How to Cite:
Priya, S., & Shyam, P. (2022). Driving pattern recognition using mobile telematics. *International Journal of Health Sciences*, *6*(S2), 8656–8669. https://doi.org/10.53730/ijhs.v6nS2.7240

Driving pattern recognition using mobile telematics

S. Priya
Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, 600089, India
Corresponding author email: spriyasrmist@gmail.com

Pranav Shyam
Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, 600089, India

***Abstract*---**Mobile Telematics is an application that enables it's users to detect driving behavior through sensors that already exist on a common smart phone device. Sensors such as Gyroscopes, GPS, Radio-Positioning, Existing Cameras, Location APIs, etc will help detect and identify valuable pieces of information, this can then be used to create unsupervised learning models on the driver's driving behavior and display feedback for the driver to identify and correct in order to minimize the risk of a potential accident. Current telematics models are not efficient and accurate enough to reduce the chances of road accidents, the sensors required to allow for users to access the needed data may also be expensive. Thus, we have found that our system is not only improved but also relatively inexpensive since we are using devices most of our user base already have -- Smartphones. Our framework consists of three main components -- Self Organizing Maps, Nine-Layers Deep Auto-Encoder, Partitive Clustering Algorithm. The self organizing maps simplifies complex pieces of data into consumable information, the deep auto encoder takes into consideration anomalies of said data and extracts its features, and the partitive clustering algorithms groups the data based on a set of driving behavior classifications, Ex: "Talking on the Phone", "Texting", etc…Furthermore the system we will attempt to create with maximal efficacy will serve as a good base for future works that could potentially save lives and reduce cost for insurance. With the collected data, a risk analysis system can be built to create a more real time application that can warn the driver of impending danger based on habit detection from the data that's been collected from not just one driver but several.

***Keywords*---**mobile telematics, IoT application, self organizing maps, partitive clustering algorithm, nine-layer deep auto-encoder, driving behaviour.

Introduction

One of the biggest crisis our modern world is currently facing is the grim reality of road accidents. Over a million people tragically die every year due to road accidents. Crash Injuries is said to be the 8th largest cause of death and the biggest cause of death for young adults between the ages 5-29 [1]. The economic crisis of this issue is even larger, estimates suggest that between 2015-2030, road injuries will cost the world economy about 1.8 Trillion US$ [2]. Both the loss of life and the economic burden that comes with it takes a heavy toll on society as a whole, so it is of paramount interest to solve this issue as quickly as possible.

The Solution -- An unsupervised Pattern Recognition framework that takes data from existing sensors present in a common smartphone to create an accurate detailed model that detects patterns in driving behavior by picking out dangerous habits that could potentially increase the chance of road accidents. Modern IoT devices have made this increasingly possible to the point where we are capable of utilizing existing hardware sensors in a common smartphone to create a pattern recognition system that collects an enormous amount of data effective enough to create a risk analysis model that can protect the driver from potential road accidents.

Telematics is an IoT application that largely relies on perceptive sensors such as gyroscopes, Cameras, GPS, Radio Positioning, etc… that detects the condition/location of its environment and passes it back to the server as data which can later be used for pattern recognition purposes via machine learning. By mapping a set of behaviors into distinct characteristics and letting the algorithm train and decide if what the driver is doing fit those defined characteristics through image detection, the model can accurately react to the actions of the driver and warn them of potential risks and can keep the driver alert.

Fig 1 : Distinct Set of Behaviors

Most smartphones contain many perceptive sensors that can be used to collect data to help determine driving pattern behavior. This simplifies the expense issue since most people already have smartphones, so obtaining external hardware is not necessary. Therefore, implementing this scheme at a large scale would be plausible and people would learn to improve their driving skill thanks to the risk analysis system we will create. With the help of this framework insurers no longer need to spend money on expensive internal sensors to create an environment for driver monitoring [3].

There are two main methods to detecting driving behavior (1)Rule Based Driving Detection, which is a method used to weed out risky behaviors through various mathematical descriptions. (2)The second method utilizes Previously defined templates and mathematical descriptions of various different driving styles to help determine the safest way to drive. Pattern matching and machine learning algorithms are then used to match these pre defined driving styles in order to create a suitable risk analysis.

The objective of this research is to build an effective and inexpensive risk analysis system that utilizes mobile telematics to take in driving data to provide real time feedback to the driver that would potentially reduce the risk of road accidents.

Proposed System

Due to recent developments in IoT technology and the improvements made to the modern smartphone, the existing system has become outdated and unnecessarily expensive. A system only the rich can afford is not universally applicable. The current system overly relies on simulations and inbuilt sensors which add up to the cost of the vehicle. A system predicated on utilizing what the userbase already has is necessary for a universal risk analysis that is both cheap and effective at reducing risks of fatal injury through road accidents. Utilizing smartphones that now have the ability to support complex self-learning frameworks is the best way forward in order to universally equip people with road safety that is both safe and economically viable.

Generally speaking, a modern smartphone possess more IoT devices that will be useful for tracking driving behavior than modern day car. So it makes sense to utilize these tools. With the internet getting more widely accessible especially in cities(accident prone zones) various complex APIs that support Unsupervised Self-Learning algorithms can be implemented on the server side and the needed data can sent back onto the smartphone as real time feedback which will indefinitely reduce risks.

Image Detection by using a simple smartphone camera can allow the framework to determine the state of the driver and provide an effective and immediate response to that. The application is expected to provide an effective response after detection of risky driving behavior through a sophisticated self learning framework that is created to detect and respond to certain driving styles and driving behavior.

Literature Survey

In this section, we will summarize relevant literature we have used related to mobile telematics, pattern recognition and driving style analysis that helped create an effective risk analysis model for driving behavior. The survey also includes brief notes on machine learning algorithms used in our model that helped our system detect and distinguish driving behavior.

Existing Sensors in a Common Smartphone

To understand driving behavior we first need information our model can process to create a suitable risk analysis for the driver to use. For that we need to figure out available and feasible data collectors that already exist in common smartphones[10]. First we took a look at the limitation when it comes to present day technology of motion sensing in common day smartphones and how efficient and effective it is at finding changes, and we found promising results [4]. Most common smartphones come with inbuilt gyroscopes, accelerometers and compasses which are accurate and can help in tracking direction data which is handy when trying to learn driving behavior patterns.

GPS, Location APIs and Radio Positioning can help our systems access traffic data, the quality of specific roads the driver is driving on,etc… This can provide for useful information in the frequency and alertness of our framework which saves on processing time,server space and energy exerted by the smartphone [5]. Inbuilt Cameras in a smartphone can act as internal sensors that can detect the alertness of the driver, and warn the driver if they are not paying attention. Camera Visuals would also show signs of drunk driving, sleep levels, etc… all of which being visual cues for the framework to detect and react[6].

Algorithms for Pattern Recognition

In our research, we recognized three specific algorithms that would work to create a suitable pattern recognition framework. (1)Self-Organizing Maps(SOM). (2)Deep Auto-Encoder, (3)Partitive Clustering Algorithm,

Self-Organizing Maps

SOM is one of the most widely used algorithms in the market, and for good reason. With high amounts of data being considered, expenses to store this data even temporarily would prove to be difficult. An algorithm that can simplify that and would automatically pick out stand out features out of the heaps of available data is very useful. Self Organizing Maps are trained using unsupervised, competitive learning patterns to return summarized and simplified versions of the input data that takes up a compressed amount of space containing only relevant subscripts of the original data that is required for our data processing[7]. The algorithm will help heavily reduce server costs, as well as the RAM Requirements and Storage needed for the user to actively run this application. SOM can help efficiently reject unnecessary information and a well trained SOMs algorithm are effective in organizing and passing essential data.

Figure 2 depicts how the algorithm basically works to extract anomalous information based on required criteria through deep learning techniques.

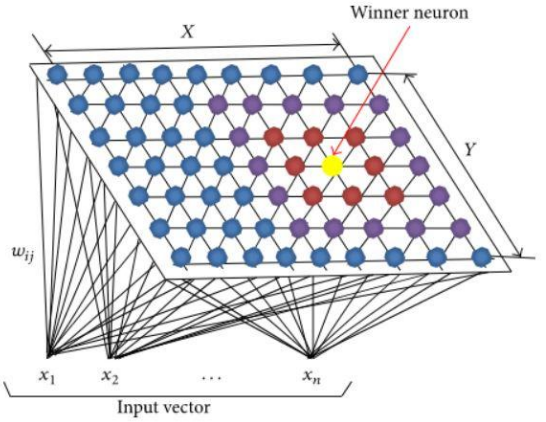


Fig 2 : Self-Organizing Maps

Deep Auto-Encoder

A Deep Learning model that is a set of several auto-encoders grouped together in a neural network architecture. One Auto-Encoder has two parts, an encoder and a decoder. The encoding layer consists of a function $h = f(W x + b)$which is used to encode the input data upto the middle layer when the decoder layer actives with function $h = f(W `x + b`)$which reunites the encoded data.

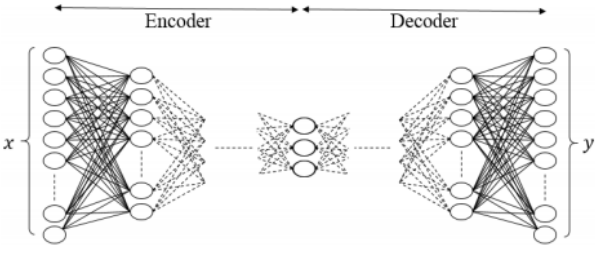


Fig 3 : Multi-Layer Deep AutoEncoder

Partitive Clustering Algorithm

This is an unsupervised learning technique required to group together unlabelled data using mathematical criterias and are categorized accordingly for suitable feedback representation in a smooth and concise manner. This is done by prototyping, which is a rule base that defines each category for accurate clustering of data. The clustering algorithm trains to sort a relatively large batch of input data and categorize them based on a defined prototype. Each clustering algorithm has a special method that defines a rule base, data that satisfy said rule base gets classified accordingly and is added into the specified category.

For our framework, we used K-means clustering algorithm. A popular partitive

clustering algorithm that we found[8] is suited for our application. K-means trains to detect anamolies that suit a certain set of rules and categorizes them based on those specific mathematical factors. So like SOM it organizes and categorizes data, only this is a lot more specific and requires higher levels of prototyping and a more complex rule-base. The algorithm works best for small sets of data after certain anamolies have already been detected.In our framework we figured it is ideal to use this as our final classification algorithm after SOM reduces the overall input data and extracts its features, and the deep auto-encoder encodes and decodes the data to build an optimal risk analysis software.

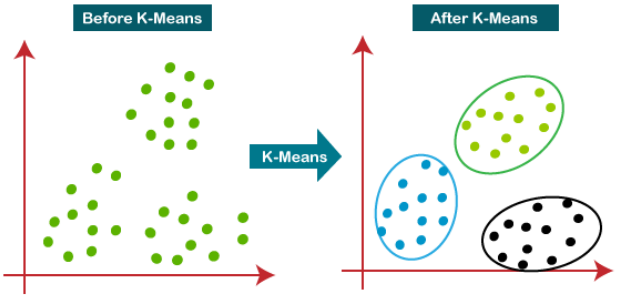


Fig 4 : K-Means Clustering

Driving Behavior and Risk Analysis Survey

To create an optimal pattern recognition framework for classifying driving behavior we needed to understand the actions which are considered most risky and result in the most accidents in order to prioritize. We have found that extensively using visual sensors like the camera can be a useful tool for the software to determine risky behavior, actions such as drunk driving, texting, radio monitoring, driving while sleepy could result in severe accidents.

Camera monitoring could alert a distracted driver in real time and ensure an accident does not occur. Actions that break road rules such as lane switching and red light jumping could also be monitored through gyroscopes and inertia sensors that detect direction change and vehicle acceleration would reduce risky behavior from the driver. Location Sensors like GPS and radio positioning can allow the software to access specific APIs that determine traffic density and bad quality and could let the driver know the optimal speed the driver is supposed to be driving in. This would prevent overspeeding. Adding to that weather APIs can be included to the framework which can be a factored into determining the suggested speed for the vehicle [10].

System Architecture

The objective is to create a Mobile Telematics system that can record driving behavior and identify potential misteps the driver can make through an unsupervised pattern recognition algorithms and prevent potential accidents. To turn this complex problem comprehensible and plausible an effective system architecture is needed. We have simplified our framework into three distinct modules:

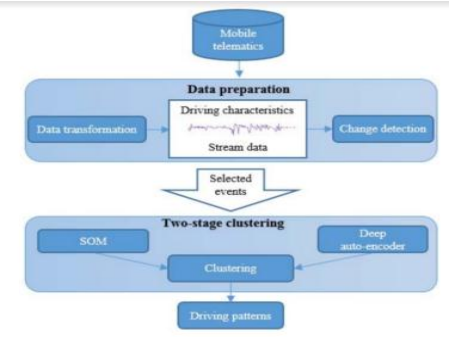


Fig 5 : Architecture Diagram

Data Collection

Sensors present in the smartphone record driving behavior and changes in driving patterns and environment. Perceptive Sensors like GPS, Radio Positioning, Gyroscopes, Compass, Smartphone Cameras,etc... are used to generate the needed data to detect what the driver is doing.

Data Preparation

In this module, the collected data is transformed in a way that can be consumable to unsupervised learning algorithims in the next next module. For the collected data to be useful for the application, the raw data collected by the smartphone detectors need to be processed. This is where Data Transformation and Change Detection Algorithm algorithm comes into play and exist to simplify the raw data. The generated streams of data need to be efficiently utilized so as to not reduce the performance of our system. Smartphone sensors are not aware of the data they record. During stops and times of inactivity, the data recorded become unnecessary. To filter out this data, Change Detection Algorithms so as to not over burden our system. With the help of the change detection algorithm the frequency at which data is recorded based on surrounding traffic, weather, etc... can largely help with the rejection of unnecessary pieces of information.

Two-Stage Clustering

After filtering out the raw data and handpicking necessary events, the Two-Stage Clustering begins and the data is processed through three algorithms -- SOM, Deep Auto-Encoder and Partitive Clustering(k-means).

Self-Organizing Maps (SOM)

For this application, SOM is used to simplify the data further reduce and filter out the existing data post-preparation. The algorithm works to return summarized and simplified versions of the input data that takes up a compressed amount of space containing only relevant subscripts of the original data that is required for our data processing. Picking out anamolies that can be identified as useful for

driving behavior analytics can allow the following algorithms to process necessary information a lot more efficently and accurately.

Deep Auto-Encoder

An algorithm used to encode and decode the data recreating it into a desired format the data to be clustered into specified categories via k-means. For the project, we will be using a nine layer Deep Auto-Encoder. This algorithm is necessary for image processing and will be helpful in identifying sudden movement patterns from the driver. Driving pattern detection therefore becomes a matter of utilizing an existing database of driving style images, which the search engine will then compress to 30 numbers, and compare that vector to all the others in its index.

K-means Clustering

K-means trains to detect anamolies that suit a certain set of rules and categorizes them based on those specific mathematical factors. So like SOM it organizes and categorizes data, only this is a lot more specific and requires higher levels of prototyping and a more complex rule-base. The algorithm works best for small sets of data after certain anamolies have already been detected.In our framework we figured it is ideal to use this as our final classification algorithm after SOM reduces the overall input data and extracts its features, and the deep auto-encoder encodes and decodes the data to build an optimal risk analysis software.

Driving Patterns

The resultant data is the required data necessary for the output of our application. A risk analysis can now be computed for our system to identify the driver's driving styles and risky habits, On identifying this the system can accordingly provide effective critiques in real-time or after an extended period of time which would reduce the probability of the driver taking risks that could result in a fatal accident.

Implementation

Any unsupervised learning model will need to be trained to be able to carry out the necessary tasks. All three of the Unsupervised Learning algorithms (i.e SOM, Deep Auto-Encoder) are trained on Jupyter Notebook, and the code is implemented on Python3.0. The type of training carried out is unsupervised and the algorithm learns by itself. Time Taken to obtain effective results => 1hour 40mins(approx)

Once trained, the code has the abiity to view changes in the image and determine/classify said behaviors through K-means Clustering algorithm from within the car. Location API are accessed to obtain other necessary pieces of data, data such as traffic density, weather type, and several other factors that could affect the conditions for driving. Python gives us the ability to use its vast set of existing open source libraries, making development easy and hassle free.

Libraries Used

Fig 5 : Python Libraries

Sklearn(Sci-Kit Learn)

Out of the array of machine learning libraries existing in open source, sklearn is one of the best and most widely used ML libraries in the current market. The library supports both supervised and unsupervised learning processes, thus being perfect for two-stage clustering that utilizes numerous different algorithms and various different kinds of data is transformed during these processes.

Tensorflow

This open source library helps build useful and effective Artificial Intelligence models, building a deep auto-encoder model for our research is complex, but the functions offered by the library makes the compartmentalizes that issue and simplifies development.

OpenCV (cv2)

With the help of open CV we were able to transform colored images into black and white, this helped make the process of image detection more identifiable. This then helped convert the existing image into numpy arrays that helped figure out distinct changes within the image.

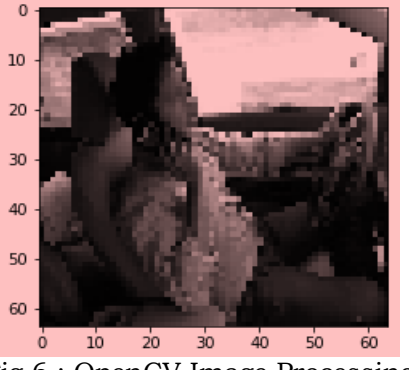


Fig 6 : OpenCV Image Processing

Matplotlib

Graphical Representation for output processing, it converts raw data and returns graphical representations of it as visual data plot points.

Keras

It is the most widely used deep-learning framework currently in the market. This framework tightly integrated to another library called tensorflow, a machine learning library. Keras offers a way for us a way to define neural networks for the rule base of our system that define the various behaviors and driving styles for the k-means clustering algorithm to classify.

Fig 7 : Classifying Behavior

Seaborn

Seaborn is a library built from Matplotlib. Similar to matplotlib it is used to create beautiful visuals for interface creation and data representation. Helpful to depict driving style analytics in concise and an aesthetic manner.

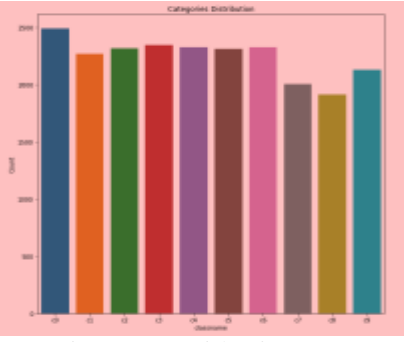


Fig 8 : Seaborn Graphical Representation

Results

An enormous amount of our own data is run through the program to check the validity and the efficiency of our code,

```
In [30]: print('There are %s total images.\n' % (test_files_size + x_train_size + x_test_size))
         There are 102150 total images.

In [31]: print('There are %d training images.' % x_train_size)
         There are 17939 training images.

In [32]: print('There are %d total training categories.' % categories_size)
         There are 10 total training categories.

In [33]: print('There are %d validation images.' % x_test_size)
         There are 4485 validation images.

In [34]: print('There are %d test images.'% test_files_size)
         There are 79726 test images.
```

Amount of Data Used

The results reveal an accurate response, K-means was successfully able to classify exactly what the driver was doing based on provided description.

```
activity_map = {'c0': 'Safe driving',
                'c1': 'Texting - right',
                'c2': 'Talking on the phone - right',
                'c3': 'Texting - left',
                'c4': 'Talking on the phone - left',
                'c5': 'Operating the radio',
                'c6': 'Drinking',
                'c7': 'Reaching behind',
                'c8': 'Hair and makeup',
                'c9': 'Talking to passenger'}
```

Data-Classification for Output Generation

```
1/1 [==============================] - 0s 10ms/step
```

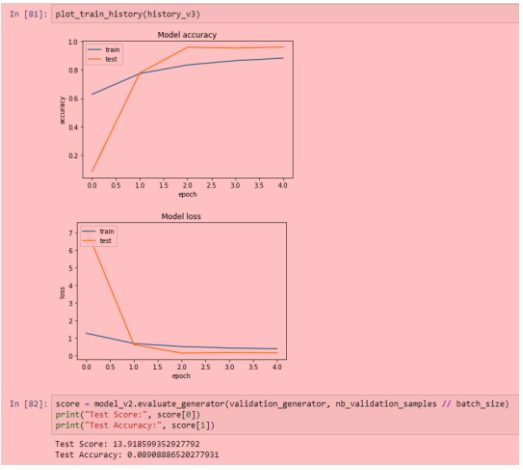
```
Y prediction: [[1.2155936e-11 1.0431237e-10 1.0326209e-16 1.0000000e+00 3.6404668e-12
  1.5199173e-14 4.5117892e-17 8.3156655e-15 2.4768763e-14 1.0994798e-15]]
Predicted: Texting - left
```

```
1/1 [==============================] - 0s 12ms/step
```

```
Y prediction: [[2.0767131e-05 1.8503353e-09 2.4333419e-06 1.9515453e-06 4.7688847e-04
  9.9948871e-01 9.4384791e-07 1.7299358e-07 6.9759080e-06 1.0532477e-06]]
Predicted: Operating the radio
```

```
1/1 [==============================] - 0s 0us/step
```

```
Y prediction: [[0.0036575  0.0005556  0.49625647 0.00401278 0.35107592 0.00583718
  0.02224432 0.02293139 0.09242272 0.00100625]]
Predicted: Talking on the phone - right
```



Conclusion & Future Works

The future for mobile telematics is promising, there is an endless amount of data that could be collected that can improve driving behavior. If given the chance the application could have the potential to completely negate road accidents all together with the help of self-learning and AI algorithms. Our system can be improved in multiple ways and is definitely far from perfect. Ideal vehicle speed without the need for road signs and signals can also be implemented, telling the driver the ideal speed to go by encourages good driving. Recording the drivers destinations, and allowing change detection algorithms to figure out if the path ahead is risky or not through location and traffic data is a promising field of research to look into. A system that reports risk factors that can be caused by weather disturbance isalso useful data for the driver to prevent risks and risk environments, APIs that assess the quality of the roads can also be extensively collected and applied on to the application. Areas were accidents have happened or remote area with numerous obstacles in the way can also be checked and reported using smartphone cameras, a more efficient algorithm is needed to achieve this but a good base to start from is here.

Acknowledgments

The authors wish to thank SRM Institute of Science and Technology for their constant support and encouragement.

References

1. https://www.cdc.gov/injury/features/global-road-safety/index.html
2. Simiao Chen, ScD Michael Kuhn, PhD Prof Klaus Prettner, PhD Prof David E Bloom, PhD "The global macroeconomic burden of road injuries: estimates and projections for 166 countries" -- September 2019.
3. Ignatovich, L. (2020). USE OF INNOVATIVE TECHNOLOGIES BY INSURANCE MARKET ENTITIES. TELEMATICS IN CAR INSURANCE. *Three Seas Economic Journal*, *1*(2), 7-11.
4. Case, MA, Burwick, HA, Volpp, KG. Accuracy of smartphone applications and wearable devices for tracking physical activity data. JAMA 2015;
5. S. Lee, G. Tewolde and J. Kwon, "Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application," *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 353-358, doi: 10.1109/WF-IoT.2014.6803187.
6. E. B. Ermis, V. Saligrama, P. Jodoin and J. Konrad, "Motion segmentation and abnormal behavior detection via behavior clustering," 2008 15th IEEE International Conference on Image Processing, 2008, pp. 769-772, doi: 10.1109/ICIP.2008.4711868.
7. D. Miljković, "Brief review of self-organizing maps," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1061-1066, doi: 10.23919/MIPRO.2017.7973581.
8. Z. Wang, Y. Zhou and G. Li, "Anomaly Detection by Using Streaming K-Means and Batch K-Means," 2020 5th IEEE International Conference on Big Data Analytics (ICBDA), 2020, pp. 11-17, doi: 10.1109/ICBDA49040.2020.9101212.
9. Saiprasert, C., Pholprasit, T. & Thajchayapong, S. Detection of Driving Events using Sensory Data on Smartphone. *Int. J. ITS Res.* **15,** 17–28 (2017). https://doi.org/10.1007/s13177-015-0116-5
10. Banu, J. F., Muneeshwari, P., Raja, K., Suresh, S., Latchoumi, T. P., & Deepan, S. (2022, January). Ontology-Based Image Retrieval by Utilizing Model Annotations and Content. In 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 300-305). IEEE.
11. M. Siami, M. Naderpour and J. Lu, "A Mobile Telematics Pattern Recognition Framework for Driving Behavior Extraction," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1459-1472, March 2021, doi: 10.1109/TITS.2020.2971214.
12. Latchoumi, T. P., Swathi, R., Vidyasri, P., & Balamurugan, K. (2022, March). Develop New Algorithm To Improve Safety On WMSN In Health Disease Monitoring. In 2022 International Mobile and Embedded Technology Conference (MECON) (pp. 357-362). IEEE.
13. Karnan, B., Kuppusamy, A., Latchoumi, T. P., Banerjee, A., Sinha, A., Biswas, A., & Subramanian, A. K. (2022). Multi-response Optimization of Turning Parameters for Cryogenically Treated and Tempered WC–Co Inserts. Journal of The Institution of Engineers (India): Series D, 1-12.

14. Naeem, Sajid, and Aishan Wumaier. "Study and implementing K-mean clustering algorithm on English text and techniques to find the optimal value of K." *Int. J. Comput. Appl* 182.31 (2018): 7-14.
15. Latchoumi, T. P., & Parthiban, L. (2021). Quasi oppositional dragonfly algorithm for load balancing in a cloud computing environment. Wireless Personal Communications, 1-18.
16. Dabiri, Sina, and Kevin Heaslip. "Inferring transportation modes from GPS trajectories using a convolutional neural network." Transportation research part C: emerging technologies 86 (2018): 360-371.
17. Wang, Wenshuo, and Junqiang Xi. "A rapid pattern-recognition method for driving styles using clustering-based support vector machines." *2016 American Control Conference (ACC)*. IEEE, 2016.
18. Saiprasert, Chalermpol, Thunyasit Pholprasit, and Suttipong Thajchayapong. "Detection of driving events using sensory data on smartphone." *International journal of intelligent transportation systems research* 15.1 (2017): 17-28.