Efficient Cloud Data Transfer via Prediction and Selective Data Dropping in Three-Tier Distributed Cloud Architecture for Smart Traffic Management

MSc Research Project
MSc in Cloud Computing
(MSCCLOUD1_JAN21)

Sonal Dipak Patil
Student ID: X21120285

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sonal Dipak Patil
Student ID:	X21120285
Programme:	Masters in Cloud Computing **Year:** Jan 2022-23
Module:	Research Project
Lecturer:	Mr. Rashid Mijumbi
Submission Due Date:	15th December 2022
Project Title:	Efficient cloud data transfer via prediction and selective data dropping in a three-tier distributed cloud architecture for smart traffic management
Word Count:	6364 **Page Count: 21**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sonal Dipak Patil

Date: 15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Efficient cloud data transfer via prediction and selective data dropping in three-tier distributed cloud architecture for smart traffic management

Sonal Dipak Patil

X21120285

Abstract

The traffic management system is an important application in today's ever-growing vehicle era. With a large number of closed-circuit television (CCTV) cameras available as the Edge node, it is possible to get information about the traffic on roads using these cameras. Also, with the advancement of the latest automobile sensors, it is possible to get the data from the vehicle so that better traffic management can be achieved. In the current work, we propose using a combinational approach that takes information from CCTV cameras and automobile sensors, which allows for smart traffic management. Our focus is on reducing the amount of data transmitted from the sensor to the cloud for further computation. The solution is based on a three-tier architecture of Edge-Fog-Cloud. We could remove 95% of the data transmitted at the edge and roughly 99% of the data on the fog side without affecting accuracy. The overall error was less than 0.07 after applying an autoregression (AR) prediction algorithm to reduced time-series data. We expect that our work can be further expanded to achieve more data reduction. The proposed approach can be used in other application areas as well, such as hospital management, smart farming, etc.

1 Introduction

The advancement in technology makes it possible to develop an intelligent traffic system. These technological advancements are closed-circuit television (CCTV) cameras installed throughout the locality for the safety of the people. Also, sensors mounted inside the car for its health monitoring can be used for traffic management as well. Both of these technologies are very useful but underutilized. Hence, we propose a smart traffic management system that increases the usability of these two systems. Traffic management means re-routing the vehicles to their best-suited paths and helping them reach their destinations in minimal time.

The sensors, such as a global positioning system (GPS), accelerometer, barometer, gyroscope, and inertial measurement unit (IMU), are pre-installed inside the car. These sensors detect the movement of the car and can also be used to identify the traffic on the road.

The traffic in the world is going to increase day by day. This year, nearly 1.5 lakh more vehicles were sold than the previous year. This difference is increasing exponentially. With the upcoming electric vehicle industry, the cars on the road would reach roughly the 300 million mark by 2030 (Murphy, 2022). It is important to develop a smart traffic management system to accommodate this ever-increasing number of cars.

The current use of CCTV cameras is to merely monitor accidents and occasional robberies that happen within their surveillance range. Because of this, more than 99% of the frames are not being used. CCTV camera frames can be used for smart traffic management, especially those cameras that are facing the road.

The frame rate for these cameras is very high (25 fps), and the vehicles are slow (roughly 1 fps). It is necessary to skip the appropriate number of frames for traffic management. These frames need to be transmitted to the cloud so that global decisions can be made for traffic management rather than local ones. With sufficiently large frames dropped before transmitting the data, it should reach the destination faster, and such new vehicle information is not lost. If excess data is delivered to the cloud, then the cloud would remain occupied processing the excess frames. However, if a lower number of frames are delivered to the cloud than desired, the computation may go wrong and could eventually lead to deadly traffic collisions.

Today's newer vehicles are equipped with a variety of sensors, including a 6-D IMU, fuel level monitor, temperature sensor, and others (Sellén and Sellén, 2022). The information about the car is exclusive to that vehicle. However, the proposed system may require data to be uploaded to the cloud.

The concept proposed in this thesis is to reduce the amount of data transmitted from sensors to the cloud so that data arrives at the cloud in the shortest amount of time possible. Also, the data has a lot of redundancy, which when sent to the cloud won't help make better decisions in the cloud. The proposed study is conducted in three parts. The first part is executed on the edge node. In this part, we remove excess frames present in the CCTV camera footage that has moving vehicles in it. The main decision is made based on the number of vehicles available in the frame and how fast the vehicles are moving.

The second part is to be executed on the fog layer, where two main variables appear simultaneously. The first data is from CCTV camera edge nodes that send the minimum number of required frames that can identify the total vehicular population on the road at any given point in time. The second data is the sensor reading available from different cars at the same junction. The third part is executed in the cloud, where we implemented the autoregression (AR)-based prediction and corresponding messages to the vehicle driver giving traffic alerts.

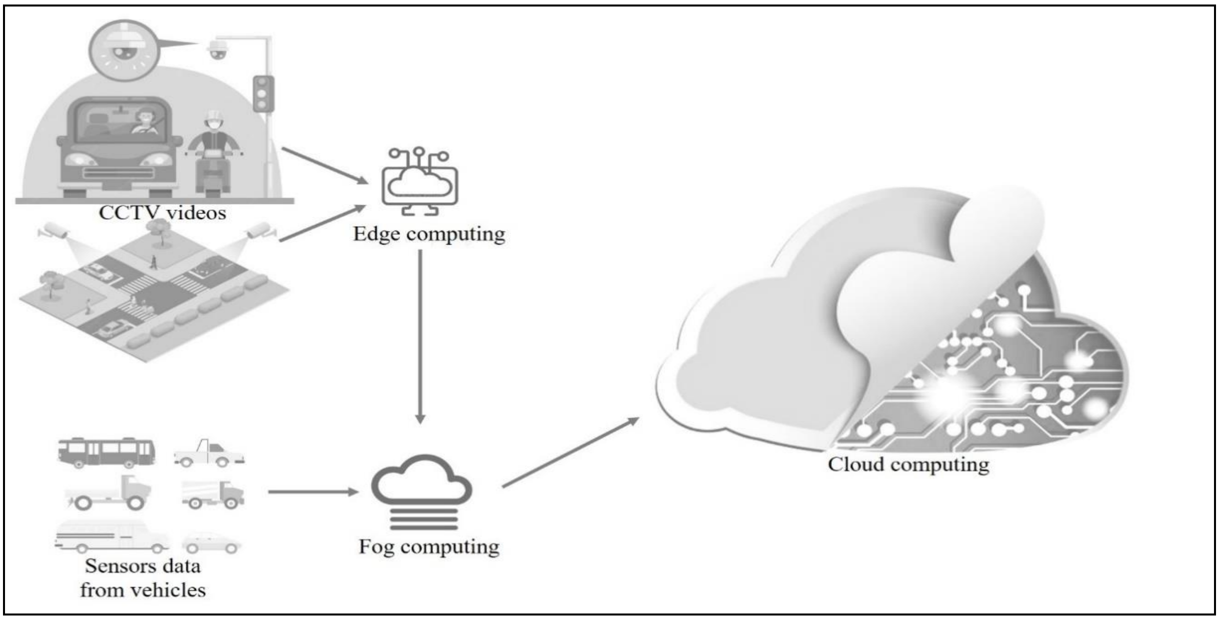

Figure 1: Schematic diagram of Proposed three-tier architecture for Smart Traffic Management.

Research Question: There is an ample amount of data produced from multiple sensors inside the vehicle to design a smart traffic management system. There is also a sufficient amount of CCTV camera footage. If we were able to drop some frames at the edge, that would speed up the entire computation. What is the appropriate percentage of frames that can be dropped? Similarly, what percentage of sensor data can be dropped in the fog node?

After interlacing the data in a fog node, that data is transmitted to the cloud. How much latency and bandwidth are there in the overall system? Once received in the cloud, how much is the error after using the prediction algorithm?

In comparison to the existing works, our work focuses on data reduction in a three-tier architecture with an AR prediction algorithm. As per the recent literature review, this is a unique combination that has not been proposed earlier. Using our proposed work, we will try to enhance accuracy and latency.

2 Related Work

2.1 Mathematical and Fuzzy logic based:

Guillen-Perez and Cano (2021) have focused on a traffic management system that could take smart decisions on traffic junctions. The parameters under consideration were waiting time, the average speed of the vehicle, car emissions from the vehicle, and overall trip time, whose practical applicability is tough. Compared to their method, we are planning to use sensor data as well as video feed obtained via CCTV footage, which will add more value to the optimization problem.

Raskar and Nema (2021) focused on a smart solution for barricade management using fuzzy logic that takes data from smart sensors, and barricade positions will be adjusted. The main drawback of this method is that it is impossible to make thousands of robotic barricades. Also, as fuzzy logic is based on rules created by humans, it might not give appropriate results in any new dynamic situation. To overcome the drawbacks in their existing system, our proposed solution will send the decisions to the human drivers and allow dynamic decision-making to play a role.

Liu and Ke (2022) have designed a traffic management system using an IoT-based smart sensor with cloud assistance for congestion avoidance. In our case, we are going to use on-vehicle sensors, but we will also be using video footage for cross-validation. Instead of their two-tier architecture, our three-tier architecture would bring flexibility, more computational ability, and dynamic decisioning.

Choudhary, Singh, and Anand (2022) have developed a multi-layer traffic management system that works on a two-tier wireless architecture having only fog and cloud. Our proposed method is built on this idea, except for the fact that we will also have edge computers with three-tier architecture. The key difference between their work and our solution is that they focus only on congestion control, whereas we focus on smart traffic management along with congestion control.

Fouladfar, Khayyambashi, and Solé Pareta (2021) have designed a complete cloud-based traffic management system that includes a 4-layer architecture, mainly cloud, fog, a local cloud-sublayer, and a sensor-things layer. The data that they used was a multimodal input with GPS position and CCTV footage. The only difference between their method and our proposed solution is that instead of using only GPS data, we will use inertial measurement unit (IMU) and odometry data. Their method uses a fixed number of elements in the network, whereas ours uses a dynamic number of elements.

2.2 Machine-vision based approaches:

Lei, Mohamed, and Claudel (2018) have designed road traffic management using an IMU sensor. Their program involved traffic trajectory estimation along with road conditioning monitoring features. Although their focus was not only to classify the road conditions but also the position of the vehicle on the road, our algorithm will use this information before deciding the final route. Our 3-tier architecture solution can provide the same results with more accuracy and speed.

Armawi et al. (2022) have used CCTV camera footage for traffic management strategies. Their focus was mainly on a quantitative study of the CCTV footage gathered from Semarang. For their study, they used 16 hours of CCTV footage. These video clips were recorded with a PTZ camera, and they applied a machine learning algorithm. Their entire work was focused on video classification, whereas we would also be working with the other sensor data.

Pereira, Melo, and Araujo (2022) have shown for the first time that a three-tier architecture can be directly applied to an intelligent traffic management system. Similar to their approach, we would like to propose a three-tier architecture system called Edge+Fog+Cloud. In our proposed method, we will merge the sensor data with the camera feed, and hence the possibility to achieve higher accuracy is greater as there are multiple cross-validations.

2.3 Machine learning and optimization based:

Zhang et al. (2021) have designed an IMU-based road surface detector. Their car also had LiDAR data and localization data. In our proposed method, we will be using sensor data in combination with the camera feed. Their prediction accuracy could reach 99.4%. Neelakandan et al. (2021) have designed an 80286-based system for the traffic signal controller. Their overall accuracy could reach up to 98.23% with a root mean square error of 0.345. Prasad et al. (2019) have designed a traffic density estimation system based on histogram-oriented gradients (HOG) and local binary patterns (LBP). They achieved an accuracy of 94.02% on a single computing machine. Lilhore et al. (2022) have used IoT-based smart vehicle systems for traffic management using adaptive machine learning, which focused on congestion avoidance by adjusting the timing of the green light in the traffic signal. Just like Liu's group, they have also focused on IoT sensor data with the Edge-Cloud computer architecture. We expect our solution on the Edge-Fog-Cloud computer will be able to perform a faster computation on the same system with more accuracy. In comparison to these works, we would improve the accuracy using our approach.

2.4 Deep learning based:

Algiriyage et al. (2020) have worked on an emergency traffic management system that uses CCTV footage as an input. Their major drawback was that all the computation was done on a local machine, and there was no consideration of using a single CCTV camera feed. On the other hand, we are trying to use multi-CCTV camera feeds. Hamami et al. (2020) have designed a CCTV-based traffic data analysis system using the Yolo v4 algorithm. They could reach the highest processing speed of 10 fps. We would be improving on their drawback of a low frame rate and trying to reach an optimal fps to easily classify fast-moving vehicles.

Fadullalah et al. (2018) have designed a deep learning-based large-scale heterogeneous intelligent traffic system. They focused on non-supervised deep learning given by deep belief neural networks (DBN networks). With DBN, they could reach a throughput of 20 GB/s with a loss factor of 54. We would like to improve their loss factor, and we might continue with the deep neural network rather than the DBN network.

Chavan et al. (2022) have designed a congestion management system for Indian road systems. In their review, they compared different technologies such as RFID, cloud, IoT, WiFi, and deep learning. They covered the majority of recently reported technologies, but not all of them. Our proposed research and similar work are not reported in their review as it is a unique combination of a three-tier architecture with deep learning and prediction.

Wang et al. (2018) have designed a fog-enabled real-time system that works with low latency and focuses on minimising the response time. They performed two-level data offloading via optimization of a single type of data, whereas we are working towards multimodal data offloading.

According to a review of related works, there hasn't been a method for achieving smart traffic management using a three-tier architecture consisting of edge fog and cloud using IMU sensor data and CCTV footage via data dropping. Also, once the data is received in the cloud, using the prediction algorithm, the original data can be reproduced. We would use our approach to focus and improve the accuracy of previous works.

3 Research Methodology

3.1 Data collection:

All the videos are obtained from www.Kaggle.com, which is an open-source dataset sharing website. The data collected in these videos is real CCTV footage obtained at a traffic intersection. Along with Kaggle, there were a few other datasets that were also available for CCTV road captures. Car Accident Detection and Prediction (CADP) and Data-world are two of these sources. The temperature and IMU data were also collected from the Kaggle dataset, with a total of more than 900 cars. The IMU readings typically consist of the type of inertial measurement unit, the sample rate of the sensor, the temperature of the IMU in degrees Celsius, the magnetic field vector in the local navigation coordinate system, the accelerometer sensor parameters, the gyroscope sensor parameters, the magnetic sensor parameters, and a random number that can be used as an initial seed.

3.2 Video pre-processing:

Since most of the collected video files can vary in terms of resolution, frame rate, bit depth, etc., it is necessary to convert them into a uniform video size and the same video properties throughout the database. This step is optional and can be avoided, as it is very unlikely to have different manufacturers provide the solution to the single traffic junction. Hence, most of the real-life videos will be collected from a similar video source, and all these CCTV cameras should typically be of the same make and manufacturer. For the purpose of this research project, this dataset was collected from different sources, and hence open-source software like image-J and K-lite Mega Codec Pack will be utilised for pre-processing.

3.3 Sensor data pre-processing:

Out of the different parameters obtained from IMU, the mainly used parameters are accelerometer sensor parameters (Gx, Gy, and Gz) and gyroscope sensor parameters (Angle x, angle y, angle z). The IMU sensor reading consists of Gx, Gy, and Gz acceleration components obtained from the accelerometer present inside the IMU, and angle x, angle y, and angle z obtained from the gyroscope present inside the IMU.

3.4 Frame removal at Edge node:

Multiple videos were analysed in a time-synchronized manner at the same time in an Edge node to detect excess frames. The data drop rate for individual videos was determined separately. The edge node then discarded the frames, which were not relevant.

3.5 Data processing at Fog:

The fog node analyses the IMU data acquired from the cars and removes the empty values as well as excess readings acquired from the sensors. The Gx, Gy, and Gz acceleration components and angles x, y, and z were determined from the IMU, and the rest of the values are discarded. Time-synchronized frames and their corresponding IMU data are interlaced before they are transmitted to the cloud. Once the data reaches the cloud, the prediction algorithm is used to reconstruct the original data for smart traffic management.

3.6 Multi-modal data:

The 1D sensor data is mainly collected in the car. The car itself, with its computer, acts as an edge node for IMU and temperature data. This data is transferred to the fog node for interlacing with the CCTV footage. Since the format of the data coming from the sensor is text, its size is much lower compared to the corresponding video frame. It is very important that each frame be in time synchronisation with the sensor data. Also, all the valid CCTV frames transferred to the fog node should be synchronised in time. The final data uploaded to the cloud is of the nature of a multi-modal signal, where each multi-modal packet consists of sensor data in text and picture data in the form of frames from different CCTV cameras.

3.7 Three-tier architecture:

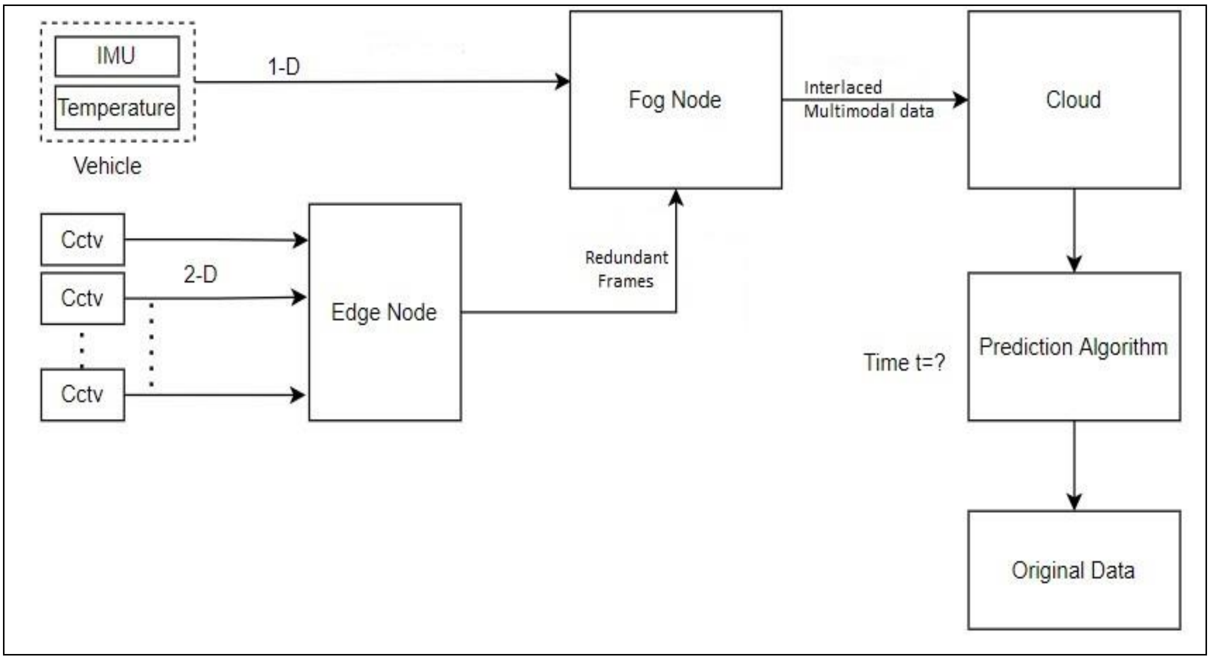

Figure 2 :Proposed system block diagram with three-tier distributed architecture.

Figure 2 shows the proposed system block diagram. The IMU and temperature readings are collected at the vehicle-edge node. The entire set of data is transferred to the fog node while the vehicle is moving within range of the fog node. All the CCTV footage is connected to an Edge node for the initial data reduction procedure. This operation must be performed at the edge because a large amount of data is transferred, resulting in a loss of bandwidth and a delay. The Fog node is responsible for multimodal data removal with time synchronization. The data received by the cloud is a multimodal packet from the fog. The cloud can then run a prediction algorithm and regenerate the original data at any given time. This can only be done if the prediction algorithm is continuously trained based on past training experiences. The overall aim of this three-tier architecture is to achieve better bandwidth utilisation with minimal delay.

3.8 Processing at the Edge Node:

The edge computer for CCTV is responsible for finding out the optimal frame rate that can be transmitted to the fog. This frame rate cannot be very high as it would increase the bandwidth requirement, nor can it be very low as information loss can occur. A simple frame subtraction can be used to calculate the amount of new information contained within two consecutive frames. The detection of key frames to be transmitted to the fog should have more stable information. The degree of blurry noise in the frame can be detected by edge detection algorithms, which can be included in future works.

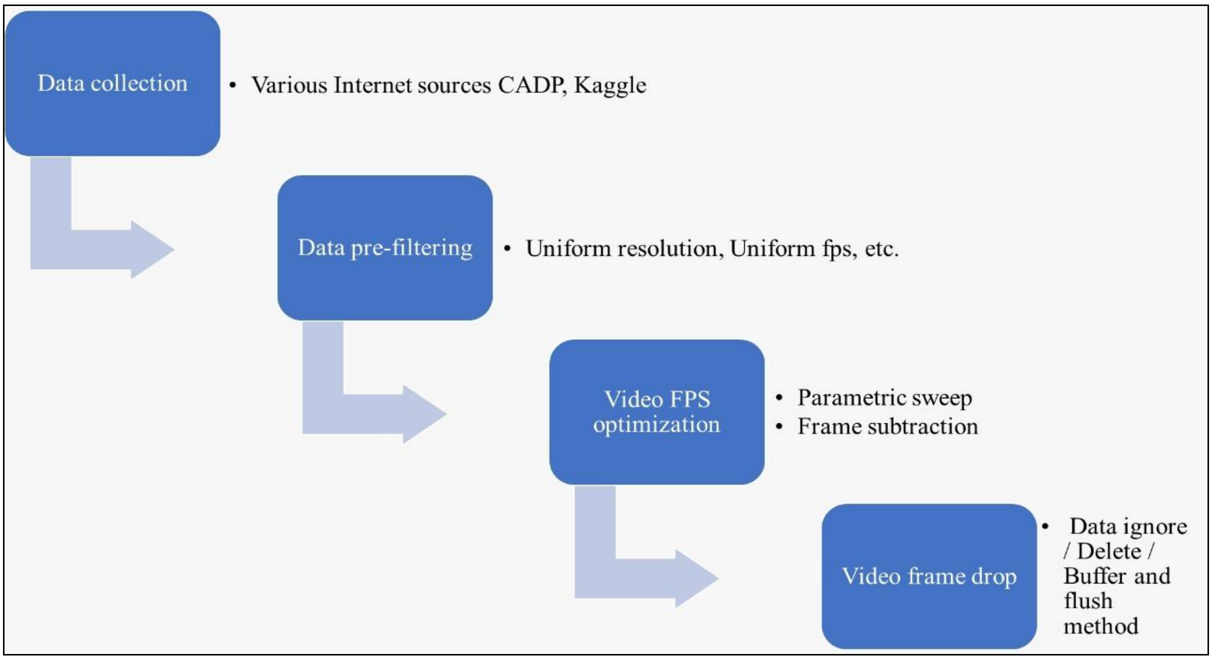

Figure 3: Steps followed at Edge Node

Figure 3 shows the data collection step followed by data prefiltering at Edge, followed by video FPS optimization, and finally frame drop. Video FPS optimization first requires selecting the proper keyframe. The key frame should have the most edges compared to its previous and next frames. Also, the subtraction between keyframes and their neighbours should not be very high, indicating noise. Frame subtraction is the simple logic on which the entire data compression process relies. As most of the consecutive frames will have redundant information, we plan to drop those frames, and this way the actual FPS will be lower than the original FPS while capturing. As the vehicles moving on the road are a dynamic event, the optimised FPS needs to be recomputed periodically so that there is no information loss. The video frame drop logic is to simply ignore the frames captured by the buffer and flush them without reading them. In simple words, it is like reducing the FPS of the recording dynamically. This idea was inspired by many algorithms in the literature. (Liljeblad, 2019) (Pawłowski, 2021)

Algorithm 1 Edge Computer Algorithm

Require: *CCTV video Data*
Require: *Acceptable Vehicle detection limit*
  $V(i, F) \leftarrow total(N)\ CCTV\ videoData$
  $AVDL \leftarrow Acceptable\ Vehicle\ detection\ limit$
  while $i\ != N$ **do**
    if $(V(f) - V(f+1)) < AVDL$ **then**
      $V(f) \leftarrow V(f+1)$
    else if $(V(f) - V(f+1)) > AVDL$ **then**
      Save V(f)
      Save i, f
    end if
  end while
  $Transmitt\ V(F, i)\ to\ Fog$

The CCTV video is fed into the Edge node as an input. We need to input the "acceptable vehicle detection limit," which determines the number of vehicles that are allowed to be dropped from consecutive frames. Each CCTV footage must be checked for repeated frames in the while loop for all of the CCTV footage. Hence, V(f)-V(f+1) is taken, and if it is less than the AVDL, the V(f+1) is considered as V(f), and the old frame is discarded. Using similar logic, the new redundant video is extracted and is sent to the fog node.

3.9 Processing at Fog:

The data arriving from Edge computers will consist of reduced-framerate CCTV footage and textual data available from car sensors. The next step is to clear the non-usable values during data pre-filtering. This is done by discarding the partially received data. The data drop rate is decided based on repetitions and deterministic value identification. Finally, data interlacing is carried out at the fog computer, where CCTV footage from all different cameras is merged with textual data from the sensors. For deterministic value identification, we will use interpolation and curve fitting. The decision to discard the value will depend on the root mean square error (RMSE).

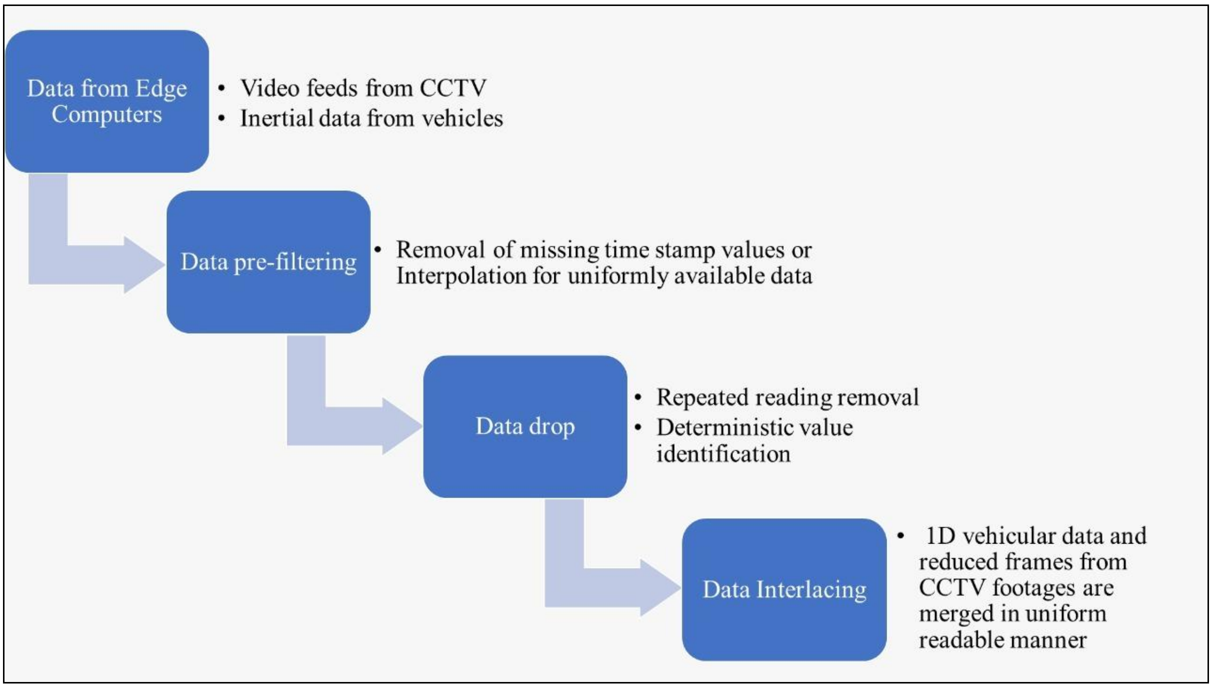

Figure 4 Steps followed at Fog Node

Algorithm 2 Fog Computer Algorithm

Require: $V(F, i)$
Require: $IMU\ data$
Require: $GPS\ data$
Require: $M\ =\ Total\ Vehicles$
  $G() \leftarrow GPS$
  $I(x..) \leftarrow IMU\ data$
  $k = 1$ index for GPS data
  $l = 1$ index for IMU data
  while $i\ != M$ **do**
    if $(G(Lt, Lg, i)! = G(Lt, Lg, i)$ **then**
      $k \leftarrow k + 1$
      $(G(Lt, Lg, k) \leftarrow (G(Lt, Lg, i)$
    end if
    if $(I(x..)! = I(x + 1..))$ **then**
      $l \leftarrow l + 1$
      $(I(x.., l) \leftarrow (I(x.., i)$
    end if
    $H(i) \leftarrow V(F, i, :), G(Lt, Lg, k), I(x.., l)$
  end while
  $Transmitt\ H(i)\ to\ cloud$

In the Fog node, the redundant video is received along with the sensor data from IMU. Both information are interlaced using an Interlacing algorithm and the hybrid frame is sent over to the Cloud.

3.10 Processing at Cloud:

The data received from the fog computer is separated into 1D and 2D data. Also, data prefiltering is carried out to remove any redundancy caused by the partially received packet. The data reconstruction is carried out in the cloud on demand. Here, linear interpolation, differential frames, etc., will be used. The minimum distance decoding allows us to evaluate the performance of the proposed smart traffic management system. The proposed method's overall error should be within acceptable limits.

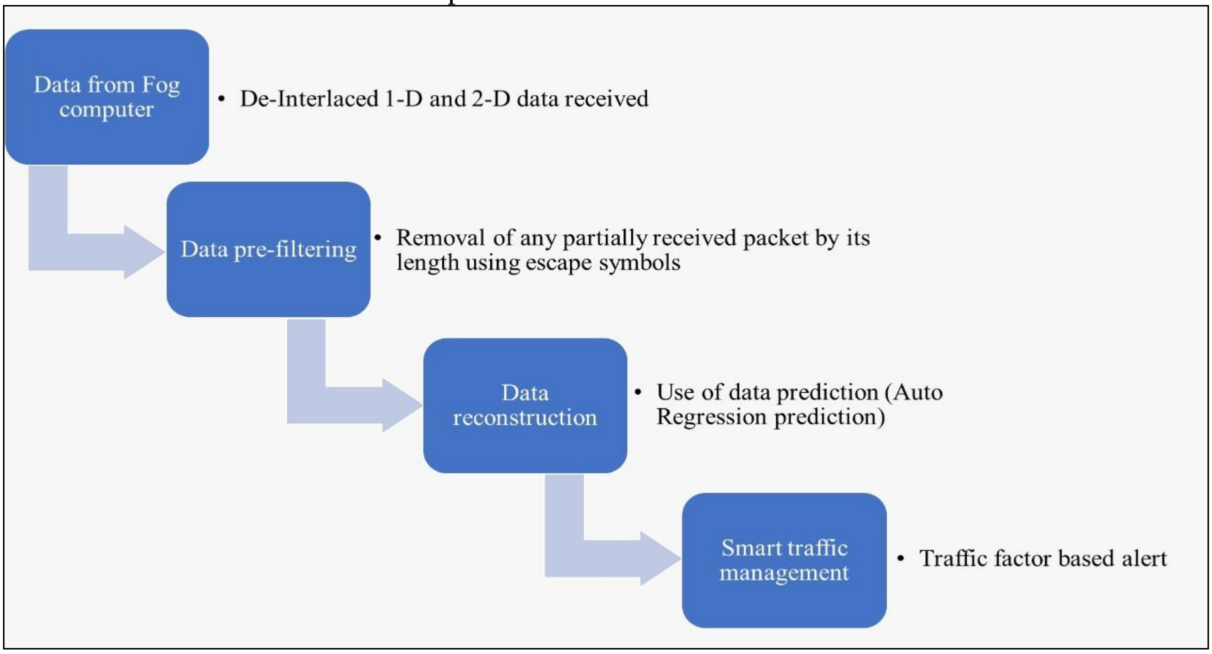

Figure 5 Steps followed in Cloud

Algorithm 3 Cloud Computer Algorithm

Require: $H(i)$
Require: $N = Total\ frames$
Require: $M = Total\ readings$ Hybrid data
  while $i\ != N * M$ **do**
    Reproduce data
  end while
  Prediction algorithm

The frames and sensor data are collected once the data is received in the cloud. The reproduction algorithm is then applied, and if an appropriate accuracy is reached, the data drop is set. Using this dataset, then, traffic management decisions are made.

4 Implementation

4.1 Dataset:

The dataset gathered from an internet source contains acceleration and gyroscopic data for 999 vehicles. The other data obtained from the dataset, such as GPS coordinates or engine temperature, is ignored for the current study. The corresponding CCTV footage was obtained with a resolution of 720x1280 pixels and an RGB color frame.

The code was developed using the MATLAB programming language for all three tiers of the proposed distributed network. A student account is created on MathWorks.com to get access to the MATLAB R2021b version. To run the simulations, an Intel Core i5, 11th generation, with a 3.10 GHz clock speed and 8GB of DDR4 RAM was used. A NVIDIA MX450 GPU with 2GB of graphics memory (GDDR5) was used for accelerated computation. All the simulations were run on a 512GB SSD with the Windows 11 operating system. For all the graphs, Microsoft Office was used. Data analysis was performed on MS Excel, whereas the diagrams were designed using MS PowerPoint.

4.2 Implementation of Algorithm on Edge Computer:

Edge computers are the CCTV cameras with minimal processing power that can manipulate videos during live streaming. This manipulation is done to achieve redundancy removal at the source itself. For the current simulation, instead of a live camera feed, we have taken pre-recorded video. To simulate the frame grabber, all the frames in the video were stored in the C driver's temp folder as an image file. To validate the successful frame grabber operation, the first frame acquired is reloaded into the buffer memory. This frame also gives information on the image resolution and its color information. Also, the total number of frames in the frame grabber allowed us to estimate the total data size that could have been taken without any data reduction algorithm. The simplest operation we propose at the edge level is to read consecutive frames and detect complete repetition by using frame subtraction. Since the data is in unit-8 format, its data type needs to be changed to float or double before performing mathematical subtraction. The entire frame difference is recorded into a variable to simulate the real-time scenario using the formula:
imdiff=abs(double(im)-double(im2))

The maximum difference between the frames is computed, and since the input video is CCTV footage, a complete scene change is not possible. For the frames that cause a large fluctuation, i.e., the frame difference is more than 70% from the normalised maximum frame difference allowed, then that frame is transmitted, or else it is dropped. The total number of frames transmitted to the fog and the total number of frames that are discarded are used to compute the data drop rate at the edge computer.

DD=1-(count/T)
where count indicates the total number of frames sent to the fog, DD indicates the data drop rate, and T indicates the total number of frames arrived at the CCTV (edge).

4.3 Implementation of Fog Algorithm:

All the images received from the edge were stored in the buffer. Just to simplify the computation during the development, we have reduced the image size to one-tenth of its original size in each dimension, i.e., the total number of frames to be transmitted from the fog to the cloud is of the size 72x128 instead of 720x1280. All the images received from the Edge were now converted into a single-row array of size 72x128. Also, their dimensions are stored as the first 2 bytes of the interlaced dataset to be sent to the cloud. After multiple experiments with various-length videos, we found that roughly 3% of the frames were transmitted, and hence every 34th frame can be assumed to be the new information frame. Simultaneously, the role of the fog computer is to get the raw data from the cars' sensors (which can be considered as edge nodes). This proposed algorithm does not focus on dropping the raw textual sensor data, as it might vary from car to car depending on its manufacturer. Also, many times, since the car is a moving vehicle, an invalid or damaged data set occurs, which provides a natural dump of the information. This natural dump is dependent on the car and the transmitter installed within the car. For the dataset, we used the car IMU temperature dataset with 999 cars and 31 sensors. Out of the different readings for traffic management, mainly six readings were crucial, which are Anglex, Angley, and Anglez, which are angles computed via gyroscope, and Gx, Gy, and Gz, which are angles computed via accelerometer. To filter out the incorrect angle readings, any value other than 0 to 360 was replaced with an invalid indicator.

For our convenience, we chose the invalid indicator to be 1200, as none of the angle values could ever reach that number. For the accelerometer, all the missing readings were also replaced with the same invalid indicator so that filtering would be easier. For each of the angle readings, repeat values were scanned and ignored along with their timestamp. If the current angle ($Anglex_{x}$) is not the same as the previous angle ($Anglex_{x-1}$), then the previous angle (Anglexx-1) is transmitted along with its timestamp, and the same variable space is replaced with the current value ($Anglex_{x}$). And if a new value comes, then the updated previous value is compared. A similar process is repeated for all three accelerometer readings. Once all the filtering is complete, the data is sent to the cloud using an interlacing model. The interlacing order has a starting character followed by three accelerometer readings, three gyroscope readings, and then the frames received from the edge. Along with the start bit, there are five more validation bits in between to ensure the proper synchronisation of the data while decoding.

4.4 Implementation of Cloud Algorithm:

The data received on the cloud from the fog is scanned for the start bit. Once the start bit is received, all the data recovery starts with gyroscope readings with two validations, the accelerometer reading with two validations, and frames received with 1 validation. For proper interlacing, all the images were transmitted in the form of a time sequence; hence, at the receiver, it is mandatory to convert 1D frame data to 2D frames. The information about the size of the image is transmitted from the fog along with the interlaced data itself.

4.5 Prediction Algorithm:

For forecasting the data, an auto-regression (AR) model is used. The forecast factor in AR was set to 4, and K points were forecasted. This way, the error is computed between the true value and the forecasted value. During the training of the network, some of the values from the known database are filtered so that the accuracy can be verified with respect to the forecast factor of 4. The entire goal of this cloud algorithm is to minimise the power while maximising the coverage.

4.6 Traffic management:

To determine the condition of the traffic, we have analysed accelerometer data. We defined the traffic factor as per the below equation.

Traffic factor = ABS(G_x(i) / max(G_x))

where Gx(i) is the current accelerometer reading in the X direction and max(Gx) is the accelerometer's maximum value in the X direction.

If the traffic factor is greater than 0.2, then an alert is generated for the user, indicating a traffic jam.

5 Evaluation

5.1 Case Study 1

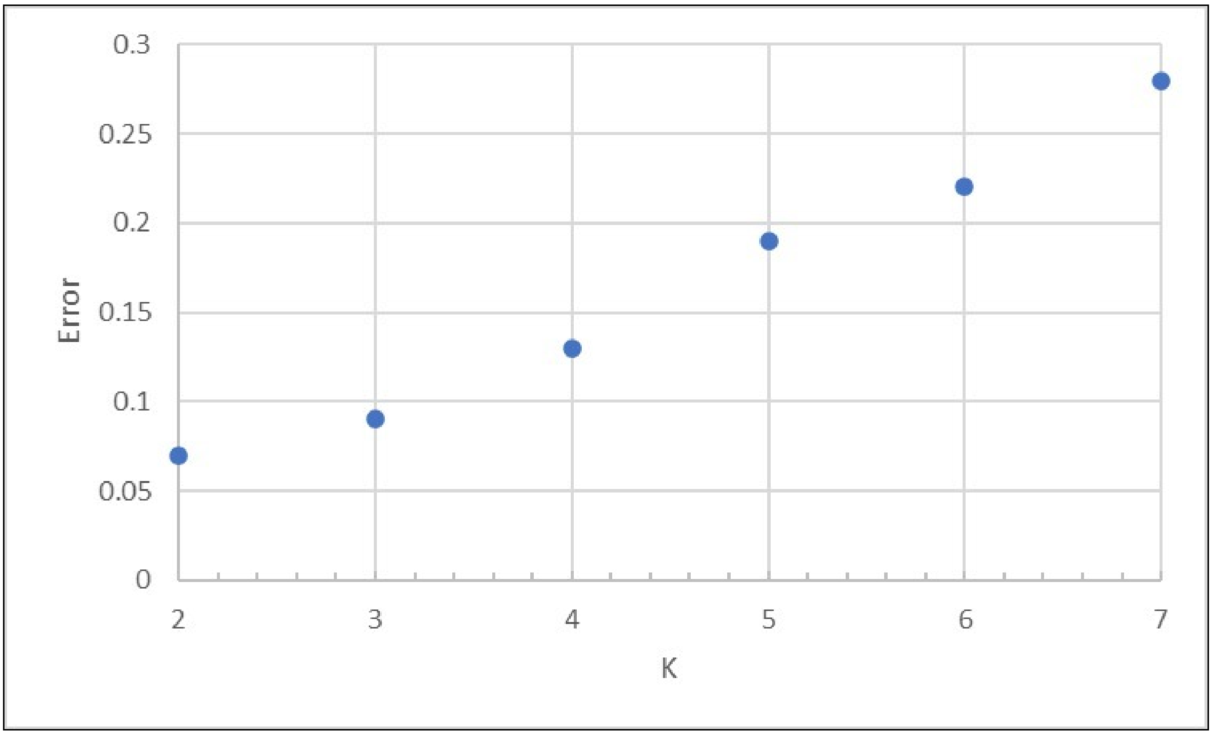

Figure 6 The effect of K number of predictions on the Percentage error. As the number of K increases the output error also increases.The minimum error occurs at K=2.

Figure 6 shows how the prediction values are affected by the number of predictions (K). For the current experiment, we considered this K going from 2 to 7 and found that the error keeps on increasing as the number of predictions increases. To achieve minimum error, predictions are made using the current value, and hence only two predictions are sufficient. Also, while using the real-time data, we found that, irrespective of the video, having two predictions gives minimal error. Hence, for our proposed AR predictor, the AR factor should be 5 and the total number of predictions should be 2 to achieve the best results.

5.2 Case Study 2

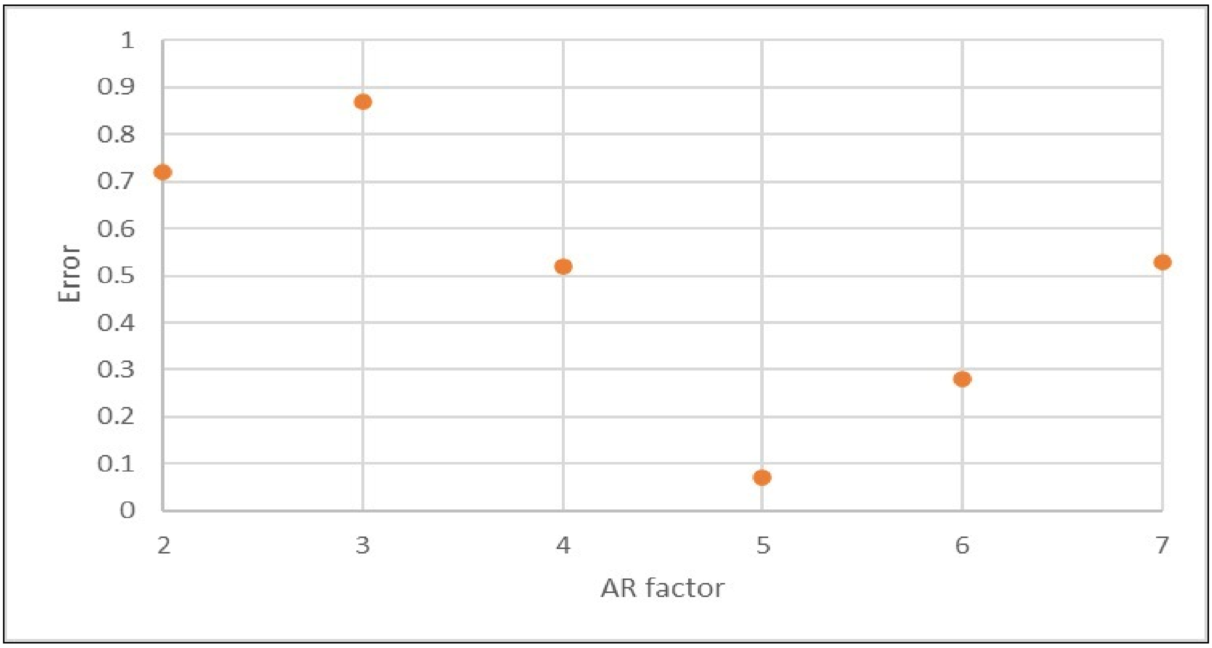

Figure 7 Effect of AR factor variation on the Prediction Algorithm Output

Figure 7 depicts the error introduced in the prediction algorithm output as a result of AR factor variation. The AR factor can be varied from 2 to 7 and beyond. As the value increases from 2, the error increases slightly and then starts decreasing after a certain point. The error is minimal at AR=5 with a value of 0.07. After that, the error starts increasing again. Hence, we decided to keep the AR factor at 5. The AR factor basically decides how many previous values the model uses to predict the new value. Here, the model uses the 5 previous values in the series to predict the new value with the minimum error.

5.3 Case Study 3

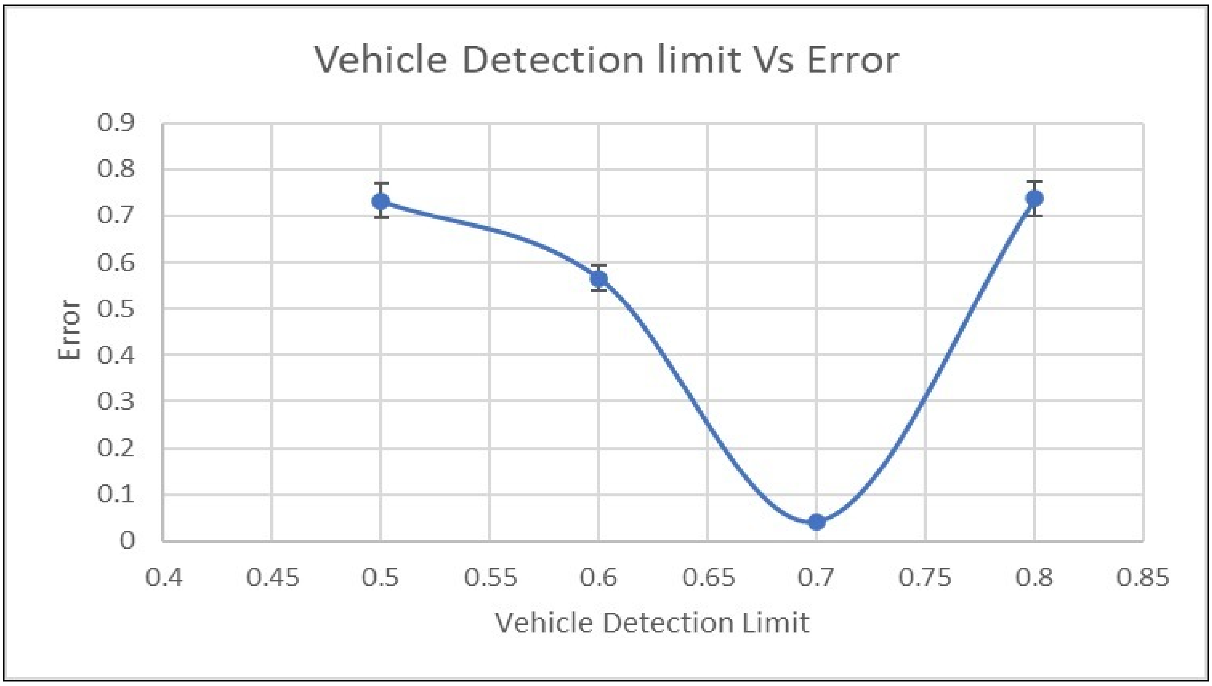

Figure 8 Vehicle Detection limit vs Error

Figure 8 shows the error introduced by varying the vehicle detection limit. As the number of frames transmitted from the edge to the fog is dependent on the vehicle detection limit, it was important to perform an experiment that could identify the ideal value between 0 and 1. For our experiment's purpose, we chose a vehicle detection limit of 0.5 and then gradually increased it to 0.8 in steps of 0.1. It was observed that the higher the value of the vehicle detection limit, the lesser the error was until a certain value (0.7), beyond which the error increased, hence it is very important to set proper vehicle detection. The error is defined as the difference between the number of frames transmitted and the actual number of frames. For ideal conditions, one frame per vehicle is the bare minimum.

5.4 Case Study 4

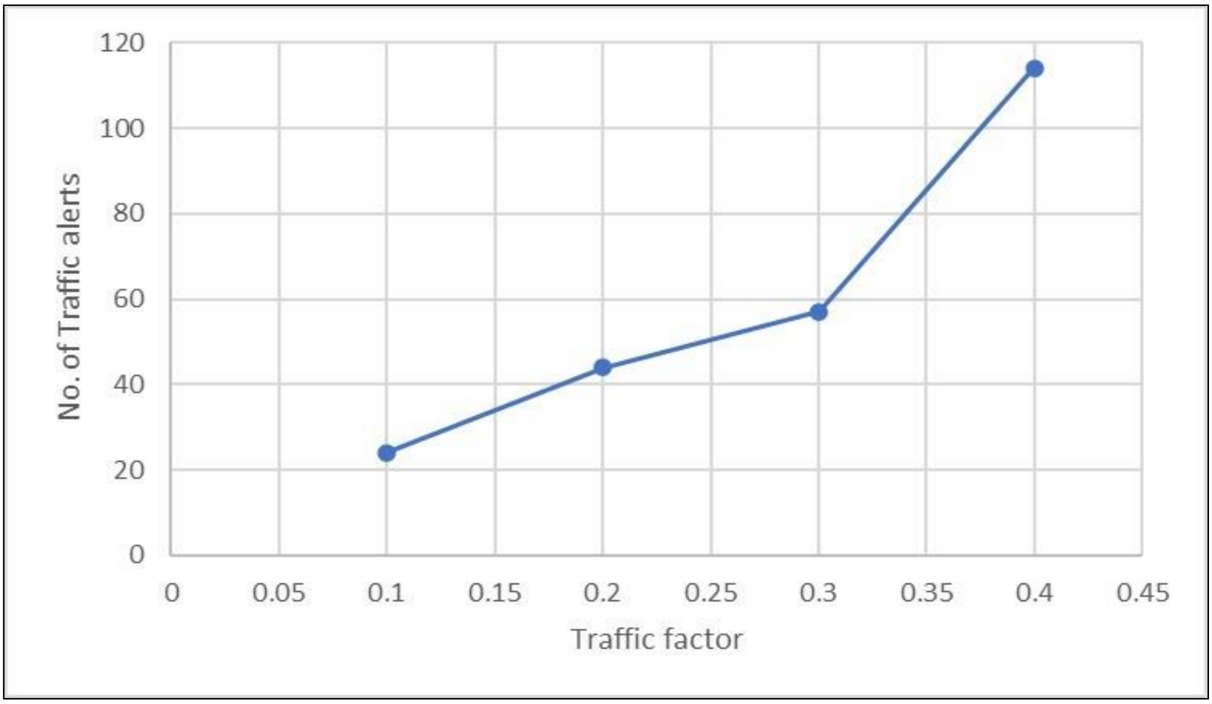

Figure 9 Effect of Traffic factor on the Number of Traffic Alerts generated.

The traffic factor is defined as the ratio between the current value and the maximum value for that particular sensor. Figure 9 shows that as the traffic factor on the road increases from 0.1 to 0.4, the corresponding number of alerts also increases. For the traffic management system, it is important that the person sitting in the vehicle receive a timely alert so that a less-travelled route can be selected. As the traffic factor increases from 0.1 to 0.4, the number of alerts generated also increases, indicating more traffic. As the traffic factor depends on the number of vehicles, it can vary from 0 to 1. The smaller the traffic factor, the easier it is to maintain the traffic. We also observed a non-linear relationship between traffic factors and the corresponding alerts generated.

5.5 Case Study 5

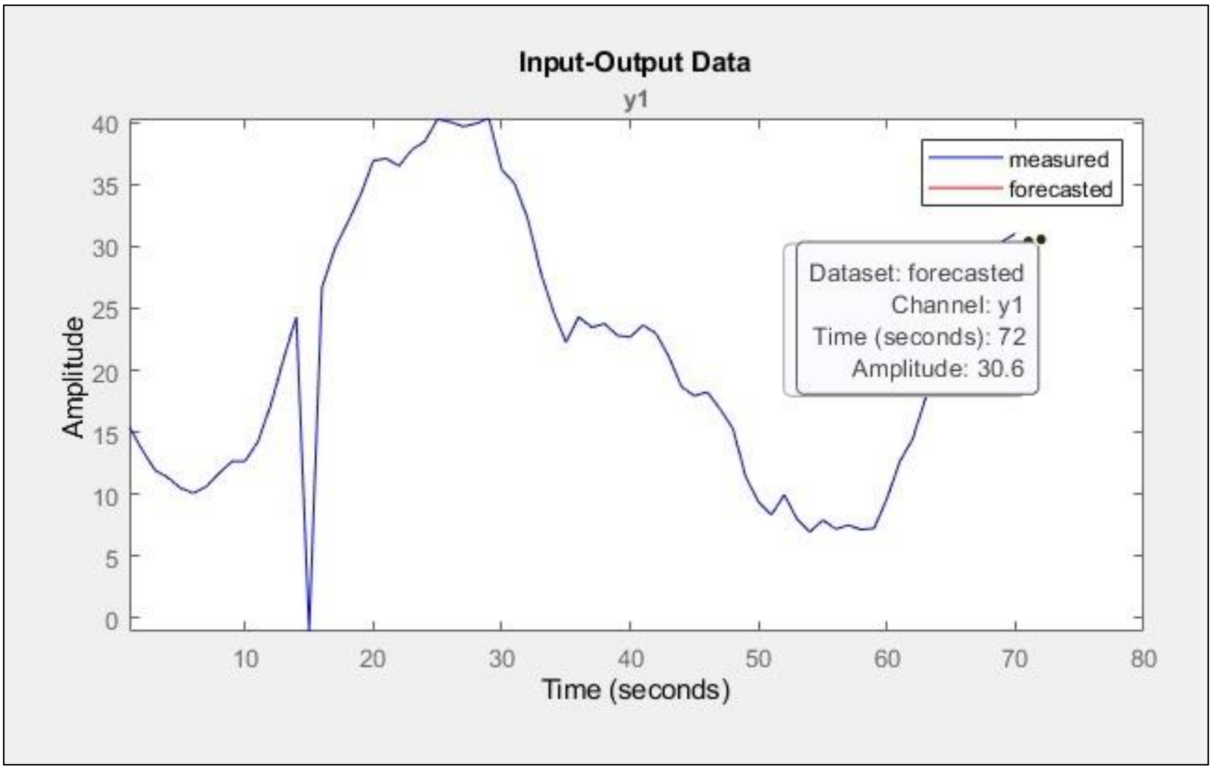

Figure 10 Plot of Time vs Magnitude of Sensor data. The measure value shows the recorded data and and preicted data is indicated in red color

Figure 10 shows how the proposed mechanism can predict the value in the cloud based on the history of the data. The prediction method used for the experiment is the auto-regression (AR) method for time series data. It can be seen that the difference between the actual and predicted values is less than 1 percent. Hence, it can be proved that the prediction algorithm used in the current proposed work introduces very minimal error and that the work is an enhancement over the works surveyed in the literature review.

5.6 Case Study 6

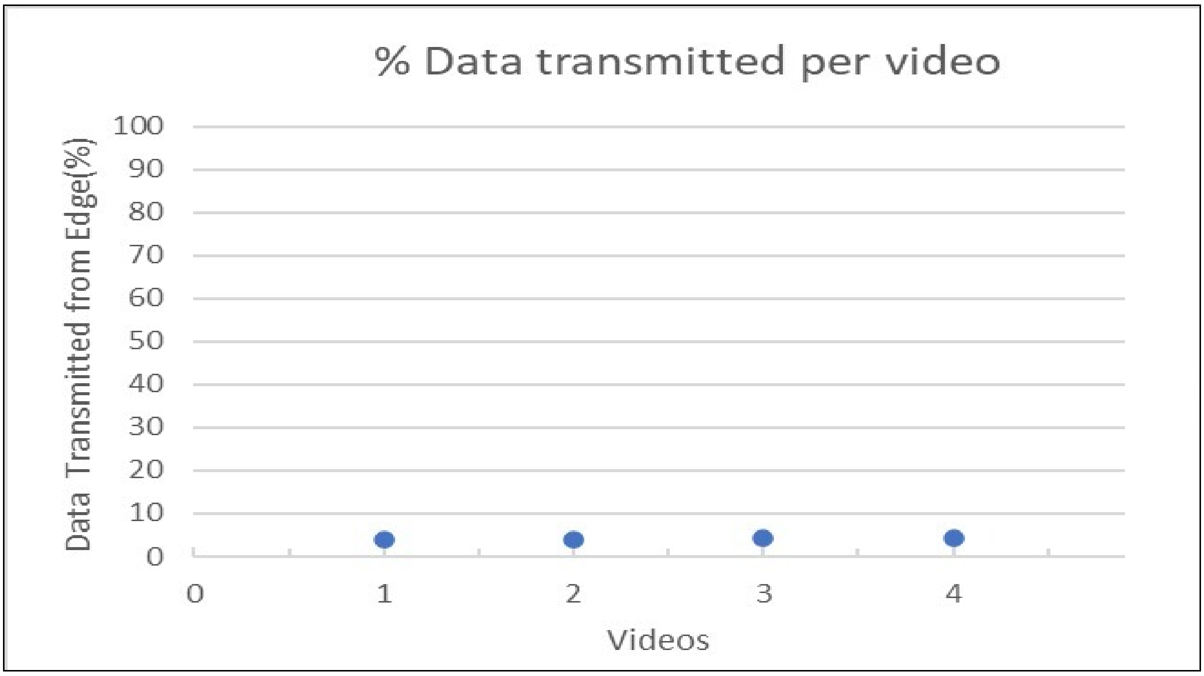

Figure 11 Percentage of data transmitted as a function of different input videos of same length

Figure 11 shows the percentage of data transmitted as a function of different videos with the same data length. As an evaluation method, we changed the input video source and checked the percentage of data transmitted from the edge. For maximum compression, the percentage of data transmission should be as low as possible, which inherently saves bandwidth and reduces latency, which was the main purpose of the research work. For evaluation purposes, we used four different videos with the same exact number of frames and constant FPS. It was observed that in the majority of cases, the percentage of data transmitted was below 10%. Hence, 90% of the remaining data is dropped at the edge computer itself.

5.7 Discussion

Because the CCTV footage obtained at the edge node was in raw video format, it was quite large. We were able to achieve up to 95% compression with the proposed algorithm, resulting in less lag during transmission. It is important to understand that only 5% of the data is transmitted for traffic analysis. Hence, it is not suitable for processing as a video. We also observed that the intervals between the frames were irregular. For our system, we could reach a reconstruction accuracy of up to 99.93%, whereas Pereira et al. (2022) and Zhang et al. (2021) also achieved an accuracy of 99.19% and 99.4%, respectively. The node computers that they deployed were capable of handling up to 32-bit operations. The frame rate that we could achieve post-compression was roughly 0.1 fps, whereas, as per the literature, Pereira et al. (2022) and Algiriyage et al. (2020) already achieve up to 25 fps. Since the main aim was compression and saving bandwidth, the lower the frame rate, the better. Most of the methods reported in the literature had less than three tiers in the architecture, except for one proposed by Pereira et al. (2022). We have also used a 3-tier architecture, which was easily deployable. As per the results obtained, when the value of K (the number of predicted values) increases, the prediction error also increases, indicating that immediate prediction is important. We did not choose only one prediction value because, with that, we would not have been able to plot the trend. The AR factor has a significant impact on the error. After conducting an experiment with varying the value of the AR factor from 4 to 6, it was found that the error was minimal when the AR factor was 5. Hence, its value was set to 5 for all the other experiments. The vehicle detection limit of 0.7 was the best, as the error was increasing with all other values. The proposed experiments were limited to only 4 videos, and in the future, more test runs could determine the best possible performance of the algorithm. The traffic factor was varied from 0.2 to 0.4 and can be extended up to 1. In this study, we also limited our vehicle detection limit from 0.5 to 0.8 with a step size of 0.1. Future researchers can change this limit from 0 to 1, as well as the step size.

6 Conclusion and Future Work

Traffic management is one of the important upcoming issues, as the number of vehicles on the road is increasing every day. We proposed a novel edge video processing technique that could remove 95% of the unnecessary data at the edge itself before transmitting it for traffic management. Our method achieved more than 99% data compression at the fog level, thereby significantly reducing the workload at the cloud. To prove the effectiveness of our proposed approach, we also applied an AR-based prediction algorithm to the cloud. We found that the absolute error was 0.07 between the actual value and the predicted value. We could easily reconstruct the data in the cloud, even after reduction. For traffic management, we computed the number of vehicles on the road and generated alerts whenever excess traffic was found in the frame. Overall, we expect this proposed system to be used in other applications such as air-traffic control, automated fertiliser distribution robots, etc.

Future researchers can focus on deploying the proposed algorithms on 360-degree pan-tilt-Zoom (PTZ) cameras. The data drop rate, which is an indicator of how much data is removed or dropped, is dynamic. We found that the data drop rate needs to be calibrated after a certain number of frames. The future researcher could also explore the opportunity to deploy the proposed algorithm on field-programmable gate array (FPGA)-based hardware that would eventually speed up the computation. The newer cars are pre-installed with sensors and require receiving periodic sensor data. The proposed algorithm can be used to reduce the transmission load of this sensor data. The proposed algorithm would be helpful for Google Maps, Apple Maps, etc. to collect the data efficiently and predict the routes more accurately and faster. The number of cars could have also been detected using the Canny edge detection technique. This will help us correlate the sensor data with the camera scenes. Overall, the proposed system provides a way to reduce that data transmission from the edge to the cloud using different data removal techniques.

References

Algiriyage, N., Prasanna, R., Doyle, E.E., Stock, K. and Johnston, D., 2020. Traffic Flow Estimation based on Deep Learning for Emergency Traffic Management using CCTV Images. In 17th International Conference on Information Systems for Crisis Response and Management (pp. 100-109).

Armawi, A., Danugroho, A., Apriliyanti, K., Asrofi, A. and Wahidin, D., 2022, March. Integrated CCTV-Based Traffic Management By Semarang Smart City During The Covid-19 Pandemic. In INTERNATIONAL CONFERENCE ON RESEARCH AND DEVELOPMENT (ICORAD) (Vol. 1, No. 1, pp. 192-209).

Brownlee, J., 2022. How to Choose an Optimization Algorithm. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/tour-of-optimization-algorithms/> [Accessed 1 August 2022].

Chavan, Rugved, Aryaman Ayush, Vidyadhar Aski, Arnav Chaubey, Arushi Sharma Arushi Sharma, and Vijaypal Singh Dhaka. "Comprehensive Review on Intelligent Traffic Management System in India." In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 2055-2061. IEEE, 2022.

Choudhary, V., Singh, Y. and Anand, P., 2022. Smart Traffic Monitoring with Fog and Cloud Computing. In Emerging Technologies for Computing, Communication and Smart Cities (pp. 317-327). Springer, Singapore.

Fadlullah, Z.M., Tang, F., Mao, B., Liu, J. and Kato, N., 2018. On intelligent traffic control for large-scale heterogeneous networks: A value matrix-based deep learning approach. IEEE Communications Letters, 22(12), pp.2479-2482.

Fouladfar, E., Khayyambashi, M.R. and Solé Pareta, J., 2021. Using cloud computing to improve urban traffic management ans optimization system. International journal of advanced research in engineering and technology, 12(4), pp.302-313.

Guillen-Perez, A. and Cano, M.D., 2021. Intelligent IoT systems for traffic management: A practical application. IET Intelligent Transport Systems, 15(2), pp.273-285.

Lei, T., Mohamed, A.A. and Claudel, C., 2018. An IMU-based traffic and road condition monitoring system. HardwareX, 4, p.e00045.

Lilhore, U.K., Imoize, A.L., Li, C.T., Simaiya, S., Pani, S.K., Goyal, N., Kumar, A. and Lee, C.C., 2022. Design and Implementation of an ML and IoT Based Adaptive Traffic-Management System for Smart Cities. Sensors, 22(8), p.2908.

Liljeblad, H., 2019. Does dynamic compression impact how well the player tracks multiple targets in a FPS game.

Liu, C. and Ke, L., 2022. Cloud assisted Internet of things intelligent transportation system and the traffic control system in the smart city. Journal of Control and Decision, pp.1-14.

Murphy, S., 2022. Number of new cars registered in 2021 soared by more than a fifth. [online] Irish Mirror. Available at: <https://www.irishmirror.ie/news/irish-news/number-new-cars-registered-2021-25926293> [Accessed 1 August 2022]

Neelakandan, S., Berlin, M.A., Tripathi, S., Devi, V.B., Bhardwaj, I. and Arulkumar, N., 2021. IoT-based traffic prediction and traffic signal control system for smart city. Soft Computing, 25(18), pp.12241-12248.

Pawłowski, P., Piniarski, K. and Dąbrowski, A., 2021. Highly efficient lossless coding for high dynamic range red, clear, clear, clear image sensors. Sensors, 21(2), p.653.)

Pereira, P., Melo, C., Araujo, J. et al. Availability model for edge-fog-cloud continuum: an evaluation of an end-to-end infrastructure of intelligent traffic management service. J Supercomput 78, 4421–4448 (2022)

Prasad, D., Kapadni, K., Gadpal, A., Visave, M. and Sultanpure, K., 2019, December. HOG, LBP and SVM based traffic density estimation at intersection. In 2019 IEEE Pune Section International Conference (PuneCon) (pp. 1-5). IEEE.

Raskar, C. and Nema, S., 2021. Modified fuzzy-based smart barricade movement for traffic management system. Wireless Personal Communications, 116(4), pp.3351-3370.

Sellén, M. and Sellén, M., 2022. 15 Common sTypes Of Car Sensors (& What They Do). [online] Mechanic Base. Available at: <https://mechanicbase.com/engine/car-sensors/> [Accessed 1 August 2022].

Wang, X., Ning, Z., & Wang, L. (2018). Offloading in Internet of vehicles: A fog-enabled real-time traffic management system. IEEE Transactions on Industrial Informatics, 14(10), 4568-4578.

Zhang, Y., Ma, Z., Song, X., Wu, J., Liu, S., Chen, X. and Guo, X., 2021. Road Surface Defects Detection Based on IMU Sensor. IEEE Sensors Journal, 22(3), pp.2711-2721.