Individual Assignment Applied Data Science (TDT4259)

Author: Ana Barrera Novas

(Student number 128680)

Autumn 2024 (11/10)

NTNU - Norwegian University of Science and Technology

Index

1. One pager

In this section, a quick overview of the problem, motivations and solutions is given. The domain chosen is traffic management.

1.1. Intent

Every single one of us has felt frustrated on the road at some point in our lives. And every single one of us knows about terrible accidents and pollution impacts that being on the road implies. The driving system (quite universal) tries to minimize this, but the regulations and signs are not enough, so technological systems were included (radars, electrical signs, traffic lights...). And even though they were designed and modified through time taking into consideration data of the road (how many cars go through and at what frequency, how much time is needed for pedestrians to cross it...), it's still fixed, and could be improved. It could become **dynamic** and intelligent.

1.2. Desired outcome

Improving the traffic management system would lead to very worth pursuing outcomes. Less traffic congestion would imply less time loss in people's everyday lives, improving their **quality of life**. Also, less pollution, as cars would stay less time on the road. This is very important in our actual world, now that **environment-friendly** actions and consciousness have gained power. And obviously, one of the most important matters and motivators for this solution: a reduction of the number and severity of the **accidents**.

1.3. Deliverable

The solution would be a set of smart systems that work individually or together for different ends but same objective. Being more clear, I will classify the main problem into four sub-problems: **real-time traffic optimization**, **traffic prediction**, **anomaly detection** and **route optimizations**. To solve them, after some research, I found some data science techniques that seem the best and most appropriate. For traffic optimization, **reinforcement learning** and **computer vision** would be the most logical to use; for traffic flow prediction, **time series forecasting** would be useful; for anomaly detection, **unsupervised learning** and again **computer vision**; and for route optimizations, **graph-based algorithms** would be most appropriate.

1.4. Constraints

Taking into account that this is a very important matter and a main point in the everyday lives of everyone (driver or not), it has been already very investigated and optimized. Every step taken in this area has to be thought very thoroughly and make sure it works perfectly before putting it into practice, as a miscalculation could lead to serious accidents.

There are other elements that should be taken into consideration for this problem. **Security** is the main objective, so everything should be focused around it. **People's trust** on a system like this should be maximal in order to work, and this could be hard to accomplish. Also, this works in a really broad area, so governments are involved and it should be implemented internationally to make an actual difference in pollution-sense, for example. Furthermore, implementing a system like this could imply some **privacy issues** as cameras are involved.

2. Design document

2.1. Overview

In this document, a solution for daily traffic problems is presented. Using different data science and machine learning techniques, it is suggested an optimization for an already very well developed system. The main objective is to reduce traffic congestion by optimizing routes and traffic lights based on real-time traffic and anomalies. The document's objective is to analyze the most important aspects of the problem, constraints and solutions, and get feedback from it.

2.2. Motivation

In order to understand the real motivation behind finding a solution we have to analyze some data.

To begin with, traffic congestion can significantly increase travel times. For example, in the UK, vehicles traveling during peak hours can experience delays. Comparisons show that the average delay per vehicle mile was estimated at around 45 seconds, depending on road type and location. Also, in urban areas, congestion can lead to travel time being almost double that of free-flow conditions.

About usage of private car usage, over the last 10 years, trends have fluctuated. Initially, there was a slight decline in the use of private cars, especially in urban areas, as more people adopted public transport or alternative mobility options. However, since COVID-19 pandemic, private vehicle use grew, as people wanted to avoid public transportation. In some countries like the UK, car traffic levels in 2021 were still slightly below pre-pandemic levels but recovering quickly, with newer trends showing a higher reliance on cars for daily commutes.

And about accidents, traffic congestion contributes to a significant portion of them. It has been estimated that stop-and-go traffic increases the likelihood of rear-end collisions and lane-change accidents. For instance, traffic congestion is responsible for 30-40% of accidents in urban areas due to frequent braking and lane switching.

So, in conclusion, we can see that the problems are there, and solving them would really make a difference. And we are talking about quality of life, environmental protection and accidents, so they are very important matters. And nowadays, with the increase of solutions of this kind, I believe it is worth to begin working and implementing all the advancements made in AI in such a relevant matter.

2.3. Success metrics

One of the main goals is to decrease the average travel time for commuters. Success can be measured through a percentage reduction in travel time during peak hours. For example, a 20% reduction in average travel time during rush hours would be a clear indicator of improved traffic flow. Also, tracking occurrences of traffic jams and stop-and-go situations to notice if a reduction happened. A significant percentage like 25% less times happening could be a good goal and would definitely reflect better overall traffic flow.

Another way success can be measured is by tracking the decrease in CO_2 emissions due to reduced idle time in traffic . A decrease in emissions would reflect a more environmentally friendly system.

A reduction of the average wait time for pedestrians at crosswalks (in around a 30% as a first goal) would indicate more efficient traffic light coordination for all road users.

Feedback from drivers and pedestrians could give a good sense of success. User satisfaction rates and survey results could indicate how well the system improves their experience and time management.

2.4. Requirements & Constraints

- **Functional Requirements**:
  - **Real-time responsiveness**: The system should adjust traffic lights and suggest alternative routes within milliseconds to ensure a seamless traffic experience.
  - **System integration** (constraint): The solution should integrate with existing city cameras and radars, add more that are required for the system to work efficiently and provide real-time updates to GPS systems and government bodies.
  - **User notifications**: Drivers should receive real-time route suggestions and notifications about traffic or anomalies via the application they use (car software or external phone or specific-device application).
- **Non-functional/Technical Requirements**:
  - **Scalability**: The system should scale to handle traffic data from multiple cities to actually achieve the expected goals. Every city has different road complexities, and it should work in all of them.
  - **Security and data privacy** (constraint): Data from cameras and GPS systems must be encrypted to prevent unauthorized access and ensure user privacy. This is a very important matter in this system, because user's trust is one of the main pillars for it to work. Policies and procedures to guard data should be established, but always working under bigger regulations like for example GDPR [1].
  - **Cost** (constraint): It is important to establish an accordant budget for the system. The most expensive aspects are related to hardware, as installing high-quality cameras, radars, and sensors at intersections can be very costly (these devices need to support real-time data collection and computer vision). Additionally, cloud storage and processing for large volumes of real-time data can also rise costs, especially for complex cities. And the development of complex machine learning algorithms, such

as those for traffic optimization and anomaly detection, also requires skilled personnel and significant time.

- **Low latency**: Traffic adjustments should happen within 100ms to ensure real-time performance, with a P99[2] latency target of under 500ms.
- **Redundancy**: The system should have fallback mechanisms to handle failure like backups if cameras or radars fail. The standard system and rules still work, so if the system were to fail, going back to the traditional system would always be an option. The system has to work perfectly or else directly go to the backup, as having it working poorly is not an option because of safety reasons (constraint).

2.4.1 What's in-scope & out-of-scope?

This is a rather big problem, and also a rather big solution. This can obviously offer many issues when it comes to budget, required personal and hardware. Chopping it down makes it easier to understand and to decide which areas could really be implemented.

- Real-time traffic optimization: use camaras and radars to manage traffic lights.
- Traffic predictions: use data recovered to send information to car GPS, applications like Google Maps and the involved government or political charges for statistics and studies to improve the system.
- Anomaly detection: use cameras and radars to send information to authorities in real time in necessary cases.
- Route optimizations in real time: the system uses all the information received by the other 3 points and it sends suggestions to GPS applications of each driver for alternative routes.

There are some alternatives and functions that fall out of scope due to different reasons. This is further developed in 7.1.Alternatives. A couple of examples here:

- Automatic signs to manage traffic congestion like if it were an authority are out of scope. The system is thought to prevent it and improve it, but in case in which authorities' job is needed, the system can only notify them.
- Limiting routes or not allowing all options is not the objective. While it may be true that sending just one option to each driver and letting the system manage perfectly the traffic flow would be ideal, it limits the driver a lot, it would seem very controlling and would not be popular.

2.5. Methodology

2.5.1. Problem statement

The problem is divided into several subproblems, each framed using appropriate machine learning techniques. Real-time traffic optimization and route optimization will be framed using **reinforcement learning** for adaptive control of traffic lights, **computer vision** for vehicle and pedestrian detection, and **graph-based algorithms** for optimal route calculations. Traffic prediction will be addressed with **time series forecasting**, using historical traffic data to predict future conditions. Finally, anomaly detection will be treated as an **unsupervised learning** problem, where patterns are identified from unlabeled data, combined with computer vision (the algorithm learns the "normal" patterns of traffic from historical data and flags anything that deviates significantly as an anomaly (like accidents and roadblocks).

2.5.2. Data

The data for training the models will include traffic data from road sensors with vehicle counts, speeds and times; historical video data from traffic cameras for vehicle detection and anomaly identification; GPS data to monitor vehicle movements and predict traffic flows; and more historical data on patterns, accidents and congestion levels.

For the actual serving, the same data will be provided, but in this case they will be real-time inputs (from traffic cameras, sensors and GPS systems to continuously update predictions and optimize traffic flow.

2.5.3. Techniques

- **Reinforcement learning** (more specifically, **Q-learning**): used to optimize traffic light cycles in real-time. An RL agent is trained to make decisions about traffic light timings based on traffic flow data. The environment is modeled as intersections. The agent receives the data (real-time input from traffic sensors and cameras like vehicle counts, waiting times and pedestrian activity and selects actions (switching lights, adjusting durations…). Its goal is to minimize traffic congestion and waiting time by learning through feedback (positive rewards for reduced delays and improved traffic flow).
- **Computer vision**: used to detect vehicles and pedestrians in real time for optimizing traffic flow and keeping it safe. Object detection algorithms like **YOLO** (You Only Look Once) combined with a **CNN** can identify vehicles and pedestrians, count their numbers, and track their movement. This data will inform the RL system, allowing it to make smarter traffic control decisions.
- **Time Series Forecasting**: used to predict future traffic conditions and optimize routes to prevent congestion. A model like **SARIMA** (Seasonal AutoRegressive Integrated Moving Average) could be used. These types of models will consider factors such as the time of day, weather, special events, and historical patterns of congestion. The predictions will then feed into route optimization systems.
- **Graph-based algorithms**: used to find optimal routes for drivers, reducing travel time and avoiding congestion. The road network will be modeled as a graph, where

intersections are nodes and roads are edges with weights corresponding to travel time or congestion levels. Route optimization algorithms like a **Dynamic A*** will be then used to find the best path and continuously update routes as conditions change.

- **Unsupervised** (or **supervised**) **learning**: used for anomaly detection. Until now I've been only considering unsupervised learning, but it is true that if, in the training process, there were enough instances of accidents, breakdowns and other anomalies, supervised learning could be used. Still, unsupervised learning models will be used to detect unusual patterns in traffic data that deviate from the norm. These models will be trained on normal traffic conditions, and deviations from this baseline will be flagged as anomalies. Clustering algorithms like **DBSCAN** can be used to identify outliers.

For all of this, data has first to be prepared. To handle missing data, forward or backward filling could be used. For outliers, like erroneous sensor readings, they can be identified and removed using statistical methods. Also, feature extraction is important in this. Vehicle counts, waiting times, and intersection types will be converted into features for the RL and time-series models. Temporal features like time of day, day of the week, and weather conditions will also be critical for traffic prediction models. And lastly, to ensure accuracy, the data has to be normalized for the different algorithms.

2.5.4. Experimentation & Validation

To validate the system, historical traffic data will be split into training and test sets (train-test split). For traffic prediction, metrics like the mean absolute error will be used to measure performance. For traffic optimization we can use the mean delay time to see if the system improves traffic flow. And in computer vision tasks we can track precision, recall and F1 score to ensure that vehicles and pedestrians are detected correctly.

Running A/B testing, some intersections will use the smart system (treatment), while others keep the old static system (control). From there, a comparison travel time, traffic congestion, and accident rates between the two groups can be studied to see the impact.

Key success metrics include better traffic flow and reduced delays, while guardrail measures will ensure the system doesn't unintentionally cause more accidents or other problems.

2.5.5. Human-in-the-loop

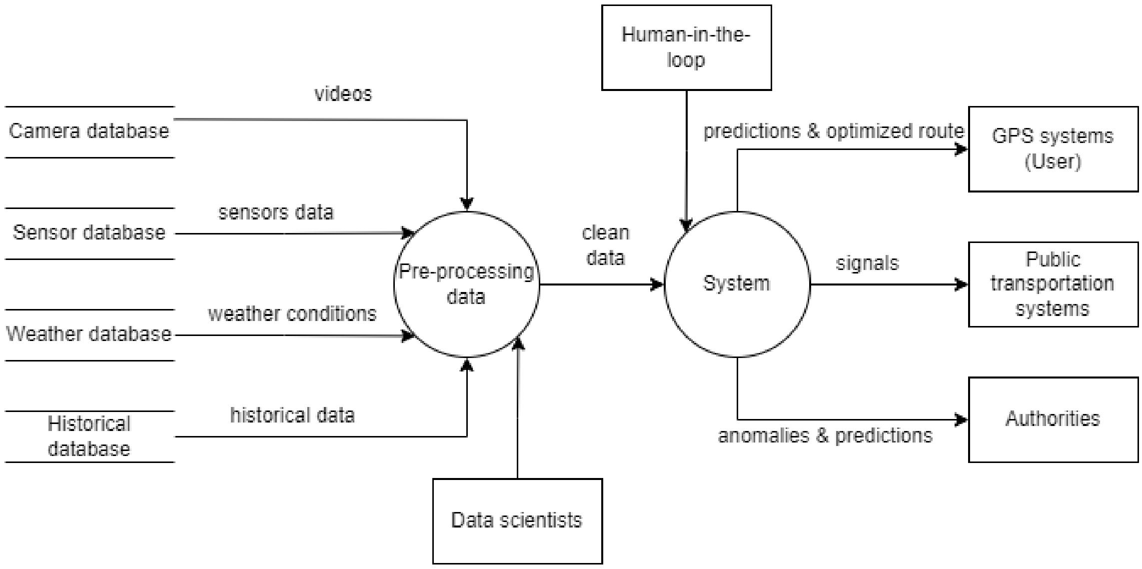
Due to the importance of the system working properly and the extreme negative consequences in case of malfunction, it needs to incorporate human oversight in some areas. For example, anomalies can be reviewed manually before alerting emergency services.

Additionally, drivers can have the option of overriding route suggestions, and authorities can manually control traffic lights in emergency situations.

Apart from all of this, continuous monitoring and feedback from everyday operators will allow the system to reduce errors, learn and improve over time.

2.6. Implementation

2.6.1. High-level design



2.6.2. Infra

The system will be held mainly on the cloud, as it has to connect to various sources to obtain the information and then spread it again once processed. It could still be considered an on-premise part of the system for the real-time traffic optimization. If everything was connected properly, all the information needed for each intersection is there; but this has more privacy problems and is not scalable. So, cloud storage will handle traffic data logs, while cloud-based compute services will run the ML models and real-time optimization algorithms.

2.6.3. Performance

Due to the real-time requirements, the system's latency is critical, and it also needs to handle high throughput. The system will be designed to scale horizontally by adding more servers or microservices as traffic data increases. By distributing the workload across multiple instances, the system can handle more data (higher throughput) without significant performance drops.

2.6.4. Security

The system will use tools like OAuth2.0 or JWT tokens to securely handle requests and keep the access safe. Traffic data obtained by cameras and sensors will be encrypted both while being sent and while being stored. To protect the system from unauthorized access and attacks, it will be behind a firewall and use tools like intrusion detection.

2.6.5. Data privacy

Since the system collects large amounts of data, including video feeds from public spaces, it will ensure working with GDPR or other applicable data privacy laws. This is sticking to strict rules regarding the collection, processing, and storage of personal data, particularly anonymizing sensitive data like license plates or faces. Also, if required, it could have data retention policies to automatically delete unnecessary data or delete certain data per request.

2.6.6. Monitoring & Alarms

To make sure that the system runs smoothly (as said many times before, errors may have big consequences) it will have monitoring with tools like Prometheus to track critical metrics such as system latency, traffic flow accuracy and sensor downtime. Automatic alarms when the latency gets to an adequate threshold will notify administrators and logs will be collected and stored to analyze them.

2.6.7. Cost

Costs will depend heavily on the cloud services chosen, data volume, and compute power needed. The computing costs with EC2[3] or some equivalent could rise to 1000 € per month for running the models in real-time. The data storage with S3[4] or some equivalent could cost around 300 € per month to store traffic videos and sensor data. And the data transfer (network costs) could approximately add 200 € per month. Added to all of this, some other costs could rise with database management and monitoring and alarm services. This is an estimation, as this is very variable due to different city sizes and complexities.

2.6.8. Integration points

This is an important matter in this system. It will need to integrate with already existing traffic cameras, radars and sensors. And on the downstream, it will need to send results, predictions and optimizations to GPS applications, authorities to monitor and manage anomalies, and public systems for real-time updates.

2.6.9. Risks & Uncertainties

There are some risks and uncertainties that add some complexity to the problem. To begin with, the clearer problem for everyday users is data privacy. Storing and processing camera videos and vehicle information could raise some privacy concerns. Also, as mentioned earlier, the system's reliability is very important. Being real-time, any downtime could lead to traffic congestion and even accidents. The model performance could raise some uncertainty, as machine learning models could struggle to handle extreme traffic conditions. And lastly, the system has to integrate in various cities and countries, and this implies working with different regulatory environments and entities, which could lead to more constraints and slowing the deployment.

2.7. Appendix

2.7.1. Alternatives

These are alternatives that I considered but decided not to use in the end.

- Developing a whole GPS system and infrastructure interconnected which would be the only option to use the part of the system which is personalized (optimized routes). This was discarded because imposing the application seems uncomfortable and not everybody would use it, so the benefits would reduce.

- Creating dynamic signs that would directly manage congestion situations. This would have been very complex, responsible and would have many government constraints. This is an authority's job, that with the explained system would be hopefully reduced (congestion won't happen as much), but the human touch is still important in many situations.

2.7.2. Milestones & Timeline

The first step is to gather the data and requirements, so the first milestone would be completing the requitements document and project plan (around 2 months). The next step would be making an initial system design, and the milestone to present a working prototype for a reduced area (around 4 months). After that, the development of the system itself starts, with AI modelling, backend development and integration with cameras and sensors. The milestone would be having functional models running in test environments (6 months). Next, testing comes into scene to measure improvements and refining systems (2 months). And lastly, the full deployment is made across a city and monitoring for long-term (around 5 months). After that, maintenance and optimization will be crucial.

2.7.3. Glossary

Definition of business or technical terms.

[1] GDPR. The General Data Protection Regulation is a legal framework that sets guidelines for the collection and processing of personal information from individuals who live in and outside of the European Union.

[2] P99. 99% of the requests should be faster than given latency.

[3] EC2. Provides on-demand, scalable computing capacity in the Amazon Web Services (AWS) Cloud.

[4] S3. Massively scalable storage service based on object storage technology in the Amazon Web Services (AWS)

2.7.4. References

Statistics and data consulted

https://www.tomtom.com/traffic-index/

https://etrr.springeropen.com/articles/10.1186/s12544-022-00554-1

https://www.gov.uk/government/publications/road-congestion-and-travel-times-statistics-guidance/travel-time-statistics-background-quality-report

https://gdpr-info.eu

https://medium.com/@rishabh011/what-is-a-good-p99-benchmark-7ec89de19559

Methodologies

https://www.influxdata.com/time-series-forecasting-methods

https://www.ibm.com/topics/computer-vision

https://en.wikipedia.org/wiki/Reinforcement_learning

https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-q-learning

https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

https://www.linkedin.com/advice/0/what-model-validation-how-can-you-do-effectively-xskjf

https://medium.com/@sujathamudadla1213/what-is-forward-and-backward-fillling-of-imputation-in-exploratory-data-analysis-eda-74ec33aca4bd

https://prometheus.io/docs/introduction/overview/

https://aws.amazon.com/es/ec2/

https://aws.amazon.com/es/s3/