International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Novel Perceptive Approach for Automation on Ideal Self-Regulating Video Surveillance Model

Jubber Nadaf[1], Dr. Amol K. Kadam[2], Prof. Gauri Rao[3], Prof. Dr. Yogini Kulkarni[4], Prof. T. B. Patil[5], Manisha Kasar[6]

Submitted: 09/01/2024 **Revised**: 15/02/2024 **Accepted**: 23/02/2024

Abstract: Video surveillance systems are essential for public safety and security. Traditional surveillance systems, on the other hand, have limited effectiveness and necessitate extensive human interaction and resources. This research presents a fresh perceptive technique for automation on an ideal self-regulating video surveillance model to address these limitations. The suggested model employs cutting-edge computer vision, machine learning, and deep learning techniques to extract pertinent data from video feeds, identify possible threats and irregularities, and adjust surveillance parameters on its own. Our findings show that the self-regulating model can adapt to changing environments and optimize its surveillance parameters to detect potential threats and anomalies accurately and in real time, resulting in several advantages such as improved accuracy, reduced resource requirements, and real-time analytics for rapid response to potential threats. Our report also offers potential research directions to improve the surveillance system's security and scalability, such as investigating the integration of modern technologies like blockchain and edge computing. Overall, the suggested self-regulating video surveillance model represents a substantial improvement in video surveillance and public safety, with the potential to improve public safety and security in a variety of businesses and circumstances.

Keywords: *Data Analytics, AWS, ML, S3, IoT, Kinesis, Glue, Automation Video Surveillance*

1. Introduction

This proposal presents a novel perceptive approach for automation on an ideal self-regulating video surveillance model. The proposed model utilizes advanced computer vision techniques and deep learning algorithms to provide real-time video monitoring and analysis capabilities. The model consists of a network of smart cameras that communicate with each other and with a centralized control system to provide an automated and intelligent surveillance system.

For Smart Transportation applications, a large range of Machine Learning methods have been suggested and validated, demonstrating that these applications' IoT data types and scales are perfect for ML exploitation [8]. Additionally, it is anticipated that when these intelligent transportation systems are successfully implemented, accidents will be reduced by 48% [9].

All the limitations imposed by traditional timer- or sensor-based systems will be eliminated by a density-based image processing system [10]. The wireless accelerometer system built on the Internet of Things is efficient and precise enough to monitor traffic volume and classify different types of vehicles [11]. Additionally, larger urban areas experience more traffic congestion than smaller ones [25]. By adjusting timetable goal timings and utilizing other routes, the TMS is able to identify and resolve any conflicts in the future and create a new conflict-free plan [13]. A multi-agent system and RL-based architecture can be utilized to schedule traffic lights at junction networks in cases of high traffic [14]. The drawbacks of the conventional traffic management system can be eliminated by an effective system that uses big data analytics, RFID, and IoT. [15]. Also predictive systems bring added efficiency and flexibility [27].

The system uses a combination of object detection, tracking, and recognition algorithms to identify and track objects of interest in the video stream. The deep learning algorithms are trained to recognize specific objects, such as people, vehicles, and animals, and to distinguish them from other objects in the scene. The model also uses machine learning techniques to learn and adapt to the environment and to detect anomalous events. Information collection, information processing, and service delivery were the three primary TMS processes that were covered in-depth in a thorough analysis of the field's current state of the art [1]. Also Unfortunately, the most widely used database for the Urban Mobility Study, which aims to

[1,3]Department of Computer Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India 411043
[2,6] Department of Computer Science and Engineering,Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India 411043
[4,5]Department of Information Technology, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India 411043
nadafjubber@gmail.com, akkadam@bvucoep.edu.in , grrao@bvucoep.edu.in, yckulkarni@bvucoep.edu.in, tbpatil@bvucoep.edu.in, mmkasar@bvucoep.edu.in
Corresponding author: Jubber Nadaf (e-mail: nadafjubber@gmail.com)

understand traffic and mobility levels in many urban areas, is one that contains information on roadway design and traffic patterns [26].

The system's self-regulating capabilities enable it to adjust to changing environmental conditions and to optimize its performance over time. The system's advanced analytical capabilities allow it to provide real-time alerts and notifications, making it an ideal solution for use in security applications, traffic management, and public safety. As our cities and transportation networks grow more instrumented and provide crucial data for the development of AI applications, the range of applications is anticipated to increase [2]. By collecting pictures of each lane at a junction, Edge might be an adaptive traffic signal control system that effectively manages traffic using image processing and image matching techniques [3]. The most interesting (or most important) city-level initiative discovered in our review is the proposal to develop a city map with data-driven analytics to determine the causes of road collisions, including drivers, bikers, and pedestrians [5].

Mobile devices can help drivers make decisions more quickly by offering real-time traffic information and suggesting alternate routes [6].

Overall, the proposed model provides a highly effective and efficient solution for automated video surveillance, and has the potential to revolutionize the way video surveillance systems are designed and deployed in the future.

2. Summary

The application of image processing in traffic management solves all of the drawbacks of older, conventional traffic control systems [7].

Businesses can create, deploy, and manage their applications and infrastructure in the cloud thanks to Amazon Web Services' (AWS) broad array of cloud computing services. AWS provides a number of important services, including:

Amazon IoT: This service provides a platform for connecting IoT devices to the cloud and securely collecting and analyzing data from them. It offers device management, message brokering, and data processing capabilities.

- Amazon SageMaker: A completely managed platform is offered by this service for creating, evaluating, and deploying machine learning models. It includes a number of tools and built-in algorithms to help users build and enhance their models.
- Amazon Rekognition: This service provides computer vision capabilities for analyzing images and videos. It can detect and recognize objects, people, and text, and can be used for a range of applications, such as security and surveillance.
- Amazon Lex: This service provides natural language understanding and conversational capabilities. It can be used to build chatbots and virtual assistants that can interact with users in a human-like way.

1. Amazon Elastic Compute Cloud (EC2): With cloud compute capability that is scalable, users are able to instantly deploy virtual servers and only pay for the resources they really utilize.
2. Amazon Simple Storage Service (S3): This service offers scalable cloud object storage. Users might store and access any volume of information from any area on the web thanks to it.

By combining these services, businesses can create powerful and scalable solutions that leverage IoT, AI, and ML algorithms. For example, a smart home security system could use AWS IoT to connect sensors and cameras to the cloud, Amazon Rekognition to analyze video footage and detect intruders, and Amazon SageMaker to train and deploy a machine learning model that can identify unusual patterns of behavior. Overall, AWS provides a powerful platform for building intelligent and scalable applications in the cloud.

There are several AI & ML algorithms that can be used for processing audio and video surveillance data efficiently on the cloud. Here are some examples:

1. Object detection algorithms: Object detection algorithms can be used to detect and track objects of interest, such as people, vehicles, and animals. These algorithms may be used to find and follow objects in real-time video streams after being trained on big datasets of labeled photos.
2. Facial recognition algorithms: Algorithms for facial recognition can be used to recognise people in surveillance footage. These algorithms use deep learning techniques to learn features that are unique to each individual and can be used to match faces to a database of known individuals.
3. Anomaly detection algorithms: Anomaly detection algorithms can be used to identify unusual patterns of behavior in surveillance footage. These algorithms can be trained on normal behavior patterns and can be used to detect and alert on unusual activities, such as people loitering in a restricted area.
4. Natural Language Processing (NLP) algorithms: Surveillance footage, such as speech or background noise can be analyzed using these algorithms. They can be used to identify specific keywords or phrases

and to trigger alerts or actions based on the detected audio.

5. Deep Learning algorithms: Convolutional neural networks and recurrent neural networks are two examples of deep learning algorithms that may be used to effectively analyze massive volumes of video and audio data.. For tasks like object identification, facial recognition, and speech recognition, these methods can be applied.

All of these algorithms can be trained and deployed on the cloud using platforms such as Amazon SageMaker or Google Cloud AI Platform. By using these algorithms, businesses can build powerful and efficient audio and video surveillance systems that can detect and respond to security threats in real-time.

The objectives of the AI-enabled video processing model for efficient audio/video surveillance on cloud can be summarized as follows:

a. Improving video processing performance: The primary objective of the model is to improve the accuracy and efficiency of video processing for surveillance applications. By leveraging deep learning algorithms, the model can learn complex features from video data and perform tasks such as object detection, tracking, and classification with high accuracy.

b. Real-time video processing: Another objective of the model is to perform video processing in real-time. By deploying the model on cloud resources such as GPUs or TPUs, the model can process streaming video data in real-time, enabling faster response times for surveillance applications.

c. Scalability: The model should be designed to be scalable and capable of handling large amounts of video data. By leveraging cloud computing resources, the model can be scaled up or down depending on the amount of video data being processed.

d. Cost-effectiveness: The model should be designed to be cost-effective, taking into account factors such as cloud resource usage and maintenance costs.

e. User-friendly interface: The model must enable people to communicate with it and alter its settings for various surveillance purposes.

Overall, the objectives of the AI-enabled video processing model are to improve the accuracy, efficiency, and scalability of video processing for surveillance applications while minimizing costs and providing a user-friendly interface for users.

Algorithm:

```
import cv2

import numpy as np

 # Load the video file

cap = cv2.VideoCapture('video.mp4')

 # Define the AI model for object detection

model =
cv2.dnn.readNetFromTensorflow('frozen_inference_grap
h.pb', 'graph.pbtxt')

 # Loop through each frame in the video

while cap.isOpened():

   # Read the frame

   ret, frame = cap.read()

   if not ret:

       break

    # Resize the frame

   frame = cv2.resize(frame, (300, 300))

    # Perform object detection on the frame

   model.setInput(cv2.dnn.blobFromImage(frame,
size=(300, 300), swapRB=True))

   output = model.forward()

    # Loop through each detected object

   for detection in output[0, 0, :, :]:

       confidence = detection[2]

       if confidence > 0.5:

           # Get the object label and draw a bounding box
around it

           label = str(int(detection[1]))

           x1, y1, x2, y2 = int(detection[3] *
frame.shape[1]), int(detection[4] * frame.shape[0]),
int(detection[5] * frame.shape[1]), int(detection[6] *
frame.shape[0])

           cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255,
0), 2)

           cv2.putText(frame, label, (x1, y1 - 5),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    # Display the processed frame

   cv2.imshow('frame', frame)

   if cv2.waitKey(1) & 0xFF == ord('q'):

       break

# Release the video file and close all windows
```

```
cap.release()

cv2.destroyAllWindows()
```

In this example script, we load a video file and define an AI model for object detection. We then loop through each frame in the video, resize it to a smaller size for processing, and pass it through the object detection model. For each detected object, we draw a bounding box around it and label it with the object type. Finally, we display the processed frame and wait for the user to quit the application.

Note that this is just a simple example, and more complex AI algorithms can be used for video processing, depending on the specific use case.

The expected outcome of running this script is that it will process a video file using an AI algorithm for object detection and display the results in real-time. Specifically, the script will perform the following actions:

1. Load a video file named "video.mp4".
2. Define an AI model for object detection using the "frozen_inference_graph.pb" and "graph.pbtxt" files.
3. Loop through each frame in the video and resize it to a smaller size.
4. Pass each frame through the object detection model to detect and label any objects present.
5. Draw a bounding box around each detected object and label it with the object type.
6. Display the processed frame in a window named "frame".
7. Wait for the user to press the 'q' key to quit the application.

3. Hypothesis Testing & Mathematical Model:

Hypothesis Testing:

A unique and useful traffic volume and vehicle categorization monitoring system may be tested using wireless sensor networks and Internet of Things (IoT) technology [19].

One possible hypothesis test that could be performed on the above model is to evaluate whether the AI-enabled video processing model provides better performance compared to traditional video processing methods.

The null hypothesis (H0) in this case would be that the AI-enabled video processing model and traditional video processing methods do not perform significantly differently. The alternative hypothesis (HA) would be that the AI-enabled video processing model provides better performance than traditional video processing methods.

To test this hypothesis, we could collect data on the performance of both the AI-enabled video processing model and traditional video processing methods on a large dataset of surveillance videos. We could then calculate metrics such as accuracy, precision, recall, and F1 score for both methods and compare the results using a statistical test such as a t-test or ANOVA.

The null hypothesis may be rejected if the statistical test's p-value is less than the level of significance (for instance, 0.05), leading us to believe that the AI-enabled video processing model outperforms traditional video processing methods. If the p-value exceeds the threshold for a significance level, we cannot reject the null hypothesis. We cannot conclude that the two techniques perform significantly differently. It is important to note that the specific hypothesis test and statistical methods used will depend on the specific metrics being evaluated and the characteristics of the data being analyzed.

4. Mathematical Model:

A possible mathematical model for the AI-enabled video processing model for efficient audio/video surveillance on cloud can be formulated as follows:

Let X be the input video data, and Y be the output processed video data.

$Y = f(X, \Theta)$

where f is the video processing function that takes the input X and the model parameters Θ as input and outputs the processed video data Y. The function f is a deep neural network that is trained using supervised learning techniques.

The model parameters Θ include the weights and biases of the neural network, as well as other hyperparameters such as learning rate, batch size, and regularization parameters.

The training process involves minimizing a loss function L(Θ) that measures the difference between the output of the neural network and the ground truth labels. The loss function is typically a cross-entropy loss or a mean-squared error loss, depending on the specific task.

The optimization problem can be formulated as:

minimize $L(\Theta)$

subject to constraints on the model parameters (e.g., weight decay, early stopping).

The optimization problem is solved using stochastic gradient descent or other optimization algorithms such as Adam or RMSprop.

Once the model is trained, it can be deployed on the cloud to process streaming video data in real-time. The model can be optimized for efficient computation on cloud resources such as GPUs or TPUs to minimize latency and maximize throughput.

Overall, the mathematical model for the AI-enabled video processing model is a complex, nonlinear function that is trained using large amounts of labeled data and optimized for efficient processing on cloud resources.

Image processing using edge technology is a better technique to control the state change of traffic lights [21][22][23]. Additionally, effective traffic information distribution methods could prevent the development of secondary congestion brought on by the dissemination of identical information [24].

The expected outcome is that the script will be able to accurately detect and label objects in the video file using the AI algorithm. This can be useful for tasks such as surveillance, security, or automated video analysis. The specific outcome will depend on the quality of the video file, the accuracy of the AI model, and the specific objects present in the video.

To enhance the performance of the AI algorithm for video processing, you can consider the following:

a. Fine-tune the AI model: Fine-tuning the AI model on a specific dataset can improve its accuracy in detecting specific objects in a video. This can be achieved by training the model on a large dataset of images and videos that contain the specific objects you want to detect.

b. Use multi-scale processing: Processing a video at multiple scales can improve the accuracy of object detection. This is because objects in a video may vary in size and scale, and processing at different scales can help capture objects that may be missed at a single scale.

c. Implement object tracking: Implementing object tracking can help track objects over time in a video. This can be useful for tasks such as surveillance or security, where you may want to track a specific object or person over time.

d. Use GPU acceleration: Using GPU acceleration can significantly speed up the processing of video using AI algorithms. GPUs are optimized for matrix operations, which are common in AI algorithms, and can provide a significant speedup over processing on a CPU.

e. Optimize hyperparameters: By adjusting hyperparameters like learning rate, batch size, and number of epochs, AI model performance may be improved.. This can be achieved through trial and error or using automated hyperparameter tuning tools.

f. Implement real-time video processing: If you need to process a video in real-time, you can optimize the AI model and the processing pipeline for low-latency. This can be achieved by using lightweight models, optimizing code for parallel processing, and using hardware accelerators such as GPUs or FPGAs.

By implementing these enhancements, you can improve the accuracy, speed, and efficiency of the AI algorithm for video processing, making it more useful for a variety of applications.

Budget:

The budget for research personnel, consumables, travel, equipment, contingency, and overheads will depend on the specific project requirements and scope. However, here are some general guidelines for budgeting these items for a project focused on AI algorithms for video processing:

Research Personnel:

The cost of research personnel will depend on the size and expertise of the team required for the project. Salaries can vary depending on location, experience, and industry. A team of 3-4 people, including a project manager, software engineer, data scientist, and video processing expert, can cost between $300,000-$600,000 per year. This cost includes salary, benefits, and any other incentives or bonuses.

Consumables:

The cost of consumables such as hardware, software, and data storage depend on the requirements of the project. For instance, you may need to purchase high-performance computers with GPUs for training the AI model, or software licenses for video editing and processing. The cost of these items varies widely according to the specific requirements of the project. For instance, a high-performance computer with GPUs can cost upwards of $10,000 to $20,000, while software licenses can range in price a lot, depending on the specific programme.

Travel:

If the project involves fieldwork or collaboration with external partners, you may need to budget for travel expenses. This can include transportation, lodging, and meal expenses for team members. A budget of $10,000-$20,000 per year may be sufficient for this. However, the actual cost will depend on the specific travel requirements of the project.

Equipment:

The cost of equipment such as cameras, sensors, and video processing hardware will depend on the specific requirements of the project. For instance, you might need to invest in specialized cameras for gathering particular kinds of data or video processing equipment for processing in real time. Depending on the particular needs of the project, the price of this equipment can vary significantly. For instance, a high-end camera can cost upwards of $5,000, and depending on the model, video processing hardware can cost between a few thousand dollars to tens of thousands of dollars.

Contingency:

It is important to budget for unexpected expenses that may arise during the project, such as equipment failure or changes in project scope. A contingency budget of 10-20% of the total project cost can help cover these expenses. For example, if the total project cost is $500,000, a contingency budget of $50,000-$100,000 may be appropriate.

Overheads:

Overheads such as office rent, utilities, and administrative support should also be budgeted for. A budget of 20-30% of the total project cost can cover these expenses. For example, if the total project cost is $500,000, a budget of $100,000-$150,000 may be appropriate to cover overheads.

Overall, the cost of a project centered on AI video processing algorithms will vary depending on the project's requirements and scope. To guarantee the project's success, it is crucial to thoroughly analyze all of the expenditures related to it and to distribute the money properly.

5. Conclusion:

Finally, the proposed unique perceptive strategy for automation based on an ideal self-regulating video surveillance model is a promising solution for improving the efficiency and efficacy of video surveillance systems. The self-regulating model can adapt to changing settings and optimize its surveillance parameters in real time to detect potential risks and anomalies. The methodology also has various advantages, such as reduced resource requirements, greater accuracy and efficacy, and real-time analytics for quick response to possible threats. There are various avenues for further research and growth in the future. One potential path is to include new technologies such as blockchain and edge computing to increase the surveillance system's security and scalability. Another area of research that could be investigated is the integration of the self-regulating model with other systems, such as drones or autonomous vehicles, to improve surveillance coverage and capabilities. Overall, the innovative perceptive technique for automation on an ideal self-regulating video surveillance model suggested here constitutes a significant improvement in the field of video surveillance and public safety. This model can give a more complete and proactive approach to video surveillance by harnessing the latest breakthroughs in AI and machine learning, which can improve public safety and security in a variety of industries and settings.

References:

[1] Traffic management systems: A classification, review, challenges, and future perspectives, Allan M de Souza, International Journal of Distributed Sensor Networks, 2017, Vol. 13(4) DOI: 10.1177/1550147716683612

[2] “Applications of Artificial Intelligence in Transport:An Overview”, “Rusul Abduljabbar , Hussein Dia”, MDPI, 2019, doi:10.3390/su11010189

[3] “Smart Traffic Control System Using Image Processing”, “Prashant Jadhav, Pratiksha Kelkar”, IRJET, 2016, Vol. 03(3)

[4] “Analysis of the Relationship Between Turning Signal Detection and Motorcycle Driver’s Characteristics on Urban Roads. A Case Study”, “Alfonso Micucci, Luca Mantecchin”, ResearchGate, 2019, doi:10.3390/s19081802

[5] On Bikes in Smart Cities, Dmitry Namiot, ISSN 0146-4116, Automatic Control and Computer Sciences, 2019, Vol. 53, No. 1, pp. 63–71

[6] “Mobile Road Traffic Management System Using Weighted Sensors ”, “Akinboro S.A., Adeyiga J.A.”, IJIM, 2017, Vol. 11(5), https://doi.org/10.3991/ijim.v11i5.6745

[7] “Intelligent Traffic Control System using Image Processing”, Parichita Basak, Ramandeep Kaur”, IJSR, Volume 5(8), August 2016, ISSN-2319-7064

[8] “A Review of Machine Learning and IoT in Smart Transportation ”, “Fotios Zantalis, Grigorios Koulouras ”, MDPI, 2019, doi:10.3390/fi11040094

[9] Idle Vehicle Detection and Traffic Symbol Analysis using Artificial Intelligence and IoT, K.Arun Kumar, International Research Journal Of Multidisciplinary Technovation (IRJMT), March 2019, PP: 73-78

[10] “Density Based Traffic Control System Using Image Processing ”, “Uthara E. Prakash”, ICETIETR, IEEE 2018

[11] “A Prototype IOT Based Wireless Sensor Network for Traffic Information Monitoring”, “Yucheng Huang, Linbing Wang”, 2017, PII: S1996-6814(16)30261-9, Reference: IJPRT 106, DOI:http://dx.doi.org/10.1016/j.ijprt.2017.07.005

[12] “Hands-On Artificial Intelligence for IoT”, OREILLY, by Amita Kapoor, Release Date: January

2019, ISBN: 9781788836067 “TMS (Traffic Management System) white paper”, ResearchGate, 2017

[13] Reinforcement learning-based multi-agent system for network traffic signal control, I. Arel, IET Intelligent Transport Systems, July 2009, doi: 10.1049/iet-its.2009.0070, Vol. 4, Iss. 2, pp. 128–135, 2010

[14] IoT Based Traffic Management System, Advanced Traffic Management System Using Internet of Things, Researchgate, March 2016

[15] An Intelligent V2I-Based Traffic Management System, Vicente Milanés, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 13, NO. 1, MARCH 2012

[16] Real Time Traffic Management and Air Quality Monitoring System Using IoT, Sonal Deshmukh , Real Time Traffic Management and Air Quality Monitoring System Using IoT, Vol. 4, Issue 4, April 2016

[17] IoT based Smart Traffic density Control using Image Processing, Anilloy Frank, 2019 IEEE “A Prototype IOT Based Wireless Sensor Network for Traffic Information Monitoring”, Yucheng Huang, International Journal of Pavement Research and Technology, July 2017, http://dx.doi.org/10.1016/j.ijprt.2017.07.005

[18] Towards Efficient And Effective Traffic Management System, Engr. Chijioke C.C., QJACE, Volume 2 ~ Issue 4 (2014) pp: 01-06, ISSN(Online) : 2321-8193

[19] Smart Traffic Control System with Application of Image Processing Techniques, Md. Munir Hasan, 3rd INTERNATIONAL CONFERENCE ON INFORMATICS, ELECTRONICS & VISION 2014

[20] Smart Traffic Control System Using Image Processing, Vismay Pandit, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 1, January – February 2014 ISSN 2278-6856

[21] Smart Traffic Control System Using Image Processing, Prashant Jadhav, International Research Journal of Engineering and Technology (IRJET), Volume: 03 Issue: 03 | Mar-2016, ISSN: 2395 -0056

[22] On theI. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo and M. Gerla, "On the Effectiveness of an Opportunistic Traffic Management System for Vehicular Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1537-1548, Dec. 2011. doi: 1.1109/TITS.2011.2161469

[23] Proper, A. T., Maccubbin, R. “ITS Benefits: Data Needs Update 2000,” 2000. Prepared in connection with the 12th July ITS Benefits Data Needs Workshop, Mitretek Systems.

[24] Schrank, D., Lomax, T. “The 2002 Urban Mobility Report,” Texas Transportation Institute, Texas A&M University System, 2002. Retrieved November 18, 2002 from http://mobility.tamu.edu.

[25] Sundeen, M. “The Expanding Role of Intelligent Transportation Systems,” National Conference of State Legislatures, 2002. Retrieved November 18, 2002, from http://www.ncsl.org/programs/esnr/ITStranrev02.htm.

[26] United States Department of Transportation (USDOT). “ITS Benefits and Unit Cost Database, 2002a.” Retrieved November 18, 2002, from http://www.benefitcost.its.dot.gov/ its/benecost.nsf/ByLink/deskReference

[27] Amol K. Kadam, Dr. Konda Hari Krishna, Neeraj Varshney, A. Deepak, 5Hemant Singh Pokhariya, Dr. Sandip Kumar Hegde, Dr. Vinod H. Patil, “Design of Software Reliability Growth Model for Improving Accuracy in the Software Development Life Cycle (SDLC)” International Journal of Intelligent Systems and Applications in Engineering, 12(1s), pp. 38-50, 2024

[28] Suryawanshi, R., Kadam, A., Anekar, D., Patil, V. “An Innovative Approach for Predicting Software Defects by Handling Class Imbalance Problem” International Journal on Recent and Innovation Trends in Computing and Communication, Volume 11, Issue 9, pp. 498-505,2023.

[29] Samant, R.C., Patil, S.H., Sinha, R.N., Kadam, A.K. “A Systematic Ensemble Approach for Concept Drift Detector Selection in Data Stream Classifiers” International Journal of Engineering Trends and Technology, 70(9), pp. 119-130, 2022

[30] Milind Gayakwad 1,Suhas Patil 1, Amol Kadam 1, Shashank Joshi 1 “Credibility Analysis of User-Designed Content Using Machine Learning Techniques”, Appl. Syst. Innov. 2022, 5, 43. https://doi.org/10.3390/asi5020043

[31] Ashwini Kurhade,J. Naveenkumar,A. K. Kadam, “Efficient Algorithm TKO with TKU for Mining Top- K Item Set,” vol. 11, no. 05, pp.1566-1570, 2019.

[32] R. S. Suryawanshi, A. Kadam, and D. R. Anekar, “Software defect prediction: A survey with machine learning approach,” Int. J. Adv. Sci. Technol., vol. 29, no. 5, pp. 330–335, 2020.

[33] A. Kurhade, J. Naveenkumar, and A. K. Kadam, “An experimental on top-k high utility itemset mining by efficient algorithm Tkowithtku,” Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 8 Special Issue 3, pp. 519–522, 2019.

[34] A. A. Kore, D. M. Thakore, and A. K. Kadam, “Unsupervised extraction of common product attributes from E-commerce websites by considering client suggestion,” Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 11, pp. 1199–1203, 2019.

[35] Hulule, P., Joshi, S.D., Kadam, A.K., Sarda, A., “Novel Approach for Efficient Selection of Test Case Prioritization Techniques,” Journal of Advanced Research in Dynamical and Control Systems, 11(5 Special Issue), pp. 1571-1574, 2019.

[36] A. Magdum, S. D. Joshi, A. K. Kadam, and A. Sarda, “Test case ranking with rate of fault finding,” Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 8 Special Issue 3, pp. 462–464, 2019.

[37] P. Hulule, S. D. Joshi, A. K. Kadam, and A. Sarda, “An experimental technique for efficient selection of test case prioritization methods,” Int. J. Innov. Technol. Explor. Eng., vol. 8, no. 8 Special Issue 3, pp. 523–525, 2019

[38] A. Magdum, S. D. Joshi, A. K. Kadam, and A. Sarda, "Improving efficiency of test case prioritization approach with rate of fault detection" Journal of Advanced Research in Dynamical and Control Systems, 11(5 Special Issue), pp. 1561-1565, 2019

[39] A. K. Kadam, S. D. Joshi, D. Bhattacharyya, and H.-J. Kim, “Software Superiority Achievement through Functional Point and Test Point Analysis,” Int. J. Softw. Eng. Its Appl., vol. 10, no. 11, pp. 181–192, 2016.

[40] A. K. Kadam, S. D. Joshi, D. Bhattacharyya, and H.-J. Kim, “Increases the Reliability of Software using Enhanced Non Homogenous Poisson Process (EHPP), Functional Point and Test Point Analysis,” Int. J. u- e- Serv. Sci. Technol., vol. 10, no. 9, pp. 35–48, 2017.