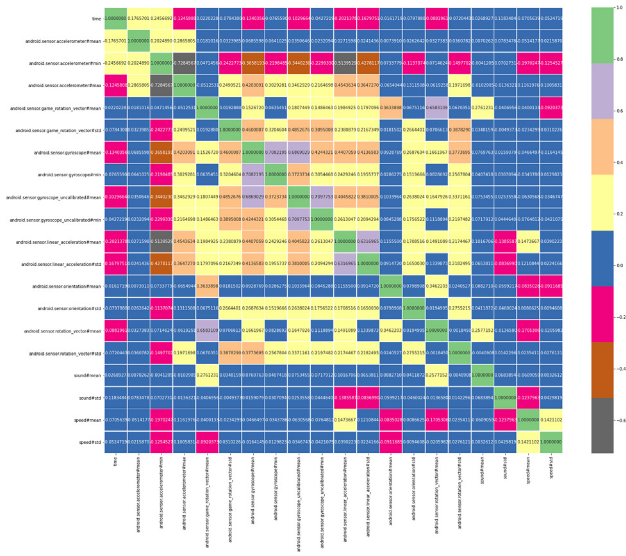

Fig. 7 Correlation matrix of the 36 feature dataset after the implementing the Rule.

Table 1 Performance analysis of SWNN by varying the learning rate with population size 5.

	Population size = 5		
	Accuracy	Precision	Recall
Range1 (0.0001, 0.009)	0.781	0.701	0.771
Range2 (0.01, 0.06)	0.801	0.799	0.787
Range3 (0.06 0.125)	0.883	0.853	0.853

Table 2 Performance analysis of SWNN by varying the learning rate with population size 10.

	Population size = 10		
	Accuracy	Precision	Recall
Range1 (0.0001, 0.009)	0.867	0.857	0.857
Range2 (0.01, 0.06)	0.901	0.901	0.901
Range3 (0.06 0.125)	0.976	0.966	0.966

Table 3 Performance analysis of SWNN by varying the learning rate with population size 20.

	Population size = 20		
	Accuracy	Precision	Recall
Range1 (0.0001, 0.009)	0.971	0.970	0.970
Range2 (0.01, 0.06)	0.988	0.988	0.988
Range3 (0.06 0.125)	0.994	0.989	0.989

formance of the proposed solution with another standard benchmark algorithm. The algorithms are Random Forest [20], Decision Tree [21], and K-Nearest Neighbor [22]. From the past few decades, Random Forest, Decision Tree, and K-Nearest Neighbour algorithms have achieved higher prediction accuracy, precision, and recall over the other machine learning models in various fields related to the Internet of Things, such as smart transportation systems, smart healthcare, smart agriculture, etc. Thus, to show the outperformance of the proposed SWNN model in the agent-based logistic transportation framework, we have introduced Random Forest, Decision Tree, and K-Nearest Neighbour algorithms for comparative analysis over a benchmark dataset. The performance of these algorithms is shown in Table 8.

Table 4 Parameters of Swarm-NN.

Sl. No.	Parameter Names	Value
1	NN layer size	3
2	Noneuron/layer	1
3	No of NN in population	20
4	Learning range for neural population	0.6 to 0.125
5	ϕ (Eq. 16, Eq.17)	0.25

From Tables 5–7, it is observed that the performance of the algorithm increases with the number of features. Now, if the minimum number of features are available then the performance of the SWNN is also better than that of the standard algorithms. The performance of the SWNN over the dataset with 12 features is 24.3% more than that of the average accuracy of the other three algorithms, Tables 5 and 8. In the case of the dataset with 32 and 36 features, the performance increases by 26.3% and 14% respectively.

Table 5 Comparative analysis of proposed method (SWNN) with standard algorithms over 12 features.

Algorithms	Precision	Recall	Accuracy
Random Forest	0.71	0.71	0.77
Decision Tree	0.67	0.68	0.69
K-nearest Neighbour	0.51	0.49	0.51
Proposed Method	0.88	0.89	0.9

Table 6 Comparative analysis of proposed method (SWNN) with standard algorithms over 32 features.

Algorithms	Precision	Recall	Accuracy
Random Forest	0.75	0.76	0.8
Decision Tree	0.67	0.67	0.68
K-nearest Neighbour	0.54	0.53	0.54
Proposed Method	0.89	0.9	0.94

Table 7 Comparative analysis of proposed method (SWNN) with standard algorithms over 36 features.

Algorithms	Precision	Recall	Accuracy
Random Forest	0.83	0.83	0.84
Decision Tree	0.85	0.84	0.88
K-nearest Neighbour	0.78	0.79	0.8
Proposed Method	0.97	0.97	0.98

Table 8 Comparative analysis of the proposed method over different machine learning algorithms.

	12 Features	32 Features	36 Features
Random Forest	0.77	0.80	0.84
Decision Tree	0.69	0.68	0.88
K-nearest Neighbour	0.51	0.54	0.8
Proposed Method	0.90	0.94	0.98

The performance of the standard machine learning algorithms and the proposed SWNN approach is compared in Table 5, where the random forest achieves the second-highest performance and the SWNN outperforms all other methods on the dataset with 12 features. The random forest yielded a 23% inaccuracy, while the proposed SWNN approach produced 10% inaccuracy. Within the limited simulated environment, there is a significant 13% improvement in performance. The random forest has the second-best performance over the 32 feature dataset with a reported error rate of 20%. The performance of the proposed SWNN approach has improved by 0.04% for this 32-feature dataset compared to the previous one. This significant improvement in the SWNN method claimed that the performance is attributable to the increased availability of features in the dataset. Now, a similar pattern of performance improvement can be seen with the dataset with 36 features, which produces only 0.02% error for logistic type prediction. The performance of the decision tree is the second-best in the list for the dataset with 36 features. However, the performance of the decision tree is 10% less than the proposed SWNN approach, which makes the SWNN approach the best-suited method for predicting this kind of logistic transportation.

This performance is an enhancement in the proposed algorithm is only possible for the parallel implementation of the neural networks from the neural population and its weight adjustment strategy. The learning of the proposed method is dynamic because the selection of the best neural network is performed in each iteration and then the learning of the best neural network of the population is shared with the other neural network in the population. This performance of the proposed SWNN strategy also reflects its stability in the performance over the different datasets. The performance quality of the proposed method is represented graphically in Figs. 8 and 9 and this shows that the quality of the solution is also outperforming the other strategies of machine algorithm presented here with a limited simulation environment.

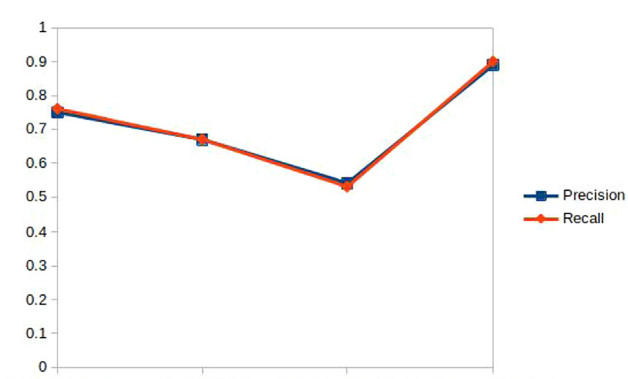

Fig. 8 Comparative analysis of the proposed method with other standard machine learning algorithm based on precision and recall value.

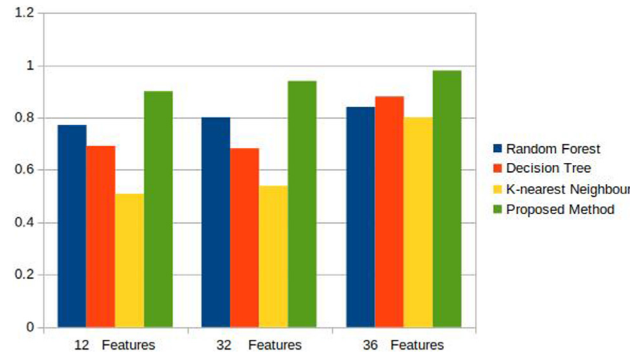

Fig. 9 Comparative analysis of the proposed method with other standard machine learning algorithm over different numbers of features.

4. Conclusion

This paper has designed a new logistic agent-based model for analyzing public transports at the edge of the networks and making decisions intelligently for public transportation. For analyzing real-time sensory data of public transportation, retrieved from multiple vehicle sensors, we have introduced a new Swarm-Neural Network (SWNN) model. The SWNN model is constructed with a set of parallel neural networks, and it fits within the intelligent logistic transportation framework for data analytics. The proposed model shortens the transit time of every small-scale logistics delivery to its destination. The main goal of the SWNN model is to detect the transportation model efficiently by training the weight and bias matrix of the neural networks. The performance of the proposed SWNN model is evaluated over a standard dataset, namely the TMD dataset that consists of multiple sensory data of public transports such as accelerometer, gyroscope, magnetometer, and audio sensors. During the evaluation, the features of the sensory data are extracted based on a 5-s time interval before analyzing the dataset using the proposed SWNN model. A set of simulation analyses with multiple features have been performed to show the outperformance of the proposed SWNN model over the standard machine learning techniques. The comparative analysis represents that the proposed SWNN model achieves 78–98% accuracy over the standard machine learning algorithms. In the future, we will develop a logistic agent-based intelligent framework at edge networks with advanced communication technology such as 6G networks to reduce network congestion and analyze the vehicular sensory data with higher accuracy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was funded by the University of Jeddah, Saudi Arabia. The authors, therefore, acknowledge with thanks the University technical supports.The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number MoE-IF-20-01.

References

[1] L. Zhu, F.R. Yu, Y. Wang, B. Ning, T. Tang, Big data analytics in intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 20 (1) (2019) 383–398.

[2] M. Won, Intelligent traffic monitoring systems for vehicle classification: A survey, IEEE Access 8 (2020) 73340–73358.

[3] M. Abbasi, A. Najafi, M. Rafiee, M.R. Khosravi, V.G. Menon, G. Muhammad, Efficient flow processing in 5g-envisioned sdn-based internet of vehicles using gpus, IEEE Trans. Intell. Transp. Syst. 22 (8) (2021) 5283–5292.

[4] B. Ji, X. Zhang, S. Mumtaz, C. Han, C. Li, H. Wen, D. Wang, Survey on the internet of vehicles: Network architectures and applications, IEEE Commun. Stand. Mag. 4 (1) (2020) 34–41.

[5] J.J.Q. Yu, A.Y.S. Lam, Autonomous vehicle logistic system: Joint routing and charging strategy, IEEE Trans. Intell. Transp. Syst. 19 (7) (2018) 2175–2187.

[6] M. Humayun, N. Jhanjhi, B. Hamid, G. Ahmed, Emerging smart logistics and transportation using iot and blockchain, IEEE Internet Things Mag. 3 (2) (2020) 58–62.

[7] Y. Zhang, G. Zhang, R. Fierro, Y. Yang, Force-driven traffic simulation for a future connected autonomous vehicle-enabled smart transportation system, IEEE Trans. Intell. Transp. Syst. 19 (7) (2018) 2221–2233.

[8] A. Munusamy, M. Adhikari, M.A. Khan, V.G. Menon, S.N. Srirama, L.T. Alex, M.R. Khosravi, Edge-centric secure service provisioning in iot-enabled maritime transportation systems, IEEE Trans. Intell. Transp. Syst. (2021) 1–10.

[9] X. Zheng, W. Chen, P. Wang, D. Shen, S. Chen, X. Wang, Q. Zhang, L. Yang, Big data for social transportation, IEEE Trans. Intell. Transp. Syst. 17 (3) (2016) 620–630.

[10] Y. Liu, Big data technology and its analysis of application in urban intelligent transportation system, in: 2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS), 2018, pp. 17–19.

[11] S. Goudarzi, N. Kama, M.H. Anisi, S. Zeadally, S. Mumtaz, Data collection using unmanned aerial vehicles for internet of things platforms, Comput. Electr. Eng. 75 (2019) 1–15.

[12] B.-J. Chang, J.-M. Chiou, Cloud computing-based analyses to predict vehicle driving shockwave for active safe driving in intelligent transportation system, IEEE Trans. Intell. Transp. Syst. 21 (2) (2020) 852–866.

[13] M. Shengdong, X. Zhengxian, T. Yixiang, Intelligent traffic control system based on cloud computing and big data mining, IEEE Trans. Industr. Inf. 15 (12) (2019) 6583–6592.

[14] Q. Hao, L. Qin, The design of intelligent transportation video processing system in big data environment, IEEE Access 8 (2020) 13769–13780.

[15] M. Asadi, M. Fathy, H. Mahini, A.M. Rahmani, An evolutionary game approach to safety-aware speed recommendation in fog/cloud-based intelligent transportation systems, IEEE Trans. Intell. Transp. Syst. (2021) 1–10.

[16] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, A. Mouzakitis, A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles, IEEE Trans. Intell. Transp. Syst. (2021) 1–16.

[17] C. Chen, B. Liu, S. Wan, P. Qiao, Q. Pei, An edge traffic flow detection scheme based on deep learning in an intelligent transportation system, IEEE Trans. Intell. Transp. Syst. 22 (3) (2021) 1840–1852.

[18] F. Qiao, J. Wu, J. Li, A.K. Bashir, S. Mumtaz, U. Tariq, Trustworthy edge storage orchestration in intelligent transportation systems using reinforcement learning, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 4443–4456.

[19] K. Gu, K. Wang, X. Li, W. Jia, Multi-fogs-based traceable privacy-preserving scheme for vehicular identity in internet of vehicles, IEEE Trans. Intell. Transp. Syst. (2021) 1–18.

[20] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[21] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and regression trees, Routledge, 2017.

[22] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Stat. 46 (3) (1992) 175–185 [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879.