Collaborative Cloud-V. Edge System for Predicting Traffic Accident Risk Using Machine Learning Based IOV

Zeroual Djazia [*]
University of Biskra, ALGERIA

Okba Kazar[†]
United Arab Emirate University, UAE.

Saad Harous[‡]
University of Sharjah, UAE.

Saber Benharzallah[§]
Batna 2 University, ALGERIA

Abstract

Smart city development is profoundly impacted by cutting-edge technologies such as information and communications technology (ICT), artificial intelligence (AI), and the Internet of Things (IoT). The intelligent transportation system (ITS) is one of the main requirements of a smart city. The application of machine learning (ML) technology in the development of driver assistance systems, has improved the safety and the comfort of the experience of traveling by road. In this work, we propose an intelligent driving system for road accident risks prediction that can extract maximum required information to alert the driver in order to avoid risky situations that may cause traffic accidents. The current acceptable Internet-of-vehicle (IOV) solutions rely heavily on the cloud, as it has virtually unlimited storage and processing power. However, the Internet disconnection problem and response time are constraining its use.

In this case, the concept of vehicular edge computing (V.Edge.C) can overcome these limitations by leveraging the processing and storage capabilities of simple resources located closer to the end user, such as vehicles or roadside infrastructure. We propose an Intelligent and Collaborative Cloud-V.Edge Driver Assistance System (ICEDAS) framework based on machine learning to predict the risks of traffic accidents. The proposed framework consists of two models, CLOUD_DRL and V.Edge_DL, Each one complements the other. Together, these models work to enhance the effectiveness and accuracy of crash prediction and prevention. The obtained results show that our system is efficient and it can help to reduce road accidents and save thousands of citizens' lives.

Key Words: IOV, deep learning, deep reinforcement learning, cloud computing, V.Edge computing, cloud-V.Edge collaboration.

[*] Computer Science Department. Email: d.zeroual@univ-batna2.dz.
[†] Department of Information Systems and Security.
[‡] Computer Science Department.
[§] Computer Science Department.

1 Introduction

Every year the lives of approximately 1.3 million people are cut short, as a result, of road traffic accidents. Between 20 and 50 million people, suffer non-fatal injuries, however, many of these cases result in different kinds of disabilities. Timely and accurate prediction of traffic accidents has great potential to ensure traffic safety and reduce economic losses. To enhance this traffic safety, many studies have been conducted to help the development of an Active Traffic Management System. The main areas of interest covered by these studies are i) black-spot detection where road traffic accidents have been concentrated [16]. ii) Detection of traffic incidents in real time and alert people to reduce their effects [37]. iii) Road accidents prediction, where the prime goal of this research is to predict the road accidents before they occur [24, 17, 32]. Road accident prediction is a field of significant and contemporary scientific interest. The prediction with high spatiotemporal resolution is difficult, mainly due to the complex traffic environment, human behavior and the lack of real-time traffic-related data [29]. With the recent development of Internet-of-vehicle (IOV) technology and the advancement in wireless communications, and computational systems, new opportunities have opened-up for intelligent traffic safety, comfort, and efficient solutions.

The interest in machine learning has increased exponential due to the wide availability of parallel computing technology and a large amount of training data [5]. In particular, the success of deep learning (DL) technique led the researchers to investigate the application of machine learning for traffic accident prediction. Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment; its combination with DL generates a new powerful algorithm called deep reinforcement learning (DRL). These algorithms have recently been very successful in implementing road safety applications. However, the enormous resources required by these algorithms pose a great challenge to the limited computational and storage resources that are available on-board every vehicle.

The Internet-Of-Vehicle (IOV) solutions to traffic safety problems rely heavily on the cloud, as it has virtually unlimited storage and processing power; where data must be moved from the data source location (IOV sensors) to a centralized location in the cloud. However, in addition to the Internet disconnection problem, the cloud might be far from the location of sensors and devices generating these data, which will cause the response time to be slow. Therefore, this might restrict the use of a solution that is based on the cloud, for sudden car accident prediction.

The concept of V.Edge Computing is an efficient alternative to overcome the limitations of using machine-learning models in the cloud platform. Many emergency predictions take place close to the end user; therefore, they can be processed at the edge nodes. This reduces the impact of communication delay and internet disconnection. We propose an Intelligent Collaborative Cloud-V.Edge Driver Assistance System (ICEDAS) framework based on machine learning, which predicts the risks of traffic accidents.

This framework takes advantage of the strengths of the two platforms, where a Deep Q-Learning Network (DQN) algorithm is adopted in the cloud, in order to train an intelligent agent to warn the driver of any foreseeable risk of traffic accident based on the huge historical data available on the cloud. On the other hand, a deep learning algorithm can be deployed on the V.Edge platform for inference, covering potential response absences by the cloud in predicting sudden traffic risk due to the platform's proximity to the end user. The DL algorithm is trained in the cloud, taking advantage of its scalability and high-end computing resources for model training. Figure 1 illustrates the three layers of our system. The proposed (ICEDAS) aims to achieve the following main objectives:

1) The system must be able to react in a timely manner to warn the driver before entering a critical state.

2) The system must deliver adaptive messages to each driver who is at risk of a traffic accident based on their personal conditions.

3) The system must have the ability to use the cloud and V.Edge to predict the accident risk prediction in an efficient manner.

The rest of this paper is organized as follows: Section 2 presents the literature review of existing works. In Section 3, we briefly describe the proposed system architecture. In Section 4, we provide details regarding the intelligent Cloud-V.Edge system for predicting road accident risk. The experimental results are presented in Section 5, and the work is concluded in Section 6.

2 Related Work

Many researchers have investigated the use of machine learning for traffic safety and accident risk prediction during the past few decades. In this section, we review the different categories of these studies.

2.1 Traffic Accident Prediction using Classical Techniques

Numerous researchers have approached the prediction of traffic accidents by considering it as either a classification problem or a regression problem. In this section, we will explore several studies that have utilized classical machine learning techniques to address this problem. For instance, in [20] Lv et al. investigated the identification of potential traffic accidents by employing the k-nearest neighbor method with real-time traffic data. Hossain and Muromachi [12] utilized a Bayesian belief net (BBN) for real-time crash prediction on basic freeway segments of urban

Figure 1: Vehicular cloud – edge system architecture

expressways. In another study by Lin et al in [18], a novel Frequent Pattern Tree (FP Tree) approach based on important variable selection was proposed to achieve an acceptable level of accuracy in real-time traffic accident risk prediction. Chang and Chen in [6] developed a decision tree model to build a classifier for accident prediction, achieving training and testing accuracies of 55%. In [13] Karlaftis and Vlahogianni compared statistical methods with neural networks (NN) in transportation-related research and demonstrated the promising potential of NN-based solutions. Furthermore, Zhang et al utilized in [40] a statewide live traffic database to develop real-time traffic crash prediction models. They compared Random Forest (RF), Support Vector Machine (SVM), and Extreme Gradient Boosting (XGBoost) models. Although many previous systems have treated traffic accident risk prediction as a classification problem, their prediction accuracy has been relatively unsatisfactory.

2.2 Deep Learning for Traffic Accident Prediction

Recently, with the rapid advancements and remarkable achievements in machine learning technologies, several recent studies have embraced deep learning methods for predicting traffic accidents. In the study [28] by Ren et al., a deep learning approach based on Recurrent Neural Networks (RNN) was proposed to predict traffic accident risk, where risk was defined as the number of accidents occurring in a specific region at a given time. Another method by Chen et al called "STENN" in [7] was introduced for traffic accident prediction, incorporating multiple factors such as spatial distributions, temporal dynamics, and external factors to improve prediction accuracy. Yu et al. [38] developed an autoencoder deep architecture to examine the impact of human mobility on traffic accident risk. They utilized this approach to gain insights into how human mobility patterns influence the occurrence of traffic accidents. In the work conducted by Yuan et al. [39], a Hetero-Convolutional Long-Short Term Memory (Hetero-ConvLSTM) model was proposed to forecast the number of traffic accidents in Iowa. This model incorporated both spatial and temporal features, enhancing the accuracy of accident predictions. The focus of the research by Gutierrez et al. [10] was on developing a Deep Learning Ensemble Model that utilizes information extracted from social media to predict traffic accidents. Peng et al. [26] presented DeepRSI, a real-time road safety prediction framework that utilized mobile sensing data collected in Vehicular Ad-Hoc Networks (VANETs).

2.3 Deep Reinforcement Learning for Traffic Safety

In recent years, deep reinforcement learning (DRL), an advanced form of artificial intelligence, has gained significant importance in intelligent decision-making across various domains. DRL has found applications in robotics [35], healthcare [33], Natural Language Processing [2], and sentiment analysis [34].

In the field of transportation systems, DRL algorithms have been widely utilized, particularly in traffic control tasks. For example, DRL has emerged as the most popular machine learning methodology for traffic signal control [11]. In ramp metering control [19], a DRL-based method was proposed to leverage video traffic data for improving the efficiency of ramp metering. This approach utilized traffic video frames as inputs and learned optimal control strategies directly from visual data. In the context of intelligent transportation, an improved Deep Q-Learning Network (DQN) method has been adopted to train intelligent agents for guiding vehicles to their destinations and avoiding congestion [15]. Furthermore, in reference [5], Deep Reinforcement Learning (DRL) has been utilized to develop autonomous braking systems capable of intelligently regulating vehicle velocity to prevent collisions. For autonomous driving or self-driving cars, DRL algorithms have garnered considerable attention and have been the subject of extensive research [3, 4, 30, 41].

The DRL algorithm has proven highly efficient in solving complex decision-making problems that were previously beyond the capability of traditional machine learning techniques. However, when operating in a dynamic environment, such as in the case of traffic prediction and prevention, the algorithm requires frequent updates of the data being exploited in order to provide reliable predictions. Additionally, due to the significant storage and computing resources required, its application is best suited for deployment on a cloud platform.

3 Proposed System Architecture

Our main goal is to develop a framework that leverages machine learning techniques to help drivers in safe driving practices. We plan to achieve this by analysing large amounts of data from previous accidents. The proposed framework consists of an intelligent and collaborative driver assistance system, called 'ICEDAS' that operates between the cloud and a vehicle's edge. Figure 2 illustrates the two layers in this framework, which work together to safeguard drivers and minimize the risk of road accidents.

3.1 Cloud Layer

Cloud computing is one of the most significant trends in the information technology evolution, as it has created new opportunities that were never possible before [14]. Due to its storage capacity and computing power, we consider it the suitable location to generate the two machine learning models in our system. The first model is DRL, which is the main component in our framework. It runs on the cloud to predict accident risks. The second model is a DL, also generated in the cloud and then deploy it to the V.Edge device for inference when needed, to cover the absence of prediction by the cloud.

3.2 V.Edge Layer

Vehicular Edge Computing (VEC), based on the edge computing motivation and fundamentals, is a promising technology supporting ITS services, and smart city applications [22]. In our system, the V.Edge is used to replace the cloud in certain cases, such as internet disconnection or bandwidth overload. Vehicles equipped with cameras, radars, GPS, and other devices can sense both the internal and external environment and collect various information such as speed, road quality, position, and more. These data are either sent to the cloud in real-time for prediction by DRL, or used by the edge itself to replace cloud prediction in generating

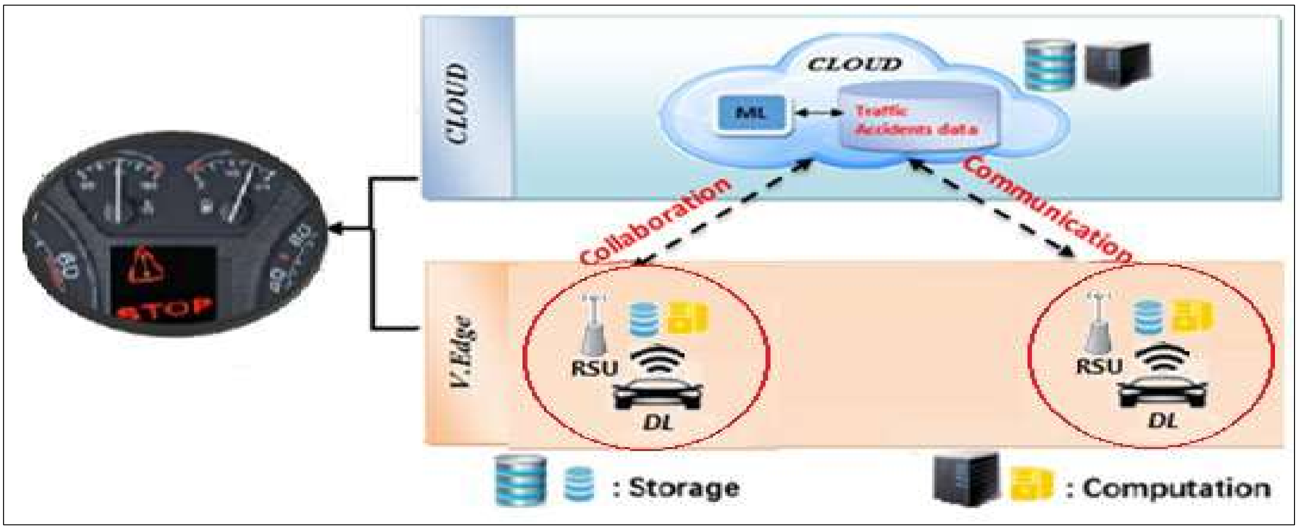


Figure 2: Cloud_DRL and V.Edge_DL system architecture

accident risk alerts using the inferred DL model.

4 Methodology

In this section, we describe the general structure of our ICEDAS. We first define possible scenarios then we discuss its operations in detail.

4.1 Scenarios

Many accidents occur when driving conditions suddenly change. ICEDAS must detect the potentially accident-causing events in advance and help the driver take the appropriate actions to avoid them. To predict a traffic accident risk, we focus on many contributing factors that frequently cause traffic accidents. They are often related to *Drivers, Roads or Vehicles* such as: *Driver's age, Driver's Sex, Driver's experience, Road condition, Light condition, Weather condition, Type of vehicle, Service year, etc.*

When a sudden change in any of the car's normal conditions is accurately detected, it may be difficult to adapt properly to this change, which may become a threat to the car. In this case, we need an intelligent risk prediction system that adapts to different situations of this risk. Markov Decision Process (MDP) is a powerful technique for modeling sequential decision-making problems. We used MDP to formulate our problem. In MDP framework, an agent interacts with a given environment state by taking actions at discrete time steps. In our system, we assume that the traffic environment follows the discrete-state. Figure 3 describes this Markov process. The state (SN-risk) implies that the system did not detect any risk. Once a risk is detected, the state (SN-risk) is changed to the state (S-risk). In practical scenarios, it is difficult to know the transition probabilities of the Markov process and the distribution of the environment states. Therefore, reinforcement-learning approach can be applied to learn the risk prediction policy through the interaction with the environment.

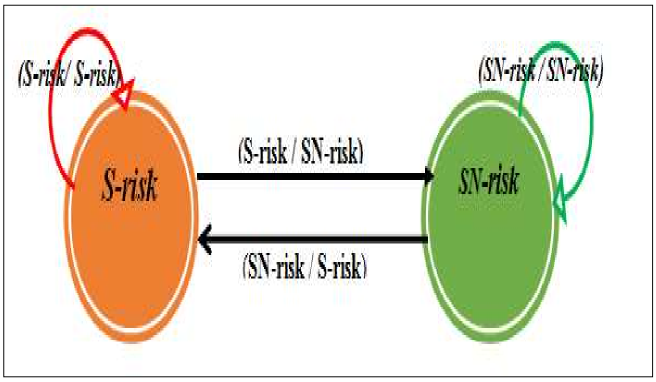


Figure 3: Traffic environment-state description by the discrete-state Markov process

4.2 Cloud Deep Reinforcement Learning for Traffic Accident Risk Prediction

In this section, we present the details of the proposed Cloud_DRL based risk prediction system. We first present the structure of DQN and explain in detail how it works to train the learning model based on accidents data available in the cloud.

4.2.1 Key Elements of Cloud _DRL. There are four key elements in this DRL system: Cloud-Agent, observation/state, action, and reward scheme.

We formulate traffic accident risk prediction problem as a reinforcement learning problem shown in Figure 4, where the Cloud-Agent interacts with the vehicle traffic environment in discrete time steps (t_0, t_1, t_2… t_N). The agent's objective is to reduce the number of accidents.

- *Cloud –Agent***:** the agent observes the state of each vehicle, in its environment, defined by St_i at the beginning of time step t_i, then selects an action $At_i \in A$ to perform. The use

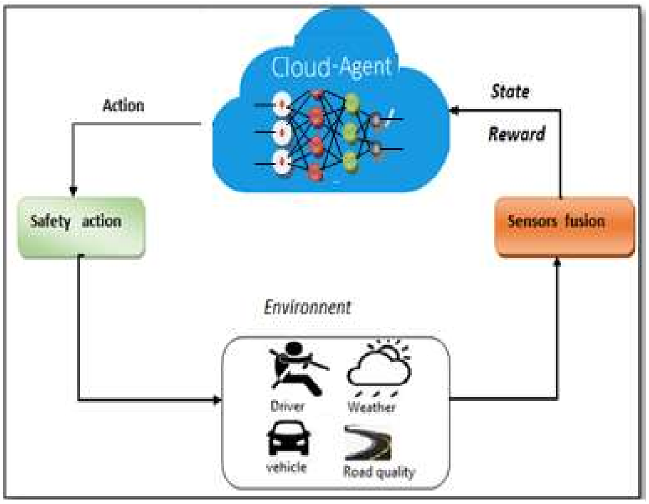


Figure 4: Proposed Cloud DRL based accident risk

of deep neural network (DNN) model, in this case, is very appropriate because of the huge number of states. The DNN take input observations about traffic accidents and produces action decisions that should be taken as its output. The DNN architecture is a multilayer-network where the Cloud–Agent explores the information (available in the Cloud) about various accidents that have occurred previously and recommend the best actions that must be applied to avoid similar accidents from happening again.

- *Action*: refers to the decision recommended by a Cloud - Agent. It is a feedback on a state of risk accident, which is one of the following actions (*Stop, Deceleration, and No-Change of lane)* as an output to avoid this risk of accident.
- *State*: is an efficient representation of current road traffic condition. The representation variables contain multiple parameters reflecting the circumstances of a specific zone of an urban transportation network to precisely describe the complexity of its dynamics. The agent learns through interacting with the environment episode by episode, where each episode ends with the prediction of accident risk for a vehicle, and the next episode starts.
- *Reward (penalty):* the agent gets a reward Rt_i at the end of time step t_i as a result of the applied action At_i. The key requirement for a successful application of reinforcement learning is to design a reward function that frames the goal of an application and guides the learning towards a desirable behavior [23]. To reduce the traffic accident risk, it is reasonable to reward the agent at each time step for choosing an action that led to the avoidance of accidents [9]. Therefore, we determine the reward (penalty) Rt_i for the agent who chooses an action At_i at time step t_i as follows:

$$Rt_i = \left\{ \begin{array}{c} 0 \quad \longrightarrow \ (No-Change) \\ 1*(\mathrm{N}) \longrightarrow (Deceleration) \\ 2*(\mathrm{N}) \longrightarrow (Stop) \end{array} \right\} \qquad (1)$$

Where *N is a negative integer,* which represents the severity of an action. The agent can perform one of these actions (*No-Change*, *Deceleration or Stop*) according to accident severity: *{(0) Negligible risk, (1) Serious risk and (2) Fatal risk}*

The goal of reinforcement learning system is to achieve a safe road traffic system with no accident risk rate during the evaluation time (T). This is represented by the Total-Reward ($T\_Rt_i \approx 0$):

$$T\_Rt_i = \sum_{i=1}^{T} |\mathrm{Rt_i}| \qquad (2)$$

4.2.2 Deep Q-Network (DQN). There are classical RL algorithms such as Q-learning, Policy Gradient (PG), Actor Critic, etc. Q-learning is one of the popular RL methods, which search for the optimal policy in an iterative fashion [5]. This algorithm is not suitable when we have a huge number of states and complex state transitions. In this work, a DQN algorithm that uses a DNN is utilized for predicting accident risks, with the aim of enhancing both the speed and accuracy of predictions. For each episode, the Cloud-Agent observes

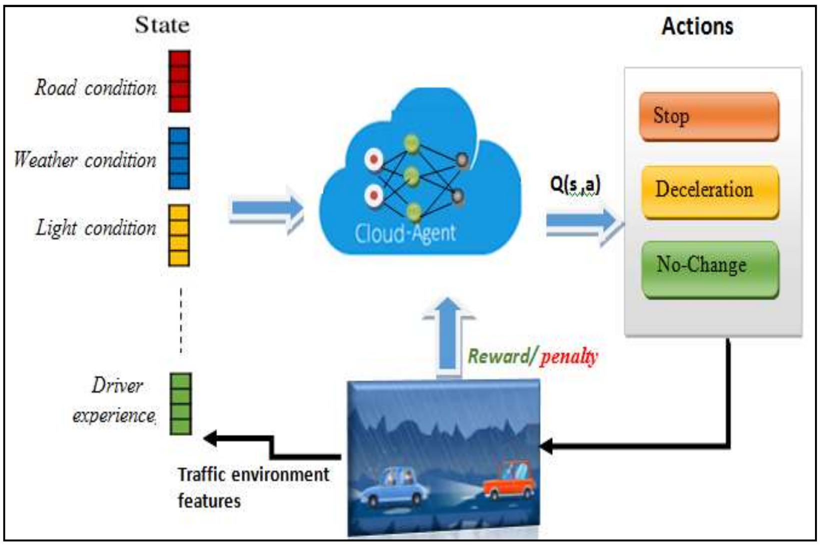


Figure 5: Cloud _DQL accident risk prediction

state St_i at the beginning of time step t_i, then makes action decision according to vehicle state, and receives a sequence of rewards (Rt_i) after time steps. If the cloud agent aims to reduce vehicle road accidents, it is sufficient to choose an action that maximizes the immediate reward Rt_i.

Since the agent aims to reduce the number of accidents in the long run, it needs to find an optimal policy noted ($\pi*$) at every possible state-action pair. To find the optimal policy $\pi*$, we need to find the optimal Q-value:

$$Q\pi*(s, a) = \max Q\pi(s, a)$$
$$= Q*(s, a) \quad (3)$$

When the state space is continuous, it is impossible to find the optimal value of the state-action pair $Q*(s, a)$ for all possible states. To deal with this problem, the DQN method was proposed, which approximates the state-action value function $Q(s, a)$ using the DNN, i.e, $Q(s, a) \approx Q\theta(s, a)$, where θ are parameters of the DNN that will be learned from raw traffic accident data.

We construct such a DNN network, where the network input is the observed traffic environment state St_i and the output is a vector of estimated Q-values Q(s, a, θ) for all actions a ∈ A under observed state St_i. Figure 5 illustrates the Cloud_DQN module for traffic accidents prediction. Real-traffic accident data was collected in a buffer called a replay buffer to train our network. We build a neural network connected to several layers so that DNN approaches the Q-value. The agent learns parameters θ by training the DNN network to minimize the following mean squared error (MSE) as the loss function. MSE can be defined as the average squared difference between the target value and the predicted value [27], as shown in Equation (4):

$$MSE = \frac{1}{K}\sum_{i=1}^{k}(y_i - \hat{y}_i)^2 \quad (4)$$

Where y is the target value, $\hat{y}$ is the predicted value, and K is the number of training samples. Our target value should be the optimal Q value; the optimal Q value can be obtained by using the Bellman optimality Equation (5), where its Q value is just the sum of the reward (r) and the discounted maximum Q value of the next state-action pair [27]:

$$Q^*(s,a) = r + \Upsilon max Q^*(s', a') \quad (5)$$

Therefore, we can define our loss as the difference between the target value (the optimal Q value) and the predicted value (the Q value predicted by the DQN) and express the loss function L as (6) [27]:

$$L(\Theta) = Q^*(s,a) - Q_\Theta(s,a) \quad (6)$$

Substituting Equation (5) in Equation (6), we get Equation (7).

$$L(\Theta) = r + \Upsilon max Q(s', a') - Q_\Theta(s,a) \quad (7)$$

The Q value of the next state-action pair in the target is computed by the target network parameterized by **θ'** and the predicted Q value is computed by the main network parameterized by **θ**. The loss function is represented by Equation (8).

$$L(\Theta) = \frac{1}{K}\sum_{i=1}^{K}(r_i + \Upsilon max Q_{\Theta'}(s', a') - Q_\Theta(s,a))^2 \quad (8)$$

The target network has the same architecture as the main network but different weights. Every N step, the weights from the main network are copied to the target network, where N is a hyperparameter that can be set by the user. Using both networks leads to more stability in the learning process and helps the algorithm to learn more effectively. To find the optimal parameter θ, we use gradient descent. We compute the gradient of our loss function$\nabla_\theta \mathrm{L}(\theta)$and update the network parameter θ as:

$$\theta = \theta - \alpha\nabla_\theta L(\theta) \quad (9)$$

The algorithm for training the Cloud_DQN is defined on next page.

4.3 V.Edge Deep Learning

Deep learning is one branch among the many fields of machine learning, and it is based on artificial neural networks [21]. Since DL often requires high performance computing resources (GPUs, CPUs and storage devices) for model training and execution on massive data [36], the resources available in a vehicle may not fulfil this stringent requirement. Meanwhile, there is an imprecise trend: the more layers and parameters of a deep neural network, the more accurate the decision-making, which would undoubtedly increase the training and running cost of deep learning models (DLMs) [36]. In this case, the cloud is the best solution to handle a huge traffic accident data due to its scalability, availability of resources, and cost-effectiveness.

In some situations, such as sudden accidents, where a fast response is the most important variable in the accident risk prediction problem, it is not always effective to rely on the cloud to send risk predictions. This is because predictions sent from the cloud to the driver may be lost or delayed due to internet disconnections or bandwidth overloads.

The best solution is to build a deep learning model based on big data for traffic accident risk prediction in cloud platform, and then transfer it to the V.Edge to cover this cloud prediction absence. The V.Edge_DL can learn deep connections between traffic accidents and their spatial-temporal patterns. Deep learning is a deeper network of neurons, which consists of input layer, hidden layers, and an output layer. It aims to exploit historical traffic accident data to avoid their reoccurrence again. The input layer of the model would represent the variables that are known to influence accident severity, such as crash timing, speed limit, weather conditions, and so on. The output layer would represent the degree of risk according to the severity, which could be classified into three categories: *Negligible risk, Serious risk,* and *Fatal risk.* We construct a DL model by region, which are transferred to the *V.Edge* when it is needed. Figure 6 illustrates V.Edge_DL *construction* and *exploitation*.

Algorithm DQN algorithm

Initialize the main network with random weights θ
Initialize the target network with random weights θ' by copying the main network parameter θ
Initialize the replay memory capacity as *D;*

For *episodes = 1, M* **do**
  (1) Randomly select the initial road traffic state *s*

  For t = 1, T **do**
    (2) Observe the state *s and select* random action *a* with probability ε otherwise select
      $a = \arg\max Q_\theta(s, a)$
    (3) Execute action *a and move* to the next state s'and obtain the reward *r*
    (4) Store experience (s, a, r, s') in replay memory *D*
    (5) Randomly sample a minibatch of *K* transitions from the replay memory *D*
    (6) Compute the target value, that is, $y = r + \gamma \max Q_{\theta'}(s', a')$
    (7) Compute the loss function valueL *(θ) with* (8)
    (8) Compute the gradients of the loss and update the main network parameter *θ using (9)*
    (9) Every N step update $\theta' \leftarrow \theta$
  End For

End For

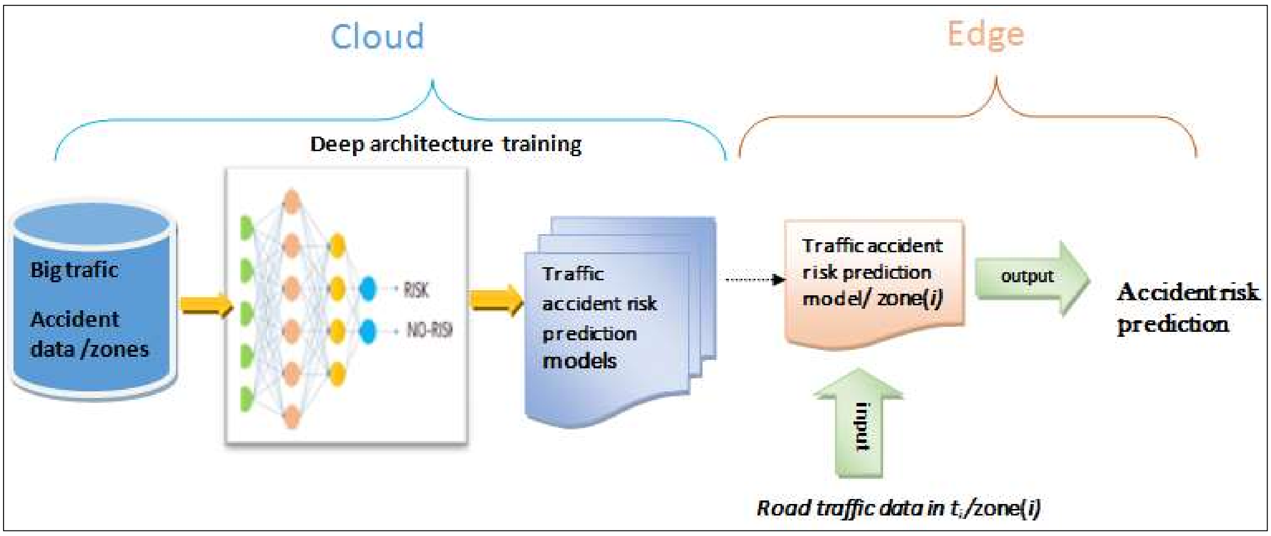


Figure 6: Regional V.Edge_DL traffic accident risk prediction

4.4 Collaboration V.Edge _DL / Cloud _DRL

The V.Edge does not have sufficient capacity to store and process a large amount of IOV data and generate DL models, so, it is not easy to ensure an absolute quality of traffic safety. Therefore, it uses models generated in cloud level, Figure 7.

We have adopted a collaborative work between Cloud platform and V.Edge platform through a distributed learning system that uses both platforms for an optimal prediction. The cooperation-communication between these platforms can have a vertical *V2C* (V.Edge- Cloud) or horizontal *V2V* (V.Edge–V.Edge) type.

4.4.1 V.Edge – Cloud.

• Communication (V.Edge - Cloud): In IOV technology, the sensors enable gathering information about the road, the vehicle and the driver, to be sent to the cloud using V2C

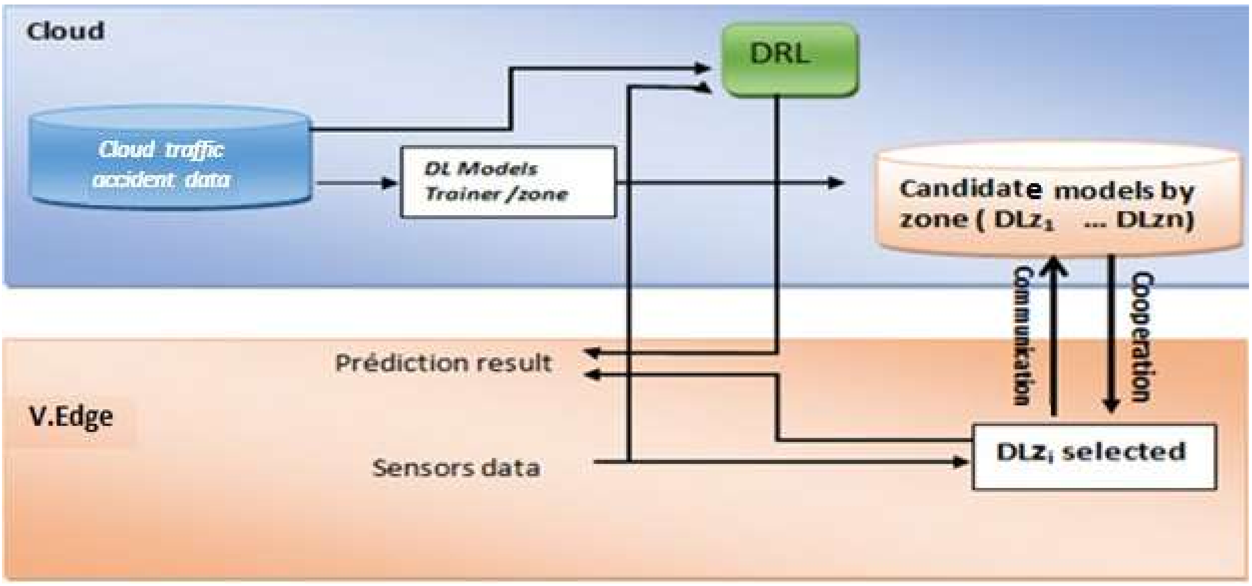


Figure 7: Collaboration / communication (V. Edge -cloud)

(vehicle to Cloud) communication and interaction. This information is used to predict if there is a risk of accident occurring and in this case, the driver is warned by an alert message sent by the Cloud_DRL entity as soon as possible.

• Collaboration (Cloud -V.Edge): Collaboration between cloud and V.Edge platforms can involve training a deep learning model in the cloud using high-end computing resources, and then deploying the model to the V.Edge device for inference when needed. This approach can help to cover potential response absences by the cloud and ensure real-time data processing and decision-making without delay.

4.4.2 V.Edge – V.Edge. Embedded a deep learning model into different vehicles enables effective collaboration and communication among them for accurate prediction of road accidents.

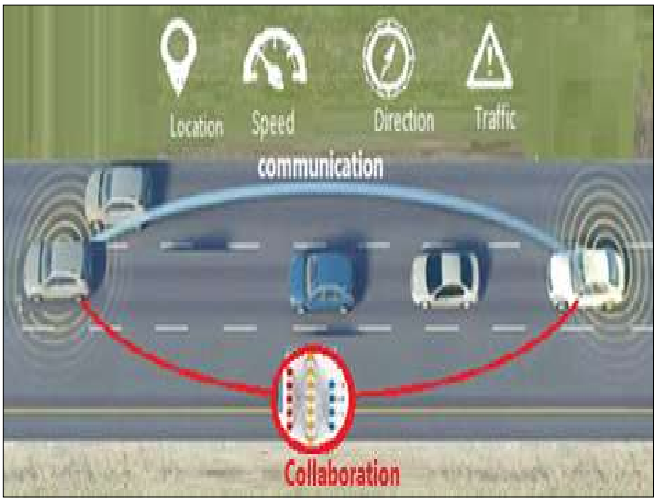


Figure 8: Collaboration / communication (V. Edge - V.Edge)

• Communication (V.Edge - V.Edge): Vehicles communicate with each other and exchange data through wireless communication protocols (V2V). The shared data may include information about the vehicle's speed, direction, location, or any other relevant data that could help the vehicles avoid potential collisions and dangerous behaviors.

• Collaboration (V.Edge – V.Edge): A vehicle can refer to another vehicle for importing the deep learning model of its new zone in case of internet problems with the cloud.

5 Experiments and Results

In this section, after describing the used data, we evaluate the effectiveness of the proposed models. Several machine learning methods are compared through a series of experiments. All implementations are in Python which utilize Tensorflow [1], Keras [8], and scikit-learn [25] libraries.

5.1 Data

To evaluate our accident risk prediction framework, we utilized road accident data from the United Kingdom, which is available on the website www.data.gov.uk. The data includes accident information ranging from the year 2005 to 2017 with 34 features, and vehicle information ranging from 2004 to 2016 with 24 features, comprising two million records. The dataset is considered a big data, which requires preprocessing to improve the performance of machine learning models and obtain accurate results. The preprocessing steps include data cleaning, data transformation, and data reduction. A machine learning feature selection method such as the Scikit-learn Random Forest library was used to identify the most relevant and correlated attributes influencing the learning process, which are depicted in Figure 9. Table 1 presents the important features description of this dataset, which will form the input vector of our ML models.

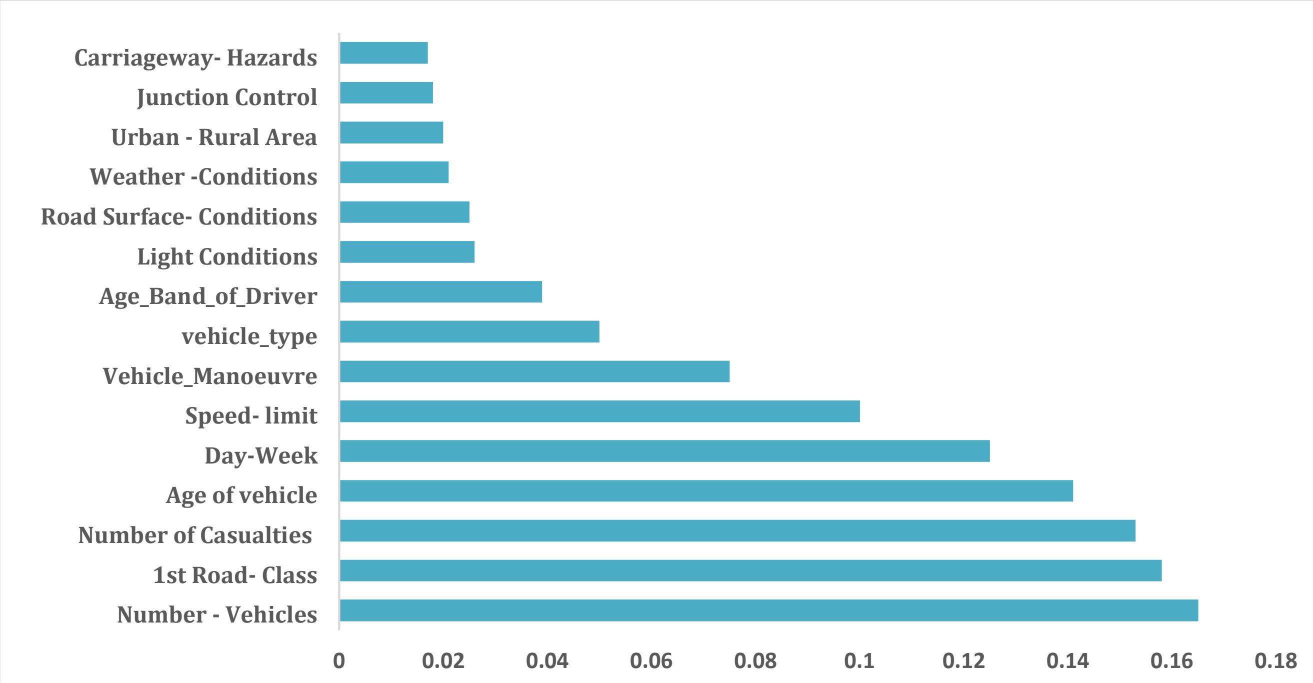


Figure 9: Attribute importance scores

Table 1: Input Factor

Variable	Label
Number – Vehicles	Vehicles involved
1st Road- Class	Motorway, A (M), A, B, C , Unclassified
Number of Casualties	Casualties involved
Age of vehicle	0-10,11-20,21-30,31-40, Above 40 years, Missing
Day-Week	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
Speed- limit	Speed limit in [mph]
Vehicle_Manoeuvre	Going ahead, Turning left/right/U, Reversing, Parked, Slowing/stopping/waiting, Overtaking, Others, Missing
vehicle_type	Pedal cycle, Motorcycle, Car, Bus, Truck, Others
Age_Band_of_Driver	<24,25–34,35–44,45–54,55–64,65–74,>75
Light Conditions	Daylight, Darkness—lights lit, Darkness—lights unlit, Darkness—no lighting, Darkness—lighting, unknown
Road Surface- Conditions	Dry, Wet, or damp, Snow, Frost or ice, Flood over 3 cm deep, Oil or diesel, Mud
Weather –Conditions	*Fine no high winds, Raining no high winds, Snowing no high winds, Fine + high winds, Raining + high winds, Snowing + high winds, Fog or mist, Other, Unknown*
Urban - Rural Area	*Urban, Rural, Unallocated*
Junction Control	*Not at junction or within 20 m, Authorized person, Auto traffic signal, Stop sign, Give way or uncontrolled*
Carriageway- Hazards	*None, Vehicle load on road, Other object on road, Previous accident, Dog on road, Other animal on road, Pedestrian in carriageway—not injured, Any animal in carriageway (except ridden horse)*

5.2 Evaluation Metrics

It is necessary to identify and estimate the efficiency and effectiveness of Cloud_DRL and V.Edge_DL in predicting traffic accidents with the data set. Our models are validated in terms of:

- Learning curves (Accuracy and Loss) for both of *Cloud_DRL and V.Edge_DL.*
- Comparison with other algorithms in terms of evaluation metrics.
- Efficiency and effectiveness in reducing the risk of road accidents with or without cooperation.

The calculation of evaluation metrics is mainly based on (N x N) confusion matrix (shown in Figure 10) that is used to display the performance of the algorithm, where N is the number of target classes. This matrix compares the actual target values with those predicted by the machine learning model. To comprehensively measure the performance of the proposed models, accuracy, sensitivity, F1 score, and other indicators are used. The concept and formula for calculating each of these indicators are shown in Table 2. Where TP denotes true positive, FP denotes false positive, TN denotes true negative, and FN denotes false negative.

		Predicted values	
		Positive	***Negative***
Real values	***Positive***	***TP***	***FN***
	Negative	***FP***	***TN***

Figure 10: Confusion matrix

5.3 Results and Discussion

During the construction of our machine learning models, the dataset was divided into training dataset (80%) and test dataset (20%).

Table 2: Main metrics for classification

Metric	**Formula**	**Interpretation**
Accuracy (Acc)	$\frac{(TP+TN)}{TP+TN+FP+FN} * 100\%$	Gives the proportion of the total number of predictions that were correct
Precision (Pre)	$\frac{TP}{(TP+FP)} * 100\%$	How accurate the positive predictions are
Recall (Sensitivity)	$\frac{TP}{(TP+FN)} * 100\%$	Gives information about the True Positives that are correctly classified during the test.
Specificity	$\frac{TN}{(TN+FP)} * 100\%$	Gives information about of True Negatives that are correctly classified during the test.
F1-score	$\frac{2*TP}{(2*TP+FN+FP)} * 100\%$	Hybrid metric useful for unbalanced classes

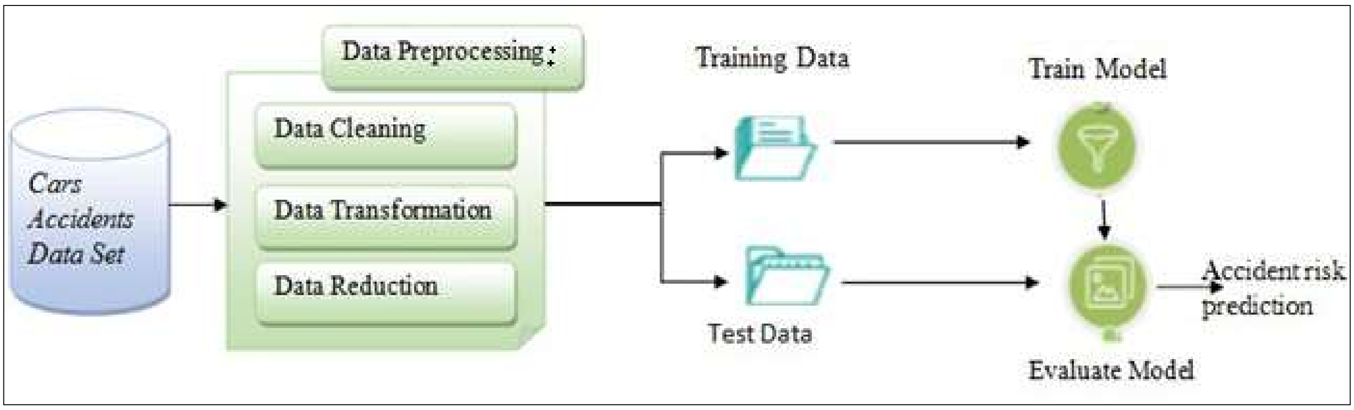


Figure 11: Experiment procedure

5.3.1 Cloud_DRL Vs V.Edge_Dl Learning Curves. To build the best traffic accident predictive framework, we used a Convolution Neural Network (CNN), which is one of the best classification algorithms based on artificial neural networks, for both the Cloud_DRL and V.Edge_DL models. CNN is designed to learn automatically and adaptively using multiple building blocks such as convolution layers, pooling layers, and fully connected layers.

For the V.Edge_DL model, the CNN algorithm consists of five convolutional layers with 32 filters of size 3, and five max pooling layers. The output of these layers was then flattened and passed through two fully connected layers before being processed by a softmax activation function to produce three output predictions. In the case of Cloud_DRL model, DRL was integrated into the same Convolutional Neural Network (CNN) architecture used in the first model, to produce the same number of outputs, each representing an action to be performed. The results obtained in terms of accuracy and loss for both models are displayed in Figure 12.

5.3.2 Performance Comparison. The proposed models Cloud_DRL and V.Edge_DL are compared to other well-known algorithms [31], such as: *Logistic Regression* (*LR*), *support vector machine (SVM), decision trees (DT), Random forests (RF)*, and *XGBoost* in terms of Accuracy, Sensitivity, Specificity, Precision, and F1-score measures. The experiment results are summarized in Table 3. Figure 13 visualized the results in Table 3.

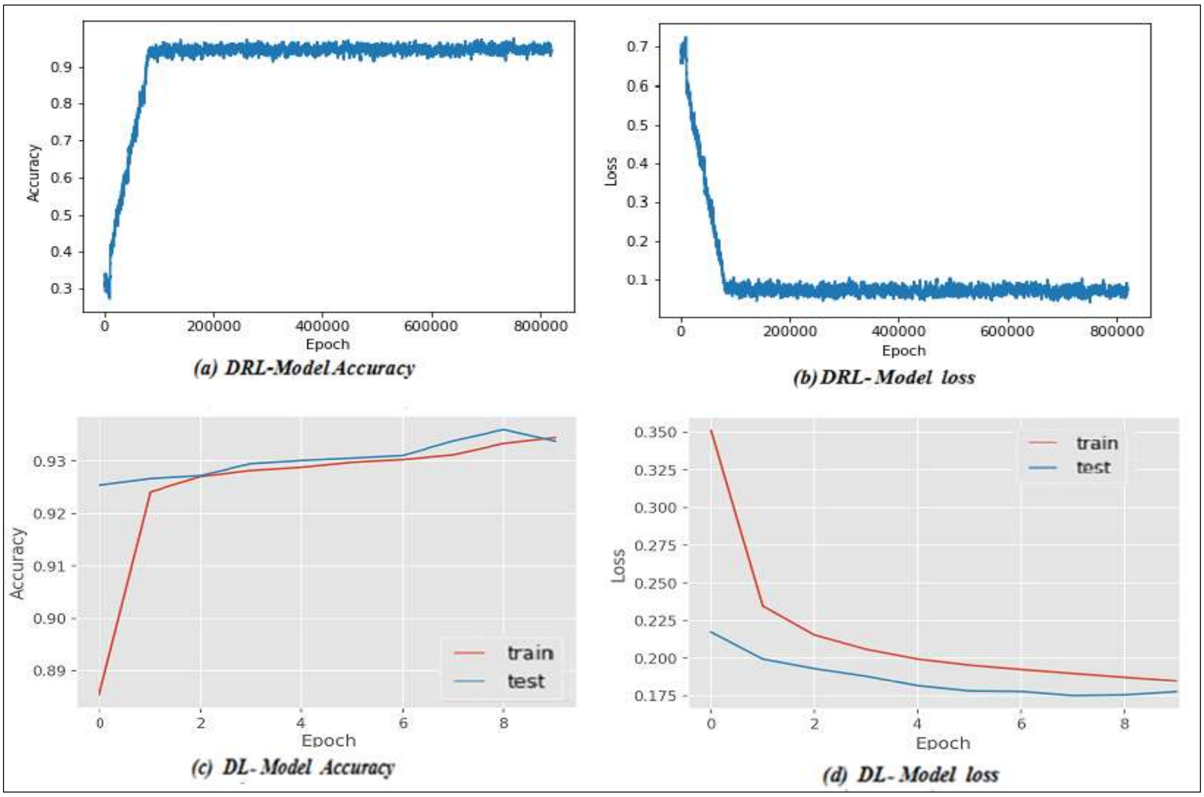


Figure 12: Learning curves for Cloud_DRL and V.Edge_Dl models

Table 3: Comparison of (Cloud_DRL, V.Edge_DL) with baseline

Classification Techniques	*Accuracy*	*Sensitivity*	*Specificity*	*Precision*	*F1 score*
Cloud_DRL	**0.94**	**0.98**	**0.92**	**0.84**	**0.91**
V.Edge_DL	**0.93**	**0.98**	**0.90**	**0.83**	**0.90**
LR	0.76	0.28	0.97	0.83	0.42
SVM	0.89	0.82	0.95	0.93	0.87
DT	0.91	0.89	0.90	0.92	0.91
RF	0.92	0.89	0.91	0.94	O.92
XGBoost	0.93	0.99	0.96	0.66	0.80

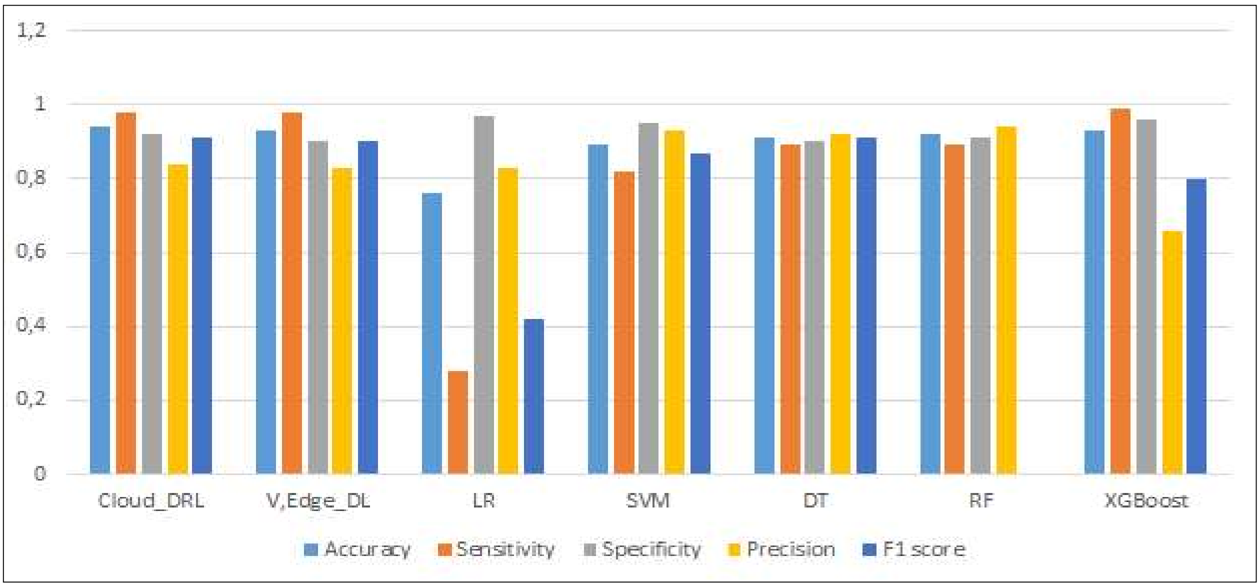
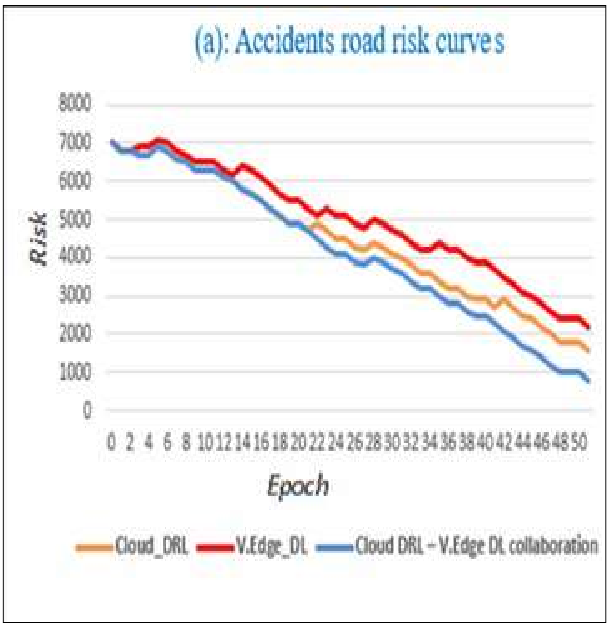


Figure 13: Visual comparison with baselines

5.3.2 Cloud_DRL – V.Edge_DL Collaboration. *Cloud_DRL*, *V.Edge_DL* collaboration leads to efficient and effective prediction of traffic accident risk. The results obtained by each model individually and then together are shown in Figure 14.

5.3.4 Discussion. Figure 12 represents the accuracy and the loss of both models *Cloud _DRL* and *V.Edge _DL*. Figure 12 (a) plots the increment of *Cloud_DRL* accuracy in function of epoch's number; its accuracy starts very low and ends very high. The main reason of this distinction is due to a balance between the two explorations and exploitation strategies. At the beginning of the algorithm, each action is performed randomly, which is useful for helping the agent learn more about its environment. Whenever the agent takes more steps, the exploration decreases, and the agent starts to exploit more of the good actions that it has detected. Towards the end of the training process, the search space becomes very limited. Therefore, the agent concentrates more on the exploitation step. This leads to a significant increase in accuracy. It is the same similar justification for the loss curve (Figure 12 (b)), which reduces the error to a minimum.



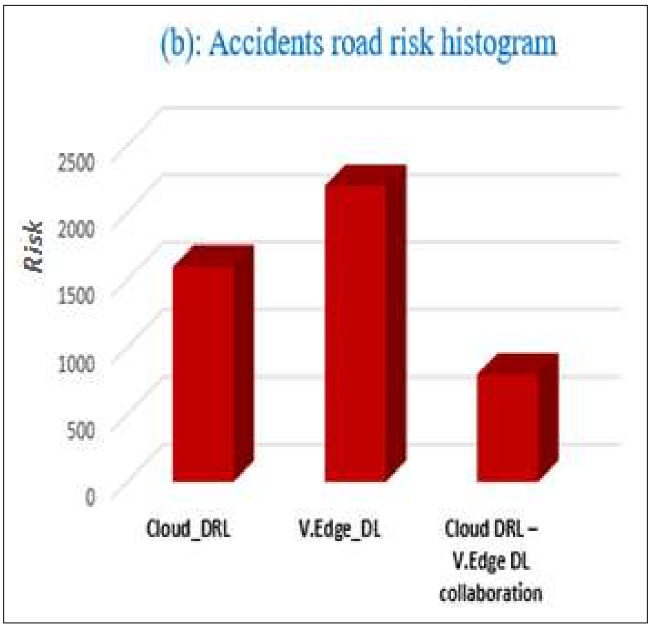


Figure 14: Cloud _DRL - V.Edge_DL Collaboration

DL has also shown better performance (Figure 12(c, d)), but DRL remains the strongest in solving dynamic problems where the environment changes over time and the optimal decision-making strategy may vary depending on the state of the environment. This poses challenges for traditional DL algorithm that lack the ability to adapt to changing conditions.

To present how well our accident risk prediction models are performing, we compared them with other algorithms that use the same performance measures. Table 3 summarizes the obtained results when applying these machine learning algorithms including *LR*, *SVM*, *DT*, *RF* and XGBoost. We note that *Cloud_DRL* and *V.Edge_DL* give a high performance in term of Accuracy, Sensitivity, Specificity, Precision, F1-score measures. We can see that the *Cloud _DRL* achieved the highest degrees of accuracy 94%, Sensitivity 98%, Specificity 92%, Precision 84% and F1-score 91%. After *Cloud_DRL* algorithm, the *V.Edge_DL* classifier generates a good result with 93 percent accuracy, 98 percent Sensitivity, 90 percent Specificity, and 83 percent Precision and 90 percent F1-score ; where all the implemented ML methods also perform excellently. Only LR performs relatively poorly with accuracy of less than 80%.

In Figure14, we tested our framework on a sample of past road accidents to evaluate its effectiveness in reducing the risk of traffic accidents by using equations (1) and (2) with (N= -100), running it through 50 epochs. We started by evaluating each model individually and then combined the two models to demonstrate the importance of their collaboration. As shown in Figure 14 (a). The red curve in the graph shows the decrease in the risk rate of road accidents when only *V.Edge_DL* was applied. The risk value decreased from 7000 to 2200 over time; in contrast, when *Cloud_DRL* was used, the risk value decreased further to 1600, as shown by the orange curve. However, the best solution for reducing traffic accident risk was achieved by combining the two models, as demonstrated by the blue curve. With their collaboration, the risk level decreased to almost zero (800). The same objective is represented by the histogram in Figure 14 (b), which shows the level of traffic safety that has been achieved by each model individually and by their collaboration.

Our research project faced several limitations, particularly regarding the quality of historical data and the complexity of machine learning techniques. The availability of high-quality historical data was a significant challenge, as it affected the accuracy and reliability of our results. Additionally, the complexity of implementing machine learning algorithms posed difficulties in achieving optimal performance. Despite these hurdles, we were able to achieve commendable results and contribute positively to the field of road safety.

6 Conclusion

Smart Cities provide a range of capabilities that can enhance the daily lives of residents. One crucial application of Intelligent Transportation Systems (ITS) is the improvement of road safety. The prediction of traffic accident risk plays a vital role in achieving this objective, a collaborative Cloud-V.Edge driver assistance system (ICEDAS) that utilizes machine learning based IOV that has been proposed to address this challenge. To leverage the advantages and mitigate the drawbacks of both platforms, the proposed framework includes two models. The first, Cloud_DRL with accuracy of 94%, utilizes a substantial amount of crash data stored in the cloud. It also suggests various preventive actions, including stopping, decelerating, or not changing lanes in cases of negligible risk. The second model, V.Edge_DL with an accuracy of 93%, functions as an assistant, compensating for Cloud_DRL's lack of prediction due to issues such as internet disconnection or bandwidth overload. To evaluate the effectiveness of the framework to reduce the risk of accidents, we conducted tests on a randomly selected sample of past road accidents running it through 50 epochs. The results obtained indicate that the collaboration between the two models significantly reduces the risk (from 7000 to less than 800), about 90% for the selected sample, surpassing the performance of either model alone.

The future work would involve integrating computer vision into the current project, using machine learning and Cloud-Edge Computing. This approach will develop advanced systems to prevent accidents and enhance road safety by analysing images and videos to detect hazardous situations, identify risky behaviors, and assess real-time road conditions. This holds great promise for further development, particularly in the field of self-driving vehicles. Additionally, given the positive results achieved in road safety, the system can find applications in other areas like fire detection and industrial risk management.

.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, and X. Zheng, "Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," ArXiv preprintarXiv: 1603.04467, 2016.

[2] M. Al-Ayyoub, A. Nuseir, K. Alsmearat, Y. Jararweh, and B. Gupta, "Deep Learning for Arabic NLP: A Survey," *Journal of Computational Science*, 26:522-531, 2018, https://doi.org/10.1016/j.jocs.2017.11.011.

[3] S. Aradi, "Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 23(2):740-759, 2020, Doi: 10.1109/TITS.2020.3024655.

[4] N. M. Ashraf, R. R. Mostafa, R. H. Sakr, and M. Z. Rashad, "Optimizing Hyperparameters of Deep Reinforcement Learning for Autonomous Driving Based on Whale Optimization Algorithm," Plos one, 16(6):e0252754, 2021, https://doi.org/10.1371/journal.pone.0252754.

[5] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, "Autonomous Braking System via Deep Reinforcement Learning," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 1-6, October 2017, Doi: 10.1109/ITSC.2017.8317839

[6] L. Y. Chang and W. C. Chen, "Data Mining of Tree-Based Models to Analyse Freeway Accident Frequency," *Journal of Safety Research*, 36(4):365-375, 2005. https://doi.org/10.1016/j.jsr.2005.06.013.

[7] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning Deep Representation from Big and

Heterogeneous Data for Traffic Accident Inference" Proceedings of the AAAI Conference on Artificial Intelligence, 30(1):338-344, February 2016, https://doi.org/10.1609/aaai.v30i1.10011.

[8] François Chollet, Keras, 2015. https://github.com/fchollet/keras.

[9] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive Traffic Signal Control: Deep Reinforcement. Learning Algorithm with Experience Replay and Target Network," arXiv preprint arXiv:1705.02755, 2017, https://doi.org/10.48550/arXiv.1705.02755.

[10] F. A. González and C. A. Pedraza, "Deep Learning Ensemble Model for the Prediction of Traffic Accidents Using Social Media Data," *Computers*, 11(9):126, 2022, https://doi.org/10.3390/computers11090126.

[11] M. Gregurić, M. Vujić, C. Alexopoulos, and M. Miletić, "Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data," Applied Sciences, 10(11):4011, 2020, https://doi.org/10.3390/app10114011.

[12] M. Hossain, and Y. A. Muromachi, "Bayesian Network Based Framework for Real-Time Crash Prediction on the Basic Freeway Segments of Urban Expressways," *Accident Analysis & Prevention*, 45:373-381, 2012, https://doi.org/10.1016/j.aap.2011.08.004.

[13] M. G. Karlaftis and E. I. Vlahogianni, "Statistical Methods Versus Neural Networks in Transportation Research: Differences, Similarities and Some Insights," Transportation Research Part C: Emerging Technologies, 19(3):387-399, 2011, https://doi.org/10.1016/j.trc.2010.10.004.

[14] J. H. Kiswani, S. M. Dascalu, and F. C. Harris Jr, "Cloud Computing and Its Applications: A Comprehensive Survey," *International Journal of Computer Applications*, 28(1):3-24, 2021.

[15] S. Koh, B. Zhou, H. Fang, P. Yang, Z. Yang, Q, Yang, and Z. Ji, "Real-Time Deep Reinforcement Learning based Vehicle Navigation," *Applied Soft Computing*, 96:106694,.2020, https://doi.org/10.1016/j.asoc.2020.106694, Get rights and content.

[16] R. Kumar Thangavel, S. Athithan, S. Sarumathi, M. Aruna, and B. Nithila, "Blackspot Alert and Accident Prevention System." 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1-6, 2019, Doi: 10.1109/ICCCNT45670.2019.8944412.

[17] M. F. Labib, A. S. Rifat, M. M. Hossain, A. K. Das, F. Nawrine, "Road Accident Analysis and Prediction of Accident Severity by using Machine Learning in Bangladesh," 2019 7th International Conference on Smart Computing & Communications (ICSCC), IEEE, pp. 1-5, 2019, Doi: 10.1109/ICSCC.2019.8843640.

[18] L. Lin, Q. Wang, and A. W. Sadek, "A Novel Variable Selection Method based on Frequent Pattern Tree for Real-Time Traffic Accident Risk Prediction," Transportation Research Part C: Emerging Technologies, 55:444-459, 2015, https://doi.org/10.1016/j.trc.2015.03.015.

[19] B. Liu, Y. Tang, Y. Ji, Y. Shen, YA Du, Deep Reinforcement Learning Approach for Ramp Metering based on Traffic Video Data," *Journal of Advanced Transportation*, 1-13, 2021, https://doi.org/10.1155/2021/6669028.

[20] Y. Lv, S. Tang, and H. Zhao, "Real-Time Highway Traffic Accident Prediction based on the k-Nearest Neighbor Method" 2009 International Conference on Measuring Technology and Mechatronics Automation, IEEE, 3:547-550, April 2009, Doi : 10.1109/ICMTMA.2009.657.

[21] Y. Ma, Z. Liu, and C. McAllister, "Deep Reinforcement Learning for Portfolio Management," *International Journal of Computers and Their Applications*, 30(2):93-106, June 2023.

[22] R. Meneguette, R. De Grande, J. Ueyama, G. P. R. Filho, and E. Madeira, "Vehicular Edge Computing: Architecture, Resource Management, Security, and Challenges," *ACM Computing Surveys (CSUR)*, 55(1):1-46, 2021, https://doi.org/10.1145/3485129.

[23] A. Murad, F. A. Kraemer, K. Bach, and G. Taylor, "Information-Driven Adaptive Sensing based on Deep Reinforcement Learning," *Proceedings of the 10th International Conference on the Internet of Things*, pp. 1-8, October 2020, https://doi.org/10.1145/3410992.3411001.

[24] J. Paul, Z. Jahan, K. F. Lateef, M. R. Islam, and S. C. Bakchy, "Prediction of Road Accident and Severity of Bangladesh Applying Machine Learning Techniques," 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC), IEEE, pp. 1-6, December 2020, Doi: 10.1109/R10-HTC49770.2020.9356987.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and E. Duchesnay, "Scikit-Learn: Machine learning in Python," *Journal of Machine Learning Research*, 12:2825-2830, 2011.

[26] Z. Peng, S. Gao, Z. Li, B. Xiao, and Y. Qian, "Vehicle Safety Improvement Through Deep Learning and Mobile Sensing," *IEEE Network*, 32(4):28-33, 2018, Doi: 10.1109/MNET.2018.1700389.

[27] S. Ravichandiran, "Deep Reinforcement Learning with Python: Master Classic RL, Deep RL, Distributional RL, Inverse RL, and more with OpenAI Gym and TensorFlow," Packt Publishing Ltd, 2020.

[28] H. Ren, Y. Song, J. Liu, Y. Hu, and J. Lei, "A Deep Learning Approach to the Prediction of Short-Term Traffic Accident Risk," arXiv preprint arXiv:1710.09543, 2017.

[29] H. Ren, Y. Song, J., Wang, Y. Hu, J. Lei, "A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 3346-3351, November 2018, Doi: 10.1109/ITSC.2018.8569437.

[30] C. Spatharis and K. Blekas, "Multiagent Reinforcement Learning for Autonomous Driving in Traffic Zones with Unsignalized Intersections," *Journal of Intelligent Transportation Systems*, pp. 1-17, 2022 https://doi.org/10.1080/15472450.2022.2109416.

[31] H. Tatsat, S. Puri, and B. Lookabaugh, "Machine Learning and Data Science Blueprints for Finance," O'Reilly Media, 2020.

[32] A. Thaduri, V. Polepally, and S. Vodithala, "Traffic Accident Prediction Based on CNN Model," 2021 5th

International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, pp. 1590-1594, 2021, Doi: 10.1109/ICICCS51141.2021.9432224.

[33] A. Tolba, O. Said, and Z. Al-Makhadmeh, "MDS: Multi-Level Decision System for Patient Behavior Analysis based on Wearable Device Information," *Computer Communications*, 147:180-187, 2019, https://doi.org/10.1016/j.comcom.2019.08.022.

[34] M. Usama, B. Ahmad, J. Yang, S. Qamar, P. Ahmad, Y. Zhang, and J. Guna, "REMOVED: Equipping Recurrent Neural Network with CNN-Style Attention Mechanisms for Sentiment Analysis of Network Reviews," *Computer Communications*, 148:98-106, 2019, https://doi.org/10.1016/j.comcom.2019.08.002.

[35] S. Wan, Z. Gu, and Q. Ni, "Cognitive Computing and Wireless Communications on the Edge for Healthcare Service Robots," *Computer Communications*, 149:99-106, 2020. https://doi.org/10.1016/j.comcom.2019.10.012.

[36] C. Yang, Y. Wang, S. Lan, L. Wang, W. Shen, and G. Q. Huang, "Cloud-Edge-Device Collaboration Mechanisms of Deep Learning Models for Smart Robots in Mass Personalization," *Robotics and Computer-Integrated Manufacturing*, 77:102351, 2022, https://doi.org/10.1016/j.rcim.2022.102351.

[37] H. Yang, Y. Wang, H. Zhao, J. Zhu, and D. Wang, "Real-Time Traffic Incident Detection using an Autoencoder Model," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 1-6, 2020, Doi: 10.1109/ITSC45102.2020.9294455.

[38] L. Yu, B. Du, X. Hu, L. Sun, W. Lv, and R. Huang, "Traffic Accident Prediction Based on Deep Spatio-Temporal Analysis," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, IEEE, pp. 995-1002, August 2019, Doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00195.

[39] Z. Yuan, X. Zhou, and T. Yang, "Hetero-convlstm: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 984-992, July 2018. https://doi.org/10.1145/3219819.3219922.

[40] Z. Zhang, Q. Nie, J. Liu, A. Hainen, N. Islam, and C. Yang, "Machine Learning based Real-Time Prediction of Freeway Crash Risk using Crowdsourced Probe Vehicle Data," *Journal of Intelligent Transportation Systems*, pp. 1-19, 2022, https://doi.org/10.1080/15472450.2022.2106564.

[41] Z. Zhu and H. A. Zhao, "Survey of Deep RL and IL for Autonomous Driving Policy Learning," *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14043-14065, 2021, Doi: 10.1109/TITS.2021.3134702.

Zeroual Djazia (photo not available) is a researcher in the Computer Science Department at Biskra University in Algeria. She received her master's degree from Batna University, also in Algeria, with a specialization in Image Parallel Processing. Currently, she holds the position of assistant professor. Her research interests primarily revolve around artificial intelligence, with a specialization in machine learning, internet of things, cloud computing, data mining, and data analytics.

Okba Kazar obtained his master's degree in 1997 from Constantine University in Algeria, specializing in the field of artificial intelligence. He later earned his PhD from the same university in 2005. He served as a Professor in the Computer Science Department at Biskra University and played a pivotal role in the establishment of the LINFI laboratory at the same institution. He is a prolific author with numerous publications to his name, and serves as a member of the editorial board for several journals. His research interests encompass a wide range of areas, including multi-agent systems and their applications, ERP, advanced information systems, web services, semantic Web, big data, robotics, internet of things, and cloud computing. Currently, he holds the position of professor at the Information Systems and Security Department at the United Arab Emirates University in the UAE.

Saad Harous Senior Member, IEEE received the Ph.D. degree in computer science from Case Western Reserve University, Cleveland, OH, USA, in 1991. He is currently the Chair and a Professor with the College of Computing and Informatics, University of Sharjah, Sharjah, UAE. He has more than 30 years of experience in three different countries (USA, Oman, and UAE). He has published more than 200 journal and conference papers. His research interests are in distributed computing, technology-based learning, fault tolerant computing, and communication networks.

Saber Benharzallah is a professor and researcher in the Computer Science Department of Batna 2 University (Algeria). He received his Ph.D. degree in 2010 from the Biskra University (Algeria). Prof. Benharzallah is currently director of laboratory LAMIE (Batna 2 University). His research interests include internet of things, service-oriented architecture, and context aware systems.