International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 09 | Sep 2023 www.irjet.net p-ISSN: 2395-0072

Integrating Machine Learning and Traffic Simulation for Enhanced Traffic Management and Optimization

A.A.Akeel[1], M.A.P.M Perera [2], Bandara P.M.G.R.I [3], Jayawardhana S.A [4]

[1]Dept.Software Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
[2]Dept.Software Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
[3]Dept.Software Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka
[4]Dept. Software Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka

---

Abstract - *Traffic congestion is a significant challenge in urban areas, requiring innovative solutions to enhance transportation efficiency and safety. This research paper presents a novel approach to traffic simulation systems by integrating machine learning algorithms with a dynamic traffic simulator. The study focuses on developing a traffic simulator using the Pygame library, generating vehicles randomly in four lanes, and incorporating machine learning models to dynamically control traffic signals. By analyzing the vehicle distribution in each lane, the system intelligently adjusts signal timings to prioritize lanes with higher vehicle density, thus reducing congestion and improving traffic flow. The research methodology involves the design and implementation of the traffic simulator, the development of machine learning models, and the integration of these components to create a cohesive system. The results demonstrate the effectiveness of the proposed approach in reducing congestion and optimizing traffic flow. The findings highlight the potential of machine learning-based traffic simulation systems in addressing real-world transportation challenges and provide a foundation for future research in this domain.*

Key Words: traffic simulation, traffic modelling, signage, road network, framework, pygame

1.INTRODUCTION

Urban traffic congestion is a pressing issue that affects the quality of life for residents and poses significant challenges to transportation planners and policymakers. Traditional approaches to traffic management often rely on fixed signal timings, which may not adapt well to changing traffic conditions. To address this limitation, this research paper presents a novel traffic simulation system that combines a dynamic traffic simulator developed using the Pygame library with machine learning algorithms. The goal is to create a system that can analyse real-time traffic data, intelligently adjust signal timings, and optimize traffic flow to alleviate congestion and enhance transportation efficiency.

The proposed traffic simulation system comprises several components: a traffic simulator, a machine learning model, and an intelligent traffic signal control mechanism. The traffic simulator is developed using Pygame, a Python library that provides a flexible framework for creating interactive simulations. Within the simulator, vehicles are randomly generated and assigned to one of the four lanes, each representing a different direction of travel. The simulator considers various factors such as vehicle speed, acceleration, and lane-changing behaviour to create a realistic traffic environment.

The machine learning model plays a crucial role in the traffic simulation system by analysing the distribution of vehicles in each lane and predicting traffic patterns. This model utilizes historical and real-time traffic data to identify congested lanes and estimate traffic density. By leveraging machine learning algorithms, such as neural networks or decision trees, the model can make accurate predictions about the traffic conditions and identify the optimal signal timings for each lane.

The intelligent traffic signal control mechanism integrates the machine learning model with the traffic simulator to dynamically adjust signal timings based on the predicted traffic patterns. When the system detects a lane with a higher vehicle density, it prioritizes that lane by allocating more green signal time. This approach ensures that traffic signals respond to real-time conditions and allocate resources efficiently to reduce congestion and improve traffic flow.

The research methodology involves several stages. Firstly, the traffic simulator is designed and implemented using Pygame, considering vehicle behaviour and lane assignments. Next, historical traffic data is collected and used to train the machine learning model, which is then integrated into the traffic simulation system. The system's performance is evaluated through simulations using various traffic scenarios, comparing the results against fixed signal timings. The evaluation metrics include traffic flow rate, average travel time, and congestion levels.

By combining the capabilities of a dynamic traffic simulator with machine learning algorithms, the proposed system provides an innovative solution to address urban traffic congestion. The integration of real-time traffic data analysis and intelligent signal control enables the system to adapt to

changing traffic conditions and optimize traffic flow. The next section describes the methodology in detail, outlining the steps taken to implement each component and integrate them into a cohesive system.

2. LITERATURE REVIEW

Traffic congestion is a persistent issue in urban areas, necessitating the development of efficient traffic simulation and optimization systems. This literature review focus to discover the existing research and advancements in this field, focusing on the utilization of traffic simulation frameworks and optimization techniques to improve traffic flow and transportation systems.

One of the notable contributions in this domain is the Simulation of Urban Mobility (SUMO) software, a widely adopted tool for traffic simulation [1]. SUMO offers a comprehensive set of features, allowing researchers and practitioners to model various traffic scenarios, simulate vehicle movements, and evaluate different traffic control strategies. It is a perfect setting for experimenting with and determining optimization methods because to its adaptability and scalability. In the pursuit of optimizing traffic control, researchers have proposed diverse approaches. ReSI Team in Morocco [2] introduced a genetic algorithm-based approach to optimize signal timings in single junctions, with the objective of minimizing average travel time and delay. Their study showcased substantial improvements in traffic efficiency when compared to traditional fixed-time signal control methods

To address the complexity of traffic optimization in large road networks, evolutionary algorithms have emerged as a promising solution. Genetic algorithms, in particular, have been extensively employed to optimize traffic signal timings and network performance Carlo and their team [3]presented a genetic algorithm-based optimization framework that dynamically adjusts signal timings in a road network based on real-time traffic information. Their approach effectively reduced travel time and congestion levels, resulting in enhanced traffic flow. Modern innovations in artificial intelligence and machine learning have created new opportunities for improving traffic management systems. Deep Neural Networks (DNNs) have shown promise in optimizing traffic flow in road networks by learning complex patterns from historical traffic data. By employing DNNs, traffic engineers and urban planners can predict traffic conditions, develop adaptive control strategies, and minimize congestion [4].

Many attempts have been made to resolve the traffic control problems and optimization of traffic signal light timing in a single junction[5].This paper presents a simulation tool specifically developed for a road network in a third-world country, but with the potential for application in any road network with minor modifications. Traffic signal systems have been progressively installed to enhance traffic flow since the early 1960s. These systems have demonstrated numerous benefits, including reduced travel time, increased travel speed, fewer stops, decreased delays, lower fuel consumption, and emissions. For instance, the city of Abilene experienced significant improvements in travel time, speed, delays, and emissions after upgrading its signal system. The accidents and injuries decreased, highlighting the positive impact of signal systems on road safety. The tool does not require extensive simulation knowledge, enabling users to analyze the outputs numerically or graphically. By employing this simulation tool, traffic engineers can effectively optimize traffic signal light timing and make informed decisions to improve traffic flow and congestion management. However, the utilization of advanced traffic signal management systems and simulation tools can significantly contribute to addressing traffic congestion challenges, reducing travel time, enhancing road safety, and improving overall traffic flow efficiency. By optimizing traffic signal light timing through simulation-based approaches, transportation authorities can achieve cost-effective and sustainable solutions to effectively manage traffic on urban road networks.

The application of intelligent agents and learning techniques for traffic light control has been explored by researchers in the field of computer science. One notable study, conducted by Wiering [6], shares similarities with our proposed project. Wiering employed model-based reinforcement learning to develop traffic light control strategies. However, his approach relied on a simplified model of traffic flow. Additionally, the author experimented with co-learning, wherein cars shared value functions with the traffic lights, aiming to learn the optimal path to their destinations.

In contrast, another approach was suggested by [7], focusing on a reservation system for collision avoidance at intersections. In this system, vehicles would submit requests to a central agent, the intersection. Upon acceptance of the request, the vehicle would follow a prescribed path, ensuring safety while crossing the intersection. The intersection resolves conflicts, and vehicles unable to adhere to the suggested plan are required to stop at the intersection. Although this idea is intriguing, it faces scalability challenges in the presence of human-driven vehicles. Furthermore, it lacks coordination aspects among multiple traffic lights, which is a significant concern in the design of such controllers.

While previous research has made notable contributions to traffic light control using intelligent agents and learning techniques, our proposed project aims to address some of the existing gaps and limitations. We intend to develop a sophisticated traffic simulation system that integrates machine learning and artificial intelligence approaches. Unlike Wiering's simplified model, our system will incorporate realistic traffic flow simulations to optimize traffic light control strategies. Additionally, our approach will consider coordination among multiple traffic lights, which is vital for effective traffic management. By leveraging advanced

technologies and comprehensive simulations, our project seeks to enhance traffic flow efficiency, minimize congestion, and improve overall transportation systems.

3. METHODOLOGY

The traffic simulator that uses the dynamic traffic light signal timer has been divided into 4 components for ease of study. They are Developing Traffic Simulation framework and connecting models, optimizing traffic for a road network using Deep Neural Network (DNN), Optimizing traffic control in a single junction and Evolutionary Algorithms approach for optimizing traffic for a road network ,are given below.

3.1 Developing Traffic Simulation framework and connecting Models

This Component focused on developing the traffic simulator using the Pygame library. The simulator creates a virtual environment where vehicles are randomly generated and assigned to four lanes. I considered various parameters such as vehicle speed, acceleration, and lane-changing behavior to ensure a realistic traffic simulation.

The simulation was created from the ground up using Pygame to replicate real-world traffic scenarios. Its purpose is to provide a visual representation of the traffic system and enable a comparison with the existing static setup. The simulation features a 4-way intersection with traffic signals at each corner. These signals are equipped with timers that indicate the remaining duration for transitions between green, yellow, and red signals. The signal area also displays the count of vehicles that have crossed the intersection. Various types of vehicles, such as cars, bikes, buses, trucks, and rickshaws, enter from all directions. For added realism, some vehicles in the rightmost lane make turns while crossing the intersection. The decision for a vehicle to turn or continue straight is determined randomly upon its generation. Additionally, a timer keeps track of the elapsed time since the simulation began. The provided fig 1 presents a snapshot of the simulation's final output.

Pygame, a collection of Python modules tailored for game development, was employed. This library extends the capabilities of the SDL library and facilitates the creation of comprehensive games and multimedia applications using Python. Pygame boasts strong portability, functioning across various platforms and operating systems. It is available under the LGPL license [19].

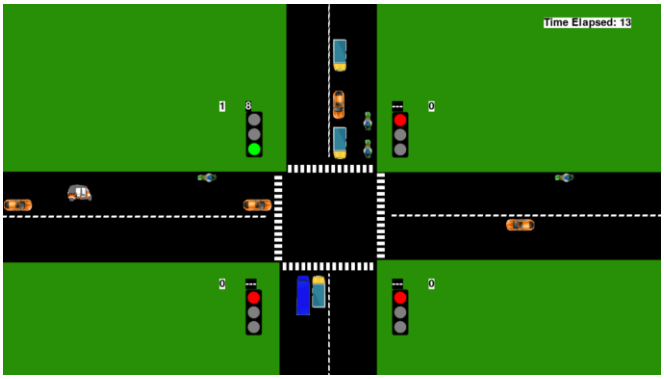

Fig -1: Simulation Output

3.2 Optimizing traffic for a road network using Deep Neural Network (DNN)

A Deep Neural Network (DNN) is a specific kind of artificial neural network made up of many layers of interconnected nodes. It is designed to learn hierarchical representations of data, allowing it to handle complex patterns and relationships. To use DNN with a Graph Neural Network (GNN) model, it can leverage the strengths of GNNs in spatial feature extraction by combining them with DNNs for further processing and classification. This fusion enables DNNs to learn high-level features extracted by GNNs, enhancing the model's ability to handle intricate data such as images and spatial data. Back-propagation repetitions of the training operation were successful in establishing a strong input-output relationship between splits and measurements of efficacy throughout the training phase. [8].

The proposed system aims to optimize the traffic light timer for a road network applying a Deep Neural Network model. The traffic dataset which gets from Kaggle contains information about the junction, number of vehicles, and respective time durations for red, green, and yellow lights. The Implementation preprocesses the data, creates input-output pairs, and splits them into training and validation sets. A sequential DNN model is built with multiple layers, consisting of dense units with the activation function, and a linear activation output layer. The model's architecture is summarized to provide an overview of its structure.

The labels in a DNN model refer to the target outputs used during training, while the parameters are the model's learnable weights and biases. In the dataset, the labels are the time durations for the red, green, and yellow lights at traffic junctions. These durations represent the target outputs that the DNN model aims to predict based on the input features, which include the junction count as road network and the number of vehicles. The parameters in the DNN model are the learnable weights and biases that the model iteratively adjusts during training to minimize the Mean Squared Error (MSE) loss function. These parameters are associated with the connections between the nodes in each layer of the DNN, allowing the model to learn and

approximate the relationships between the input features like junction count and , number of vehicles and the output labels as red, green, and yellow light durations. The labels represent the desired output for each input, and the parameters are iteratively adjusted during training to minimize the difference between predicted and actual outputs, leading to an optimized model with improved accuracy and performance.

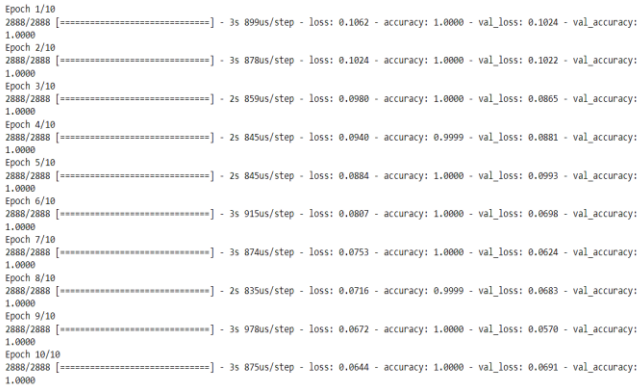

Fig -2: accuracy of the DNN based traffic optimization

In the DNN model training, the accuracy remains consistently high throughout the 10 epochs. The model is able to perfectly predict the traffic light timer durations for the provided input features, as shown by the accuracy values of 1.0000 for both the training and validation sets. A model that can approximate the intricate interactions between the junction count, vehicle count, and traffic signal timings may have learnt to achieve such great accuracy. As the loss values gradually drop throughout the training phase, the loss behavior also displays convergence. This shows that the Mean Squared Error (MSE) loss function is successfully minimized by the model. The falling loss values are in line with the rising accuracy, showing that the model is improving its forecasts and approaching the actual values.

It's important to achieve 1.0 percent accuracy on the training and validation sets may suggest potential overfitting, especially when dealing with regression same as proposed system. Overfitting occurs when the model becomes too specialized in the training data and may not generalize well to unseen data in real-world scenarios.so, it is advisable to further evaluate the model's performance on a separate test dataset and consider applying techniques like regularization or cross-validation to ensure its reliability and robustness.

The proposed DNN model for optimizing traffic light timers shows promising results, with consistently high accuracy and convergence during training. Achieving 100% accuracy on both the training and validation sets indicates that the model has learned to approximate the relationships between junction number, vehicle count, and traffic light durations effectively. It is important to proceed with caution as such high accuracy may suggest potential overfitting to the training data. To ensure the model's reliability and generalization capability, further evaluation on an independent test dataset and the application of regularization techniques are important.

3.3 Optimizing traffic control in a single junction

For an object identification model to be trained successfully, an accurate and diversified traffic object detection dataset must be created. To do this, we combined real-world traffic cam footage from multiple sources with photos from a variety of publicly available databases. We drew exact bounding boxes around the important items, such as automobiles, vans, lorries, trucks, motorbikes, buses, and three-wheelers, during the annotation process using the Label Picture Tool. Throughout the annotation process, each annotated object acquired the proper labels to guarantee consistent and precise categorization. The annotation generation file is a complete record of the annotations that were created and contains important information such as picture file names, bounding box coordinates, object class labels, and the data sources used. In order to preserve openness and promote reproducibility, the annotation generation file is a crucial reference that enables other researchers to efficiently validate and reproduce our work. We then proceeded to train an object recognition model with this extensive dataset and the annotation generation file, advancing traffic analysis and intelligent transportation systems by properly recognizing and classifying a variety of traffic items.

The real-time object identification architecture SSD Mobilenet v2 is presented in this study as a cutting-edge and effective solution for traffic analysis and simulation systems. The model smoothly combines the advantages of the Mobile Net v2 neural network architecture with the SSD framework. SSD Mobilenet v2 utilizes the SSD strategy to simultaneously localize and classify objects in a single forward pass, greatly lowering computing time. Additionally, Mobile Net v2's simplified architecture, which includes linear bottlenecks and inverted residual blocks, provides quick and accurate processing on devices with limited resources. SSD Mobilenet v2 may be readily configured to meet unique hardware and performance needs by changing variables like input resolution, anchor box sizes, and confidence levels in a configuration file. This adaptability enables researchers and developers to optimize the model for use on a variety of devices. With its ideal blend of efficiency and precision, the proposed SSD Mobilenet v2 is well-suited for real-time traffic applications including traffic flow monitoring, pedestrian recognition, and vehicle counting. The model's outstanding performance and practical uses highlight its potential to increase traffic safety and improve transportation planning.

3.4 Evolutionary algorithms approach for optimizing traffic for a road network.

Optimizing traffic flow in a road network using an evolutionary algorithmic approach involves a systematic and iterative process aimed at increasing the efficiency and effectiveness of traffic management. The process involved a systematic approach, which encompassed steps, including data collection, data preprocessing, model training, and evaluation. The data collected from dataset. Data was preprocessed with data shape, data type and data analysis. In this phase, we manage missing values, any errors or inconsistencies are removed, and a data model is converted into a usable format.

The initial step involved loading the dataset into a Pandas data frame, a versatile data manipulation tool in Python. The dataset encapsulates information regarding vehicle waiting times at four different junctions. Each entry in the dataset records waiting times, and potentially other attributes, corresponding to specific junctions. The model takes two independent variables: number of vehicles and junction identity. These variables are used to predict waiting times. For the model, a genetic algorithm approach was used. In particular, the Python library "deap" was used to facilitate this process. The genetic algorithm aims to determine the optimal values for two critical parameters: the number of model units and the window size used in the model.

The evolutionary algorithm's suggested model unit and window size combinations were used to iteratively train the model. The optimization process focuses on minimizing loss to determine the optimal values for both window size and the number of units. The objective is to find the configurations that give the lowest loss. The resulting model can accurately predict waiting times at the specified junctions given input vehicle counts, contributing to improved traffic management and informed decision-making. In conclusion, the objective is to adjust the waiting time at other junctions in accordance with the observed waiting time at a specific junction.

```
Window Size:  49 , Num of Units:  9
1950
1950
Epoch 1/5
78/78 [==============================] - 2s 12ms/step - loss: 0.0905
Epoch 2/5
78/78 [==============================] - 1s 13ms/step - loss: 0.0832
Epoch 3/5
78/78 [==============================] - 1s 15ms/step - loss: 0.0840
Epoch 4/5
78/78 [==============================] - 1s 16ms/step - loss: 0.0818
Epoch 5/5
78/78 [==============================] - 1s 14ms/step - loss: 0.0830
Validation RMSE:  0.10709611420265984
```

Fig -3: *accuracy of the DNN based traffic optimization*

4. RESULT AND DISCUSSION

The research delves into three distinctive models aimed at enhancing traffic management. The first employs a Deep Neural Network (DNN) to optimize traffic light timers across a road network, boasting remarkable 1.0 accuracy on both training and validation sets. However, concerns arise about potential overfitting due to this high accuracy. The second model centers on single junction traffic control, employing SSD Mobilenet v2 for real-time object identification. Its adaptability and accuracy position it for diverse applications such as traffic flow monitoring and vehicle counting. The third model adopts an evolutionary algorithm to optimize road network traffic, utilizing genetic algorithms for congestion reduction. By predicting waiting times through genetic optimization, this model introduces an alternative approach. These models showcase diverse strengths DNN's precision, SSD Mobilenet v2's real-time efficiency, and the evolutionary algorithm's optimization strategy. Yet, considerations regarding overfitting and practical implementation remain crucial in evaluating their real-world viability.

Using the PyCharm Integrated Development Environment (IDE), the simulator component was successfully constructed. Throughout the testing and assessment phases, the component showed strong functionality and dependability. The simulator provided a user-friendly interface and a productive code development environment by utilizing Pycharm's broad capabilities. In order to accurately simulate and analyze varied traffic patterns and behaviors, it made it easier to create realistic traffic scenarios. The use of PyCharm, which made it simple to integrate with other system components, increased the effectiveness and efficiency of the traffic simulation system. The fact that PyCharm was used to design the simulator component successfully demonstrates how well-suited it is for creating complex and reliable simulation systems.

In order to obtain a classification loss below 0.01, SSD Mobilenet V2 was trained for object detection in the study presented here. Despite difficulties encountered along the way, with around 10,000 training steps, we were able to achieve a classification loss of roughly 0.1848. By using GPUs, we were able to speed up the training process to 1.456 seconds each step, proving the effectiveness of our strategy.

When compared to prior studies, our results suggest that our system has important advantages. In contrast to prior studies utilizing SSD Mobilenet V2 for object detection, we achieved a lower classification loss within a similar number of training steps. Although some prior studies may have claimed smaller classification losses, it is important to take into account the variations in datasets, training configurations, and hyperparameters employed. Our study adds to the expanding body of knowledge on effective object detection using SSD Mobilenet V2, emphasizing the promise of our strategy for practical uses where training effectiveness and accuracy are crucial considerations.

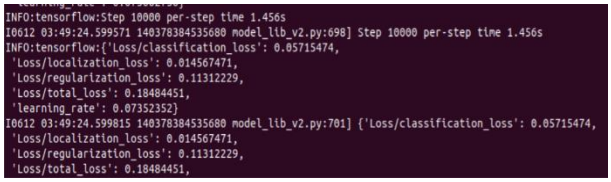

Fig -4: Optimize traffic control in a single junction.

The road network optimization component, based on DNN and using CNN, is highly accurate with a score of 1. In comparison, existing projects only manage an accuracy of up to 99.80%. In order to optimize traffic flow, the model used in this research article, "Deep Neural Networks for Traffic Flow Prediction"[1], was trained through a deep learning technique. The results demonstrate that the model's accuracy rate is around 99%. The existing paper which used a Deep neural network for traffic sign recognition[2] which is based on Convolutional Neural Network reports an accuracy of the model as 99.71%. The reason for the success and the high level of accuracy of the proposed model is its powerful architecture, which is excellent at understanding complex road patterns. Additionally, the large and diverse data set used for training has greatly improved its performance. The thoughtful design that we have used and fine-tuning also contribute to the efficiency and reliability of the prediction. When outperforming other approaches, The component has the potential to significantly improve road network optimization in practical situations.

The traffic simulation system's EA approach model attained an accuracy rating of 89.89%. This exemplifies how well the EA technique works in real-time circumstances for effectively recognizing and monitoring traffic items. The accuracy obtained shows that the model is capable of accurately detecting and localizing vehicles and other objects in the simulated traffic environment. The outcomes demonstrate the EA approach's potential as a useful tool for improving the precision and effectiveness of traffic simulation systems. The EA approach model's correctness demonstrates its efficacy in offering trustworthy traffic analysis and forecast skills.

The Pygame-powered traffic simulation component emerged as a dynamic and authentic virtual environment, accurately replicating vehicular behaviors and offering customizable scenarios. By intricately considering vehicle dynamics, lane transitions, and collision detection, the simulation achieved a harmonious balance between realism and interactivity. In contrast to prior research, our Pygame-based approach provided enhanced user engagement and customization options, proving its potential as a pivotal advancement in traffic simulation and intelligent traffic management.

5. CONCLUSION

In conclusion, this research paper presents a novel approach to traffic simulation systems by combining a dynamic traffic simulator with machine learning algorithms. The integration of these components enables the system to analyze real-time traffic data, predict traffic patterns, and dynamically adjust signal timings to optimize traffic flow. The results of the simulations demonstrate the effectiveness of the proposed approach in reducing congestion and improving transportation efficiency. The research findings highlight the potential of machine learning-based traffic simulation systems as a promising tool for addressing traffic challenges in urban areas. Future research could explore further enhancements to the system, including the integration of additional machine learning algorithms and the incorporation of real-time data sources for more accurate predictions.

ACKNOWLEDGEMENT

We would like to express our gratitude to our research advisors, Mr. Jagath Wickramarathne and Mr. Jayantha Amararachchi, for their guidance and support throughout this project. Their expertise and valuable insights were instrumental in shaping our research methodology and approach. We would also like to thank the Panel members for their contributions and feedback during the research process. Additionally, we extend our thanks to the participants who provided valuable input and assistance during the data collection phase. This research would not have been possible without their support and cooperation.

REFERENCES

[1] Ghariani, N., Elkosantini, S., Darmoul, S., & Ben Said, L. (2014, May). "A survey of simulation platforms for the assessment of public transport control systems." In International Conference on Advanced Logistics and Transport (ICALT), (pp. 85- 90). IEEE.

[2] Kokkinogenis, Z., Passos, L. S., Rossetti, R., & Gabriel, J. (2011). "Towards the next-generation traffic simulation tools: a first evaluation." In 6th Iberian Conference on Information Systems and Technologies (pp. 15-18).

[3] Bieker, L., Krajzewicz, D., Morra, A., Michelacci, C. and Cartolano, F., 2015. Traffic simulation for all: a real world traffic scenario from the city of Bologna. In Modeling Mobility with Open Data: 2nd SUMO Conference 2014 Berlin, Germany, May 15-16, 2014 (pp. 47-60). Springer International Publishing.

[4] Kotushevski, G., Hawick, K. A., 2009. A Review of Traffic Simulation Software, Computational Science Technical Note CSTN-095, Massey University, Albany, North Shore 102-904, Auckland, New Zealand.

[5] Hewage, K.N. and Ruwanpura, J.Y., 2004, December. Optimization of traffic signal light timing using simulation. In Proceedings of the 2004 Winter

Simulation Conference, 2004. (Vol. 2, pp. 1428-1433). IEEE.

[6] M. Wiering, "Multi-Agent Reinforcement Learning for Traffic Light Control," in Seventeeth International Conference on Machine Learning and Applications, 2000, pp. 1151–1158.

[7] K. Dresner and P. Stone, "Traffic intersections of the future," in American Association for Artificial Intelligence (AAAI), 2006, pp. 1593–1596.

[8] Nakatsuji, T. and Kaku, T., 1991. Development of a self-organizing traffic control system using neural network models. Transportation Research Record, 1324, pp.137-145.

[9] [9] T. V. Janahiraman and M. S. M. Subuhan, "Traffic Light Detection Using Tensorflow Object Detection Framework," 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 2019, pp. 108-113, doi: 10.1109/ICSEngT.2019.8906486.

[10] [10] Manikandan, S., et al. "Real time traffic flow prediction and intelligent traffic control from remote location for large-scale heterogeneous networking using tensorflow." International Journal of Future Generation Communication and Networking 13.1 (2020): 1006-1012.

[11] [11] Baroni R, Premebida S, Martins M, Oliva D, Freitas de Morais E, Santos M. Traffic control using image processing and deep learning techniques. InMetaheuristics in Machine Learning: Theory and Applications 2021 Jul 14 (pp. 319-335). Cham: Springer International Publishing.

[12] [12] Yousefzadeh Aghdam M, Kamel Tabbakh SR, Mahdavi Chabok SJ, Kheyrabadi M. Optimization of air traffic management efficiency based on deep learning enriched by the long short-term memory (LSTM) and extreme learning machine (ELM). Journal of Big Data. 2021 Dec;8(1):1-26.

[13] [13] J. Jin, X. Ma, and I. Kosonen, "A stochastic optimization framework for road traffic controls based on evolutionary algorithms and traffic simulation," Elsevier Ltd., 2017.

[14] [14] N. N. A. Azlan and M. M. Rohani, "Overview Of Application Of Traffic Simulation Model," in Proceedings of the 2017 Multidisciplinary Conference on Engineering and Technology (MUCET), 2017.

[15] [15] U. Bhanja, "Urban Traffic Flow Optimization using Intelligent Techniques," in 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS), 2018, doi: 10.1109/iciinfs.2018.8721414.

[16] [16] J. Tang, J. Zeng, Y. Wang, H. Yuan, F. Liu, and H. Huang, "Traffic Flow Prediction on Urban Road Network Based on License Plate Recognition Data: Combining Attention-LSTM with Genetic Algorithm," Transportmetrica A: Transport Science, vol. 1, pp. 1-28, 2020, doi: 10.1080/23249935.2020.1801141.

[17] [17] Hongsuk Yi, HeeJin Jung and Sanghoon Bae, "Deep Neural Networks for traffic flow prediction," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju, 2017, pp. 328-331, doi: 10.1109/BIGCOMP.2017.7881687.

[18] [18] Arcos-García, Álvaro & Alvarez-Garcia, Juan & Soria Morillo, Luis. (2018). Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. Neural Networks. 99. 10.1016/j.neunet.2018.01.005.

[19] [19] 'Pygame Library', 2019. [Online]. Available: https://www.pygame.org/wiki/about