GRADUATE PROGRAMS INSTITUTE

ROUTE OPTIMIZATION BY AVOIDING UNWANTED PLACES USING UNSUPERVISED MACHINE LEARNING

Muhammad Ahsan RASHID

MASTER'S THESIS

Istanbul, August 2025

ROUTE OPTIMIZATION
BY AVOIDING UNWANTED PLACES
USING UNSUPERVISED MACHINE LEARNING

Muhammad Ahsan RASHID

MASTER'S THESIS

COMPUTER ENGINEERING DEPARTMENT

COMPUTER ENGINEERING MASTER'S PROGRAM WITH THESIS

MASTER'S THESIS ADVISOR

Asst. Prof. Gizem TEMELCAN ERGENECOSAR

MASTER'S THESIS JURY MEMBERS

Asst. Prof. Inal Begum TURNA DEMIREL

Asst. Prof. Zeynep Behrin GUVEN AYDIN

BEYKOZ ÜNİVERSİTESİ

T.C.
BEYKOZ ÜNİVERSİTESİ
LİSANSÜSTÜ PROGRAMLAR ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı 2230140116 numaralı öğrencisi MUHAMMAD AHSAN RASHID'in "ROUTE OPTIMIZATION BY AVOIDING UNWANTED PLACES USING UNSUPERVISED MACHINE LEARNING" adlı tez çalışması Enstitümüz Yönetim Kurulunun ../../.... tarih ve 20../.. sayılı kararıyla oluşturulan jüri tarafından ile Tezli Yüksek Lisans tezi olarak edilmiştir.

Öğretim Üyesi Adı Soyadı	**İmzası**

Tez Savunma Tarihi

1)Tez Danışmanı: Dr. Öğr. Üyesi Gizem Temelcan Ergenecoşar

2) Jüri Üyesi: Dr. Öğr. Üyesi İnal Begüm Turna Demirel

3) Jüri Üyesi : Dr. Öğr. Üyesi Zeynep Behrin GÜVEN AYDIN

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

CONTENTS

ABSTRACT

ROUTE OPTIMIZATION BY AVOIDING UNWANTED PLACES USING UNSUPERVISED MACHINE LEARNING

This thesis examines how sophisticated route planning systems employ uncontrolled machine learning algorithms to avoid unpleasant sites users have indicated. The focus is on Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Traditional route algorithms aim to minimize journey time and distance. These approaches do not always include user preferences, safety concerns, or other subjective or situation-specific variables. This thesis developed and tested a novel route system that combines grouping and location data analytics to locate and avoid locations users do not want to go, such as busy neighbourhoods, casinos, pubs, and other unpleasant venues. DBSCAN can manage noise and locate any-shaped groups, making it ideal for this purpose. Even if undesired sites are unevenly distributed, it can analyze geographical data effectively. After defining density limitations based on user-defined parameters, DBSCAN finds dense clusters of locations based on their geographical proximity. This grouping strategy permits routes to be adjusted depending on data, unlike standard systems that require cutoff areas or zones. The thesis technique analyses the issue statement, revealing the requirement for bespoke route solutions that exceed efficiency criteria. The mathematical origins of DBSCAN and how the algorithm's core assumptions allow it to locate key geographical groupings in enormous volumes of data are also examined. The solution's design and route calculation using regional inputs, user choices, and grouping results are given. The system constantly recalculates the optimum routes to minimise exposure to these undesirable sites while fulfilling travel speed targets. The thesis evaluates system efficacy using experimental testing and scenario-based models. These tests evaluate how effectively the new pathways avoid key groups, how fast and accurately the clustering approach detects them, and how much they vary trip time and distance. The paper highlights its drawbacks, such as calculations being difficult and grouping variables being susceptible to urban contexts. This thesis reveals that personalised and context-aware navigation systems may leverage autonomous machine learning techniques like DBSCAN to determine routes, making them an exciting new subject to examine. The findings demonstrate the need of adding location-type-specific user constraints, which increase directing control and flexibility. Aligning pathways with users' preferred avoidance options makes them happy and allows for the creation of flexible navigation systems that can adjust to social and natural changes.

Keywords: Route Planning Systems, Unsupervised Machine Learning, DBSCAN, Spatial Clustering, User Preferences, Geographical Data Analytics, Personalized Navigation.

Date: August 2025.

ÖZET

DENETİMSIZ MAKİNE ÖĞRENMESİ İLE İSTENMEYEN NOKTALARDAN KAÇINARAK ROTA OPTİMİZASYONU

Bu tez, gelişmiş rota planlama sistemlerinin kullanıcıların hoşlanmadığını belirttiği yerlerden kaçınmak için denetimsiz makine öğrenimi algoritmalarını nasıl kullandığını incelemekte ve Gürültülü Uygulamaların Yoğunluk Temelli Uzamsal Kümelemesi'ne (DBSCAN) odaklanmaktadır. Geleneksel rota algoritmaları yolculuk süresini ve mesafesini en aza indirmeyi amaçlar ve bu yaklaşımlar her zaman kullanıcı tercihlerini, güvenlik endişelerini veya diğer öznel veya duruma özel değişkenleri içermez. Bu tez, yoğun mahalleler, barlar ve diğer mekanlar gibi kullanıcıların gitmek istemediği yerleri bulmak ve bunlardan kaçınmak için gruplandırma ve konum verisi analizini birleştiren bir rota sistemi geliştirmiş ve test etmiştir. DBSCAN'nin tercih edilme sebebi, gürültüyü yönetebilir ve her şekildeki grupları bulabilir olmasıdır. İstenmeyen mekanlar eşit olmayacak şekilde dağılmış olsa bile, coğrafi verileri etkili bir şekilde analiz edebilir. Kullanıcı tanımlı parametrelere göre yoğunluk sınırlamalarını belirledikten sonra, DBSCAN coğrafi yakınlıklarına göre yoğun konum kümeleri bulur. Bu gruplandırma stratejisi, kesme alanları veya bölgeler gerektiren standart sistemlerin aksine, verilere göre rotaların ayarlanmasına izin verir. Tez, sorun bildirisini analiz ederek verimlilik kriterlerini aşan özel rota çözümlerine olan ihtiyacı ortaya koymaktadır. Çalışma ayrıca DBSCAN'in matematiksel kökenlerini ve algoritmanın temel varsayımlarının, devasa veri hacimlerinde kilit coğrafi gruplaşmaları nasıl bulmasını sağladığını da incelemektedir. Çözümün tasarımı ve bölgesel girdiler, kullanıcı tercihleri ve gruplandırma sonuçları kullanılarak rota hesaplaması verilmiştir. Sistem, seyahat hızı hedeflerini korurken bu istenmeyen bölgeleri kullanmayı en aza indirmek için optimum rotaları sürekli olarak yeniden hesaplamaktadır. Tez, deneysel testler ve senaryo tabanlı modeller kullanarak sistem etkinliğini değerlendirmektedir. Bu testler, yeni yolların kilit gruplardan ne kadar etkili bir şekilde kaçındığını, kümeleme yaklaşımının bunları ne kadar

hızlı ve doğru bir şekilde tespit ettiğini ve yolculuk süresini ve mesafesini ne kadar değiştirdiğini değerlendirir. Hesaplamaların zor olması ve değişkenlerin gruplandırılmasının kentsel bağlamlara duyarlı olması gibi dezavantajlarını vurgulamaktadır. Bu tez, kişiselleştirilmiş ve bağlam farkındalığına sahip navigasyon sistemlerinin, DBSCAN gibi özerk makine öğrenimi tekniklerinden yararlanarak rotaları belirleyebileceğini ortaya koymaktadır. Bulgular, yönlendirme kontrolünü ve esnekliği artıran konuma özgü kullanıcı kısıtlamalarının eklenmesi ihtiyacını göstermektedir. Yolları kullanıcıların tercih ettiği kaçınma seçenekleriyle hizalamak sosyal ve doğal değişikliklere uyum sağlayabilen esnek navigasyon sistemlerinin oluşturulmasına olanak tanımaktadır.

Anahtar Kelimeler: Rota Planlama Sistemleri, Denetimsiz Makine Öğrenmesi, DBSCAN, Mekansal Kümeleme, Kullanıcı Tercihleri, Coğrafi Veri Analitiği, Kişiselleştirilmiş Navigasyon.

Tarih: Ağustos, 2025

ABBREVIATIONS

ACO	: Ant Colony Optimization Algorithm
AOA	: Arithmetic Optimisation Algorithm
API	: Application Programming Interface
DBSCAN	: Density-Based Spatial Clustering of Applications with Noise
GA	: Genetic Algorithm
GDPR	: General Data Protection Regulation
HDBSCAN	: Hierarchical Density-Based Spatial Clustering of Applications with Noise
HTTP	: Hypertext Transfer Protocol
KPI	: Key Performance Indicator
minPt	: Minimum point
OBL	: Opposition-Based Learning
OPTICS	: Ordering Points to Identify the Clustering Structure
OSM	: OpenStreetMap
POI	: Point of Interest
RL	: Reinforcement Learning
SA	: Simulated Annealing

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The real-time traffic updates, step-by-step instructions, and estimated arrival times provided by navigation applications like Google Maps, Apple Maps, and Waze are very helpful to users nowadays (Li and Yang, 2023). These navigation systems are dependable, fast, and simple to use, so billions of people have travelled securely in new and old places. Their algorithms optimize routes to save fuel consumption and carbon emissions, making travel quicker, shorter, and greener. Most of these algorithms assist everyone in navigating daily life. Utilitarian design prioritizes uniformity, efficiency, and maximum benefit (Bayliss, 2021). These key performance indicators (KPIs) illustrate how this design method supports the proposed navigation system. This structure is beneficial and successful, but it assumes all users have the same wants and objectives. Sometimes individuals forget how their identities, passions, and weaknesses impact how they utilize actual environments (ALawad and Kaewunruen., 2023). More theses and social activity are highlighting the issues with a universal digital system. For addicts, living near casinos, bars, or liquor shops might be deadly. What if the guiding system ignores these people's emotions and thoughts? It might take children to tempting or worrying locations. Due to their religious or moral beliefs, some may avoid specific venues. Highly pious Muslims may avoid pork and adult entertainment enterprises by going elsewhere. Orthodox Jews may have trouble reaching kosher eateries and other Sabbath-compliant destinations (Sriprateep et al., 2024). People pursue these programs for particular moral or spiritual difficulties, not to discover the shortest path.

Traditional guiding objectives may not benefit women and other disadvantaged groups. Many women prefer well-lit, busy, and safer highways, even if they are longer. This is particularly true for nighttime or solo trips. Studies on mobility and gender suggest that women are more inclined than males to adjust their travel habits for safety. Transgender and non-binary individuals may avoid violent or frightening settings. Disabled individuals may require ramps, elevators, and crossing signs. Traditional tracking systems concentrate on distance and speed instead of user group challenges, making them less helpful (Si et al., 2025).

This study suggests that user-defined location options are essential for a pleasant, fast, and inclusive digital assistance system. Flexible, user-centered, and user-specific navigation systems must replace efficiency-focused ones. Personalizing digital maps is not just a lovely treat—it might improve safety, community, and mental health. Navigation systems may better serve contemporary society if users can designate certain sites or venues to avoid.

The technology for customization improves day by day. Adding user-generated data, big data, and machine learning to guiding systems allows real-time route changes and reporting. These technologies might create user profiles that consider preferences or risks. Customers may identify regions as "avoidance zones" or exclude particular locales from search results. Maps might include community-driven safety data like crime rates, illumination, and abuse reports to assist vulnerable users in picking safer routes (Ang et al., 2022). The same data might assist communities in improving urban planning by revealing and resolving building safety issues in particular neighborhoods. Implementing these customizing options has issues. Technology, such as ensuring guiding systems can manage increasingly complicated route constraints and expand, comes first. System designers should avoid making customizing tools too complicated to discourage use, and instead, they should simplify technology (Akter et al., 2025). Morality and privacy are crucial. Collection and storage of confidential user data, including addiction, religion, and harassment, must meet strict data security guidelines. A satisfactory tailoring system requires authorization, transparency, and data ownership. Subjective or community-reported data may perpetuate prejudices or tarnish locations. For instance, labelling a community "unsafe" may reinforce racial and socioeconomic prejudices. Navigation systems should employ context-sensitive, open, and validated data sources to reduce such dangers (Sriprateep et al., 2024). They should also allow consumers to pick which information sources their route algorithms employ to avoid computational rigidity and make informed judgements.

User-defined geographical choices raise legislative, legal, social, and technical design challenges. Navigation systems have grown in importance for managing public movement and behaviour. Developers and policymakers should consider how digital infrastructure impacts privacy, safety, accessibility, and non-discrimination. Public transport planning, emergency response, and government services may be made fairer and more accountable by the public sector (Kim et al., 2022). A company may benefit from adaptive and inclusive guiding systems to attract underrepresented groups and new clients. Even more importantly, these advancements promote a cultural shift toward ethical technology and empowerment. Today's guidance systems are fantastic at making systems simple and fast, but their default optimization techniques do not take into account our goals, values, and weaknesses. Navigation tools that are more personalized and accessible to all users are moral and technological criteria (Priyadarshi, 2024). Consider and accommodate the spatial preferences of individuals with varied demands, such as addicts, religious or ethical adherents, women, and vulnerable groups. Such considerations will help developers create safer, more courteous, and more powerful digital experiences. Guidance systems should now prioritize individual requirements, privacy, and value.

1.1 Problem Statement

Standard navigation tools do not provide functionality to avoid entire categories of places. Although Google Maps API allows setting specific waypoints or avoiding highways/tolls, it does not inherently detect and avoid unwanted locations along a route.

To create a pressing need for a machine learning-driven approach to:

- Detect concentrations (clusters) of unwanted places,
- Classify regions as risky or unwanted,
- Adapt routes dynamically to avoid those clusters.

Despite their improved technology, navigation systems cannot yet manage each user’s specific geographical restrictions. Currently, users who wish to avoid sites for moral, psychological, or safety concerns have minimal help. These options vary by individual, environment, and season. Not having these characteristics worsens present route techniques and the exclusion of special needs persons. Users that walk through regions they do not want to may suffer mental discomfort, become more sensitive, or recall terrible memories due to this lack of freedom. Not all applications help users avoid toll roads, bridges, and highways. User-created groups may avoid drug businesses, adult entertainment venues, and high-crime zones.

1.2 Objectives and Goals

This thesis seeks a new definition of “optimal navigation”. When designing a system, consider the user’s physical and mental health as well as speed and distance. This thesis changes optimum advice criteria to include user-specified geographical restrictions. These standards do not modify guidance’s core aims; they improve them by aligning technological outputs with actual people’s values and goals.

Two key reasons make this thesis relevant: first, it gives guiding instruments a moral and friendly foundation. People who could not obtain such navigation options in the traditional navigation system now can have the opportunity to use the distinctive features. Second, this navigation system offers a smart, scalable system that can adapt to user demands without impacting performance or generalizability. This greatly advances geoinformatics and spatial computing.

The major goal of this thesis is to construct a guiding system that allows people to avoid certain regions. To assist, the thesis outlines these goals:

- It aims to utilize books and real-life examples to classify sites that various consumers avoid.
- DBSCAN autonomous machine learning is used to discover unwanted dense clusters of venues.
- A real-time guiding system that considers these categories is the concept. This lets automobiles avoid particular areas by taking other routes.
- Case studies or models should be utilized to evaluate the proposed system's functionality, growth, and user satisfaction.

These aims enable transitioning from general to individualized advice, making technology more helpful and benefiting society.

1.3 Scope and Limitations

This thesis aims to improve guiding systems by introducing user-defined geographical restrictions using unsupervised machine learning. It details how to utilize DBSCAN clustering to locate user-requested avoidance zones and apply them to custom routing. These zones may include bars, high-crime districts, and adult venues. Due to the variety of urban venues, the thesis focuses on this response. It can be easily adapted to various locations and scenarios with adequate location data. The scalable and portable architecture allows this. Despite its limitations, the thesis deserves some mention. First, location-based data like place ratings and crime statistics are crucial to the system. When data is scarce or incorrect, the grouping approach may not operate properly. Second, time and morality prevented it from receiving user data or feedback in real time. Thus, it does not assess the model's response to rapid occurrences or transitory risks. Instead, estimate or duplicate user intent. Finally, DBSCAN performs well in static groups but may struggle in dynamic situations without frequent data.

The speed of computation limits route method changing. Extreme avoidance of high-density zones in crowded regions may lead the system to recommend unsuitable escape routes, even if its objective is to offer alternatives. Finally, licensing restrictions prevent the model from combining with commercial mapping system APIs like Google Maps or Apple Maps; therefore, it is still under testing.

1.4 Tools Used in Thesis

This thesis makes use of a variety of computer and location techniques. These tools and systems matter:

Python Programming Language: Python offers many data science, location analysis, and machine learning capabilities; thus, it is utilized to create the grouping and routing algorithms (Dhruv et al., 2021).

Scikit-learn: This application enables DBSCAN grouping. It allows for trialling varied epsilon (ε) and minimum sample quantities to optimize cluster finding (Miranda et al., 2023).

Pandas and NumPy: When dealing with large volumes of location- or venue-based data, these technologies clean, change, and prepare data.

Folium and GeoPandas: These technologies display organized data and travel routes on live maps. They help to understand the findings' context (Candra, 2025).

OpenStreetMap (OSM) Data: The primary source of geographic data utilized to evaluate the guiding model is OSM, which has an open-access policy and clearly names venues (Grinberger et al., 2022).

NetworkX: This Python package creates graph-based networks like city maps and provides shortest-path algorithms like Dijkstra's and user-defined avoidance zones (Hadaj et al., 2022).

Jupyter Notebooks: It is used to record code, findings, and conversations for reuse (Samuel and Mietchen, 2024).

These technologies enable repeatable theses using regional data, unsupervised learning, and route planning.

1.5 Originality

This thesis approaches digital guiding by merging machine learning without human supervision with user demands. It modifies how navigation applications choose the optimum route by taking into consideration the user's avoidance options, something industry standards do not always do. New concepts include adding custom restrictions and utilizing DBSCAN to discover avoidance zones without tagged data or prior training sets.

The thesis addresses real-life moral and societal issues, making it essential. It is understood that people view and engage with actual environments differently. It advocates for more compassionate and inclusive space and transportation choices by demonstrating a user-sensitive, scalable travel strategy. There is not much academic thesis on density-based clustering for real-time map monitoring, and future iterations of this study's technique may contain more behavioural or environmental filters. After commercial license and API constraints are resolved, the thesis's adaptable architecture might integrate with current mapping systems, demonstrating cutting-edge technology.

1.6 Methodological Approach

DBSCAN and other autonomous machine learning are crucial to this thesis. Without predefined training sets or labelling, this density-based clustering approach can locate any geographical group. It may help to locate dubious spots like neighbourhood pubs, high-crime regions, and user-preferred venues. DBSCAN finds clusters by number of data points. It distinguishes "noise" (spots without clusters), "core points" (points with enough friends), and "border points" (points near a cluster but not core). With this much knowledge, the model can spontaneously build groupings that make sense and are consistent everywhere. After that, these navigation groupings may be added to a base plan to create route "avoidance zones".

The thesis recommends adding avoidance zones as hard or soft route constraints to A* and Dijkstra's shortest-path algorithms. The pathways are altered to avoid these locations while remaining short and fast. This combination helps the computer balance user preferences and guiding needs.

1.7 Thesis Structure

Chapter 2 explores existing scholarly work and current technological approaches related to navigation systems, unsupervised machine learning (particularly DBSCAN), and user-centric route planning. It identifies gaps in current research and helps position the present study within the broader academic and technical context.

Chapter 3 outlines the overall research design and methodological framework adopted in the thesis. It details how data was collected, the rationale behind sample selection, and the tools, technologies, and algorithms used to build and test the proposed system. It also addresses data analysis techniques, validation methods, limitations, and ethical considerations of the study.

Chapter 4 presents the design and architectural structure of the proposed intelligent routing system. It explains the system's objectives, feasibility, and internal components, including input, processing, and output models. The chapter also includes database and security designs, providing a comprehensive breakdown of how the system was built and the environment in which it operates.

Chapter 5 discusses the outcomes of the research and evaluates the system's performance against the initial objectives. It presents key findings, offers practical recommendations for future improvements or real-world implementation, and concludes the study by summarizing its significance, limitations, and potential directions for further research.

2. LITERATURE REVIEW

Route optimization, which seeks the fastest and most direct path, is a major thesis field in computer science, GIS, and ITS. Historically, Dijkstra's and A* search algorithms were employed. These methods rank journey time and distance. Despite working well for speed, these traditional navigation methods cannot manage more complex, user-specific challenges. Modern route design considers behaviour, social norms, safety, and morality. Understanding and adapting to the environment are getting increasingly crucial. This tendency is changing as route planning systems that avoid certain sites become more important (Abideen et al., 2023). Avoiding pubs, casinos, and other venues that are regarded to be incompatible with personal or religious beliefs is one example of how these venues may vary considerably based on personal choices, cultural perspectives, or health-related concerns. Subjective and qualitative boundaries are beyond ordinary algorithms' capabilities. Therefore, data-driven techniques, particularly those based on autonomous machine learning, are increasingly often employed to create transport systems that better meet consumer demands (Ahmad et al., 2020).

Unsupervised machine learning finds data patterns and structures without naming findings. Clustering algorithms are crucial in this system since they group data points by similarity. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) promises for route planning applications that need to find and remove unnecessary locations. The capacity of DBSCAN to detect groups of any type and distinguish noise from outliers is beneficial for this system, particularly when attempting to isolate regions with many undesired spots from the surrounding noise. Low-density zones indicate noisy sites, whereas DBSCAN groups nearby data points such as clubs or gambling venues. This makes it ideal for urban spatial analysis, where POIs might be scattered or aggregated (Bajaj et al., 2020). The information may allow routing systems to construct routes that avoid certain groups on the fly. Users get a better customised travel experience that considers their preferences.

Route planning today employs DBSCAN, which combines several thesis fields. Such as ethical AI, human-centred computing, and location data mining. This represents a shift towards algorithms that consider consumers' social and emotional states rather than merely the shortest or cheapest path. Guidelines may be utilised for several purposes by incorporating inclusivity and well-being (Ali et al., 2024). They may assist addicts in avoiding triggers and keep youngsters and the elderly safe by avoiding crime-ridden areas. Ideas for such strategies originate from many diverse fields that do not usually collaborate. Road networks are still often seen as graphs with roads as lines and junctions as dots; this is known as graph theory. While adding user-defined avoidance zones or other subjective boundaries to this network topology, computer performance suffers. To simplify unnecessary processing, clustering might be employed initially. It may teach the pathfinding algorithm to penalise or avoid pathways via "soft barriers" or tiny network regions in these situations. Other clustering systems beyond DBSCAN have been the subject of much research (Bank of England., 2022). Hierarchical clustering, K-means, and Gaussian mixture models are examples. These approaches are not practical in complicated urban situations where POI locations are not regular or predicted since they need circular or ellipsoidal cluster forms and a specified number of clusters. DBSCAN finds user-defined problematic areas even if they do not follow geometric patterns because it finds noise and is non-parametric.

Complex route planning models have many real-world applications. User preferences help mobile guiding applications, urban planning tools, emergency response systems, and self-driving vehicle routes make judgements. Religious users may avoid regions with bars, while parents may wish to keep their children away from fun places. These application scenarios demonstrate the need for adaptable AI systems for social and moral contexts. Problems must be resolved before these systems may be implemented. User-defined undesired zones are hard to categorise and detect across regions, making labelling and data collection difficult (Dixon et al., 2020). Collecting and exploiting user preferences raises privacy concerns, particularly when the choices include religion, health, or personal tragedies. Ethics are crucial while designing these systems to secure data, user consent, and transparency. Considering that cityscapes change constantly. Locations might be safe or suspicious depending on the user's mindset, time of day, and surroundings. Time complicates things; thus, it may require learning systems that can adapt to changing user conditions and real-time data streams. Researchers may investigate mixed models that combine unstructured grouping with RL or probabilistic reasoning in the future to improve route planning systems (El Hajj and Hammoud, 2023).

Route design evolves swiftly to meet consumers' evolving demands. With grouping approaches like DBSCAN, learning without being monitored has improved. These strategies assist users in identifying and avoiding unwanted situations, improving guiding systems' originality and social relevance. As this subject expands, researchers and authors must consider the practical, moral, and scientific issues raised by new technologies. It ensures that it is transparent, honest, and fair to all users. In recent years, fundamental approaches, key improvements, and novel applications have altered unsupervised learning route planning (Fahle et al., 2020). This literature review expands on these themes.

2.1 Traditional Route Optimization Techniques

Route optimisation has long been significant in transportation, shipping, and guiding systems. The purpose of these systems has always been to discover the fastest and most direct route. The A-Star Algorithm and Dijkstra's Algorithm are long-standing, predictable algorithms. Both are considered essential conventional route planning strategies. The Dutch technological researchers' (Jayathilaka and Park, 2022) algorithm was one of the first systematic approaches to identify the fastest path between points in a graph, such as a road network. This approach ensures that every network point may be reached by the fastest route if the edge weights in a weighted graph are not negative. It begins with the fastest route between two places and adds until it discovers all paths. Dijkstra's algorithm has no rules; therefore, it may find worthless or suboptimal pathways. Its high computational power makes it difficult to employ on big networks (Kaponis and Maragoudakis, 2022). In real life, temporal complexity makes things inefficient; hence, the A* Algorithm was created. This approach employs a prediction function to estimate how much it will cost to go from a specific place to the objective, making it better than Dijkstra's. This rule of thumb, often known as the Manhattan or Euclidean distance, directs the search to fewer nodes closer to the target. The A* method ensures that the synthesised cost from the current node to the target matches the true cost from the beginning. Sparse graphs and huge regional datasets benefit from this speedup (El Hajj and Hammoud, 2023).

A* and Dijkstra's algorithms use strong mathematics to determine the fastest pathways. Their fault is that they do not consider subjective or qualitative aspects. These fundamental models are easy to anticipate since they employ a fixed set of characteristics, including distance, speed, and trip cost. They cannot be changed to avoid user-selected regions or reflect real-time events like traffic, weather, road closures, or safety concerns. Due to their rigidity, these systems believe the shortest or simplest route is always preferable and work "one-size-fits-all". Users may choose comfort, safety, cultural significance, and ease above functionality in real life (Kapoor et al., 2023). This is particularly true in cities and during severe crises. Riders should avoid busy places and steep climbs like nighttime walkers. Traditional algorithms cannot manage such nuanced preferences since people must reprogramme and alter their cost functions. To address these issues, heuristic-based and metaheuristic approaches were utilised to adjust the system. These algorithms include GA, ACO, and SA. These strategies are great for optimising complex tasks with several objectives or restrictions. They simulate natural selection by creating many solutions, choosing the best, and combining them to build better ones (Kumar et al., 2022). This technique may scan a larger region, but it may need more processing power due to more challenging and various criteria. ACO models ant colony food search using phoney "pheromone trails". Pheromones on frequented roadways may assist guide searches over time. These approaches use self-learning and adaptation to create pathways for several goals, unlike fixed systems.

Despite these heuristics, major issues remain. Due to the reliance on numerical inputs and static data files, many of these approaches cannot adjust to changing conditions or user preferences. Large, real-time networks like those in smart cities may struggle to thrive due to changing traffic and user interests. How to include emotional comfort, cultural meaning, and personal experience into growth is unclear. It always has to choose between speed and accuracy when using heuristic or metaheuristic approaches. These algorithms seek a "good enough" or best local solution in a reasonable time, not a global solution. For many important uses, including utilising GPS to locate the way or preparing an emergency route, this trade-off is acceptable. But not ensuring optimality might be problematic in high-stakes circumstances like disaster assistance or war operations. Sharing rides, blending modes, and leveraging user-generated data complicate urban mobility networks. This requires more personalised and interactive route solutions. This has highlighted the shortcomings of conventional route design (Fahle et al., 2020). Though effective beginning points, they do not manage social, behavioural, and environmental aspects of mobility well, particularly when users desire greater control over their navigation.

Conventional navigation methods cannot manage unorganised data like social media comments, community-based threat alerts, and natural language inputs, making them faulty. Bus riders may comment a street is "unsafe at night" or a route "feels tic", and deterministic models require a lot of preparation and structured modelling to interpret and integrate these subjective inputs in their decision-making (Lei and Cailan, 2021). Modern routing systems were built on heuristic approaches like A* and Dijkstra's algorithms. These approaches are stiff and need static, structured datasets, making them less helpful in practice. Smarter systems that aggregate data, comprehend user desires beyond hard-coded metrics, make real-time modifications, and provide tailored, situation-aware recommendations are needed more. Therefore, artificial intelligence, machine learning, and natural language processing have been employed extensively in the route planning thesis. These algorithms may adapt to user behaviour, find patterns in data, and improve route suggestions over time. These new technologies, which fill holes left by earlier ones, will enable better, user-focused assistance systems.

2.2 Incorporating User Preferences in Routing

Since digital navigation technology is growing so fast, users desire personalised, user-focused route solutions. Simple settings in Google Maps, Waze, and Apple Maps allow users to avoid toll roads, highways, and boats. These options provide some independence, but they do not fulfil users' individual lifestyle, work, or security demands. User preferences in routing aim to move away from a rigid approach and towards one that adapts to the user's circumstances and objectives. A shipping firm may have different aims than a commuter who wants to avoid high-pressure places like roads or bridges with little room. As with hazardous, polluted, or desirable locales, people might avoid this scenario in several ways. Even though these complex demands are becoming increasingly apparent, advice tools do not properly account for them. To address this gap, years of academic theses have explored cutting-edge techniques to customise routes. Vineeth et al. (2020) argue that domain-specific applications like marine and air guiding should integrate more complicated user options in route planning.

Route design is vital for more than simply convenience, according to the thesis. Also included are safety, obeying regulations, and conserving money. Pirate sites, ecologically sensitive regions, and places with limited waterways must be avoided in coastal areas. Vineeth et al. (2020) argued for domain-specific applied clustering methods to manage massive user and environment data to provide more flexible route options. It would enable dynamically finding and marking unwanted locations.

K-means or DBSCAN may group and exclude regions based on user behaviour or standards for finding regional data patterns. Grouping algorithms may propose comparable routes to new users even if they have not requested them, as when many users avoid a given metropolitan region. Dynamic learning might be strengthened by adding machine learning, allowing systems to adapt to user interactions. A system that predicts based on user preferences and data is more likely to be customised. Today's guide systems let users pick their ideal surroundings depending on loudness, visual value, traffic, and even proximity to companies and places. After that, these qualities may automatically alter routing methods in real time, delighting consumers. If a passenger requests a more picturesque journey, more than one may be transported via parks or seaside areas, even if it takes longer. The system may also learn from users' preferences, such as their desire to avoid school zones due to noise or traffic (Kureljusic and Reisch, 2022). Despite these promising advances, preference-based routing is still far from being deployed. There is a trade-off between speed and user satisfaction. Complex tuning is required to accommodate several, often competing, user preferences. This may increase costs and time, slowing real-time navigation. Putting accurate, current information together is another issue. Unfortunately, the guiding system requires reliable and consistent location data to make sensible recommendations based on user preferences.

Data collection and storage regarding user choices worsen privacy problems. Personalising routes may improve user experience, but it requires monitoring and researching user behaviour and location. To address these issues and ensure ethical data processing and compliance with the GDPR, comprehensive data governance mechanisms should be implemented. These systems should protect user data and provide permissions. Routes that include user preferences affect more than simply personal travel (Pundir et al., 2020). Route selections may reveal how people move and what infrastructure is required to urban planners and smart city developers. If enough individuals declare they wish to avoid a given neighbourhood or intersection, urban planners may investigate noise, traffic, or safety problems and find solutions. Personalised routes benefit society, infrastructure, and people. Route algorithms now include user preferences, a major advancement in guide technology. Current systems have some rudimentary personalisation capabilities, but they do not cater for most user desires. The findings of Sharma et al. (2022) support satisfying this demand, particularly in confined transit scenarios. Grouping and machine learning provide data-driven customising potential. Real usage requires careful consideration of computer performance, data security, and privacy. The ability to accommodate consumers' more unique preferences will certainly distinguish future guidance system route innovations.

2.3 Clustering Algorithms in Spatial Analysis

Clustering algorithms have been important in geographic data analysis for years because they enable researchers and analysts to uncover geographical data structures, patterns, and relationships. These algorithms allow local data points to be organised more similarly than in other such groupings. Finding these groupings is crucial for urban planning, environmental monitoring, transportation theses, and resource allocation. Common location analysis grouping algorithms include K-Means. It divides data into K groups by centre using rules. It determines the nearest cluster centre for each data point and lowers cluster difference. K-means is popular since it is fast and easy to use. It works well for clustering in many real-life circumstances, particularly when data is spread out circularly and the number of groups is known (Shilong, 2021). However, K-means has many key drawbacks that make it unsuitable for many spatial analytic applications. Because exploratory spatial analysis requires the user to select the number of clusters K ahead of time, it may be difficult to determine how many relevant groups there are. Second, K-means implies groups are convex and the same size; therefore, it does not perform well with strange spatial patterns or levels. K-means is sensitive to mistakes and noise in real-world location data. In complicated or diverse geographical locations, these characteristics generally produce poor grouping results.

Researchers are increasingly employing Spectral Clustering to overcome these navigation issues. It is a graph-based approach that leverages data-derived similarity matrix eigenvalues and eigenvectors. Using spectral clustering instead of distance measurements creates an affinity network (Ullal et al., 2021). Notes are network nodes, and lines illustrate how similar two sites are. The network is separated using spectral characteristics. K-Means is used to group in the smaller space obtained by lowering dimensions using the Laplacian matrix. Spectral grouping aids the location thesis in several ways. Spatial data excels in non-convex and complicated groupings. This data includes natural terrains, metropolitan regions, and transportation networks. It can handle files with covered or layered patterns, making it more helpful. These properties make spectral clustering ideal for social-spatial network theses, ecosystem identification, and picture segmentation. This is because spectral grouping is limited. Computing is challenging, particularly when assembling and disassembling huge similarity vectors (Ran et al., 2023). High-resolution remote sensing and GPS-enabled equipment are becoming increasingly common, making the approach less scalable for large global datasets. Spectral clustering allows users to set a similarity function and group size. These selections may affect results quality and clarity.

Spatial analysis uses several clustering algorithms. These include K-means, spectral clustering, hierarchical clustering, model-based GMMs, and density-based DBSCAN. These algorithms' strengths and downsides depend on location data and thesis aims. Clustering algorithms still assist spatial data analysts in uncovering interesting groupings and regional trends (Bhattacharjee and Mitra, 2021). K-means and spectral clustering, two popular location analytics algorithms, are constantly improved. However, professionals are working hard to improve our approaches and create new algorithms to handle the increased complexity of size, shape, and noise.

2.4 DBSCAN: A Density-Based Clustering Approach

DBSCAN is a popular unsupervised machine learning approach for complex datasets with fluctuating densities. Choi and Hong (2021) founded DBSCAN. Data mining and spatial analytics still use it to locate any group and distinguish random spots. DBSCAN does not require knowing the number of circular clusters like partition-based algorithms like K-Means. This improves real-world flexibility and reliability. DBSCAN considers two crucial factors: epsilon (ε), the radius of the neighbourhood surrounding a point, and minPts, the minimum number of points required for a dense zone. These criteria group nearby data points. Core points are defined as at least minPts more points within a radius ε. Middle nodes keep groups together (Hu et al., 2021). The edge point is at least probability (μ) from the core point but not distant enough to fulfil the minimum Pts criteria. Points that do not meet the core or boundary guidelines are labelled "noise" to remove peculiarities and superfluous data. The computer repeatedly scans the collection to advance. The DBSCAN method creates a cluster with density-reachable locations within ε and meeting the minPts requirement. The algorithm repeats this if the point has not been visited and matches core point requirements. This approach is repeated until all density-accessible areas are located. Border points, or noise, are points that do not fulfil core point standards (Cheng et al., 2023). This dynamic, data-driven approach means DBSCAN does not force information into a style. Instead, it can adapt to data structure.

DBSCAN's ability to detect any-shaped groups is remarkable. Not all data points line up straight; therefore, this is particularly useful for geographical or spatial data. K-means and hierarchical clusters struggle to describe coastline, road, and urban sprawl architectures. DBSCAN can handle untidy, out-of-place real-world data since it does not get influenced by noise. DBSCAN excels in geography and route finding. This software seeks to optimise city walking and delivery truck travel. Route planning may be wasted because traditional grouping algorithms group vacant or crowded locations incorrectly. DBSCAN can distinguish sparse or busy regions that should not be considered from dense groupings that indicate popular zones or sites. Logistics phrases for busy or delayed areas include "noise". Transportation algorithms may concentrate on what matters most. DBSCAN can also remove unwanted areas when choosing tracking systems or smart city infrastructure (Bushra and Yi, 2021). Bad data points like accidents and delays might be provided for traffic congestion, construction zones, and risky junctions. DBSCAN finds these zones well since it is adept at detecting groups. This may aid in circumnavigating them. For logistics or marketing, excellent data groupings like consumer activity hotspots or efficient transportation routes might assist.

DBSCAN parameters may be understood, another benefit. Choosing the appropriate ε and minimum points, minPts, values for regional datasets needs topic expertise or experiment adjustments. ε determines a data point's "reachability", and minPts the area's lowest bar. By adjusting its parameters, DBSCAN may simply switch dataset densities. This eliminates the requirement for manual labelling or cluster count theories. However, DBSCAN has limits. A significant issue is the reliance on ε and minPts quantities, particularly in datasets with uneven density. With a low ε value, points may be misinterpreted as noise, resulting in the breakdown of large groups into smaller ones. If too high, cluster joining may be unsuitable. In the same manner, minPts must be carefully set; low numbers may include noise, while large values cannot develop in sparsely populated areas (Bhattacharjee and Mitra, 2021). Heuristic scaling and k-distance graphs may assist, but humans must decide the result. DBSCAN works best with low-to-intermediate-dimensional data. It may struggle in regions with many dimensions due to the "curse of dimensionality", which makes distance appear useless. DBSCAN outperforms several other algorithms for organised, geographic, and temporal datasets in interpretability, flexibility, and noise stability.

When data is noisy, unusual, and cluster forms are ambiguous, DBSCAN works better than typical clustering algorithms. It may be used for route planning, regional analysis, and other situations where understanding how data spontaneously clusters without assumptions is vital. Data-driven cluster construction based on density replaces physical assumptions. DBSCAN can identify significant groups and outliers, helping individuals make better judgements in many domains (Prasad et al., 2023).

2.5 Enhancements and Variants of DBSCAN

One of the most popular clustering algorithms, DBSCAN, is not ideal. Despite its issues, it can detect any cluster and manage noise. DBSCAN's parameter dependence is a major issue. Minimum cluster points (MinPts) and area radius parameter (ε, or epsilon) are required. Because they are fixed worldwide, these variables do not operate well on datasets with varied group densities or sizes (Al-Batah et al., 2024). Experts have found several methods to improve DBSCAN to overcome these issues. The new approaches enhance the algorithm for improved performance on larger datasets with more complicated structures.

2.5.1 OPTICS (Ordering Points to Identify the Clustering Structure)

Yang et al. (2025) added OPTICS, which improved DBSCAN. The sensitivity of DBSCAN to ε was resolved by creating OPTICS. OPTICS prioritises data points based on reachability distances rather than tight grouping for a certain ε. With this arrangement, you may generate groups with varied densities while retaining the data's density-based structure.

The reachability position of each data point is shown using OPTICS. Slopes reveal clusters, while peaks show noise or cluster edges in this graph. Find stacked and hierarchical groups without committing to a particular ε value using this approach. OPTICS provides a more flexible and full data structure representation, which is particularly useful for datasets with varying densities.

2.5.2 HDBSCAN

Amalina et al. (2024) created a stronger DBSCAN. Since it hierarchically groups DBSCAN, HDBSCAN is superior. Start by making a basic spanning tree with data points using the node gaps. Lines with increasing distance values are progressively removed to reflect shifting density restrictions. This builds group structures. The last phase in clustering is selecting the most stable groupings, which remain constant across many conditions.

HDBSCAN's ability to accommodate large gatherings is one of its strongest features. While DBSCAN may struggle to identify structures with only one ε number, HDBSCAN automatically looks for them. No need for the ε option; a minimum cluster size suffices. Because the minimum cluster size is easy to predict or alter. Because it works effectively with noisy, high-dimensional, or spatial-temporal data, HDBSCAN is popular in science and industry.

2.5.3 DBSCAN++

(Jang and Jiang, 2019) launched DBSCAN++ in 2019 to speed up and scale the original DBSCAN technique. The traditional DBSCAN approach requires several range queries to detect density around each location; therefore, it may take a long time. Especially for huge datasets. Data structures like k-d trees or approximate closest neighbour approaches let DBSCAN++ discover core points faster and remove density ratings.

DBSCAN++ decreases distance computations by improving area computing. This simplifies large-data or real-time data analysis. The enhanced scalability of DBSCAN++ allows density-based clustering to be applied on larger datasets in biology, location analysis, and social network mining with the same results.

2.5.4 Recursive-DBSCAN

Another version, Recursive-DBSCAN, improves DBSCAN on big and complex datasets (Bujel et al., 2018). This approach uses iterative DBSCAN to separate big groups that may have been incorrectly grouped due to a single global ε value. DBSCAN initial grouping may bring together substructures that might appear better as clusters with minor density differences.

To identify initial clusters, a wide ε is employed in the repeated process. Within each cluster, DBSCAN is repeated with increasingly accurate settings. This technique speeds up planning, route management, trajectory data processing, and internal substructure visualisation. Recursive-DBSCAN is often used to solve vehicle routing problems (VRPs) because clustering client addresses by physical proximity may enhance routing efficiency.

2.5.5 Comparative Analysis and Application Domains

Each DBSCAN update addresses a separate issue with the previous system. Comparative research demonstrates that OPTICS works well with datasets of varied densities but needs additional effort to get particular groupings. However, HDBSCAN's clean grouping algorithm works well in real life and accounts for noise (Shiri et al., 2023). DBSCAN++ speeds up and simplifies the procedure for large data sets. To simplify complex groupings, Recursive-DBSCAN polishes DBSCAN findings.

These versions are utilised in many places. Geospatial data analysts employ HDBSCAN and OPTICS to discover anomalies and hotspots. Due to its noise-free nature, HDBSCAN is excellent for clustering gene expression data in biology. Shipping and transportation employ Recursive-DBSCAN to set up delivery sites and discover the optimum routes. These algorithms are also used to detect social network groupings, anomalies, and weird behaviour patterns in hacking theses.

2.6 Parameter Optimization in DBSCAN

DBSCAN approach is notable for finding any-shaped groups and managing noise. The algorithm is prone to errors because of two key input parameters: epsilon (ε), the area radius surrounding a data point, and MinPts, the minimum number of points required for a dense zone (Zhang and Zhou, 2023). These characteristics greatly impact grouping. Incorrect ε selection might cause groups to become overly dispersed or too clustered. Use the incorrect MinPts number, and it may not locate enough or too many core points, changing the groups' meaning and structure.

Manually tweaking ε and MinPts is time-consuming and unsuccessful for big or complicated datasets in real-world applications. Analysts employ field knowledge or experiment to find what works. However, these algorithms may not work for all data. This has spurred theses on automating DBSCAN's parameter-selecting procedure to improve reliability and utility.

2.6.1 Automated Parameter Selection Techniques

In recent years, several innovative techniques have been developed to address this efficiency issue. These strategies enable the DBSCAN algorithm to cope with numerical, geographic, and high-dimensional datasets and reduce human intervention.

2.6.1.1 Deep Reinforcement Learning for DBSCAN (DRL-DBSCAN)

Deep Reinforcement Learning (DRL) can automate parameter selection, and Zhang et al. (2022) introduced DRL-DBSCAN. It represents parameter selection using a Markov decision process. Reinforcement learning is used to determine the optimal strategy for categorising dataset attributes using DBSCAN parameters (ε and MinPts). Selecting parameter pairs (ε, MinPts) defines agent behaviour in an environment with several datasets. A grouping quality metric like the Silhouette Score or Davies-Bouldin Index determines the prize. The algorithm learns the appropriate grouping parameters by going through several rounds and receiving benefit-based feedback. In experiments, DRL-DBSCAN performed well on noisy, mixed-density, and non-linear datasets. It outperforms k-distance graphs and other algorithms on numerous test datasets because it learns on the go. DRL-based approaches allow for continuing learning, so the model may improve its decision-making as it meets fresh data.

2.6.1.2 Metaheuristic Optimization: OBLAOA-DBSCAN

Metaheuristic algorithms have improved DBSCAN parameters in the thesis. These evolutionary and natural process-based methods may assist in locating the optimum solutions in a broad and challenging parameter space. OBLAO-DBSCAN is famous. It optimises DBSCAN using two sophisticated learning approaches. Its Arithmetic Optimisation Algorithm (AOA) follows simple maths. However, Opposition-Based Learning (OBL) fosters thesis by comparing current outcomes to their "opposites" to avoid early convergence (Huang et al., 2025). This approach generates a population of parameter pairs for each round using math operators and OBL concepts. New candidates are produced using the best parameter settings, and clustering quality criteria evaluate their performance. OBLAOA-DBSCAN outperforms basic DBSCAN in cluster purity, efficiency, and noise control. Real-world statistics on picture identification, biological data, and social network graphs demonstrate its versatility.

2.6.2 Additional Optimization Strategies

DRL, metaheuristics, and other methods for automated DBSCAN parameter adjustment have been suggested:

2.6.2.1 Grid Search and Cross-Validation

Even though they are taxing on computers, comprehensive search strategies that function across parameters may work for small- to medium-sized datasets.

2.6.2.2 K-Distance Plots and Knee Detection

The usual technique calculates the distance between each point and its kth closest buddy, using the curve's elbow to determine ε. It seldom works with complex or multisided datasets.

2.6.2.3 Genetic Algorithms (GA) and Particle Swarm Optimization (PSO)

These evolving algorithms mimic nature or culture to fine-tune parameter values. They function, but their meta-parameters must be adjusted, making things more difficult.

2.6.2.4 Bayesian Optimization

It may utilise chance to create a goal function model that works on its own and choose parameter values that may improve clustering. Clustering expensive functions is ideal using this method.

The current parameter enhancement thesis is crucial for DBSCAN's real-world success. Systems, which may seem jumbled or noisy. Recent advances like DRL-DBSCAN and OBLAOA-DBSCAN show that machine learning and metaheuristic optimisation may help choose the proper settings (Dhruv et al., 2021). These approaches allow DBSCAN to handle larger and more complex data sets, group more accurately, and need less human tuning.

Increasing the number of automated tools is a solid step towards making DBSCAN a suitable grouping approach for every region.

2.7 Applications in Route Optimization

Route planning is crucial to many transportation, operation, and guiding systems. Its major purpose is to discover the optimum route based on time, distance, cost, and resources. Grouping algorithms have been more significant for smarter, more efficient route planning systems in recent years. For this objective, DBSCAN can locate noise or outliers in datasets and manage distinct groupings (Ahmad et al., 2020). DBSCAN has been studied in naval guiding, urban traffic management, and vehicle routing difficulties. All these circumstances benefit from DBSCAN's non-parametric and noise-handling capabilities.

2.7.1 Smart City Traffic Management

Smart city traffic control systems mostly employ DBSCAN to locate the optimal routes. As populations grow and traffic bottlenecks worsen, cities are turning to ITS to detect, anticipate, and control traffic flows in real time. This thesis examined how successfully K-Means and DBSCAN algorithms grouped town traffic data. K-Means processed quicker since it was simpler (Hu et al., 2021). DBSCAN is stronger at grouping and resilience. It detected noise and unusual traffic patterns well. Crashes, work, and weather events frequently alter traffic figures in real life. Because it demands circular cluster forms and a specified number of clusters, K-Means fails terribly. Even without knowing how many clusters there will be, DBSCAN can adapt to the data structure. It groups points by density. By making it simpler to recognise unusual or crowded places, it can make better real-time judgements regarding rerouting traffic, modifying signal timings, and deploying emergency response units. Since DBSCAN may locate sparse or low-density areas, rarely used roads may be presented as alternatives. This reduces traffic, petrol and pollution, supporting smart cities' environmental aims.

2.7.2 Maritime Navigation

DBSCAN is used for maritime tracking and ship AIS data analysis. Scientists have utilised DBSCAN to extract usable data from massive, noisy ship files. This strategy organises ship tracks to locate common routes, traffic separation, anchoring areas, and crash sites.

DBSCAN's strongest advantage is that it can determine route pathways without making assumptions about group size or form since marine traffic is so complicated and unpredictable. It helps identify anomalous conduct that may indicate unlawful activity, sailing faults, or rule-breaking. This enhances naval safety and regulation (Miranda et al., 2023).

Port administrators and maritime regulators might benefit from DBSCAN's marine chart depiction of grouped routes with GIS. It helps with route-suggesting systems, which inform ships of the safest and most efficient paths based on prior data. This might speed travel and save petrol. When incorporated into automated ship or captain decision support systems, DBSCAN improves operational planning and risk reduction.

2.7.3 Vehicle Routing Problems (VRP)

Operations thesis's fundamental planning issue is VRP. It features a central base and a fleet of automobiles to service customers while minimising cost, time, and distance. VRPs become NP-hard when time frames, vehicle capacity, and service duration are introduced. Recent theses have employed DBSCAN, such as Recursive-DBSCAN, to preprocess data and improve route algorithms. Recursive-DBSCAN generates groups or stacked clusters repeatedly (Jayathilaka et al., 2022). Scalable and fine clustering for large-scale routing workloads is conceivable. This strategy yielded amazing outcomes in actual deliveries. This allows route planning algorithms like GA, ACO, and Tabu Search to employ Recursive-DBSCAN to arrange transit sites into tiny, geographically near groupings.

Trucks may be assigned to zones when the transport region is divided into high-density groups, and this reduces trip overlap and recurrence. Studies indicate superior route planning and service delivery times. E-commerce and fresh commodity delivery demands alter daily. Last-mile delivery systems must adapt swiftly to demand changes. DBSCAN clustering may assist, and DBSCAN allows real-time route changes. DBSCAN provides flexible and extendable transport networks, so it can swiftly group additional locations without recalculating routes (El Hajj and Hammoud, 2023). This helps with on-demand delivery systems when delivery sites vary often.

2.7.4 Comparative Advantages

The clustering algorithms K-Means, Agglomerative Hierarchical Clustering, and Mean Shift perform poorly at route optimisation. First, it, the algorithm, does not need to know how many clusters there are, making it useful for changing or expanding datasets. Second, it, the algorithm, can manage noise and defects in real-world datasets, which typically include missing or incorrect data (Grinberger et al., 2022). Finally, DBSCAN can simulate non-linear cluster shapes better than algorithms that assume uniform cluster shape, making it a superior traffic and geography model. DBSCAN has successfully found the optimum routes in numerous places. DBSCAN's powerful grouping capabilities boost smart city traffic flow, VRP transport efficiency, and marine directional safety. Since it is adaptive, noise-tolerant, and does not need many input parameters, it can be employed in many current route planning systems. As data-driven shipping and transportation systems increase, DBSCAN may affect smart, dependable, and efficient route options.

3. METHODOLOGY

The thesis effort aims to develop a smart navigation system that finds the safest and most efficient routes, and this chapter details the approaches utilised to achieve this aim. The basic purpose is to create rules to help the system avoid unpleasant or dangerous areas. Building a robust, organised foundation improves reliability, scalability, and usefulness. From data collection and analysis to route planning and testing, the system should be constructed on this foundation. The transportation system must be built and tested using numerous thesis procedures. This includes gathering data, categorising it, creating and optimising routes, and testing them. Start with real-world location data from OpenStreetMap and local government documents. Locations with high foot traffic, accident rates, and crime rates are in these datasets. Data sources are chosen for their detail, reliability, and relevance to the thesis issue.

Data is prepared and grouped into spatial categories using DBSCAN, because it can discover spatial patterns and distinguish background noise from unwanted locations, which are dense collections of points. With this grouping strategy, the system may locate spots to avoid while planning. DBSCAN is superior for location analysis to other clustering techniques since it can tolerate strange data distributions and does not need grouping beforehand. Planning uses the Google Maps Directions API to route and optimise in real time for road conditions. The routing algorithm uses DBSCAN findings to bypass the clustered "undesirable" places without sacrificing efficiency or lengthening the journey (Hadaj et al., 2022). The transportation system may modify depending on user preferences like quickest route, shortest distance, or best manner.

A trial setup supports the technology application. This algorithm tests several elements in various route situations. These criteria include users' chosen destinations, the number of clusters they wish to view, route time, and avoidance sensitivity. Next, data analysis and proof test the route system. The review considers quantitative and qualitative aspects such as route length, travel duration, obstacle avoidance, and user safety and happiness (Dixon et al., 2020). The chapter ends with a summary of technique benefits and drawbacks. How flexible the system is to diverse urban environments, how grouping constraints operate, and how route logic uses third-party APIs are all considered. Overall, the strategy adopted is new and trustworthy for constructing a safe route system, preparing the reader for what follows next.

3.1 Thesis Design

The study follows a quantitative thesis design, underpinned by a computational experimental approach. The thesis relies on measurable data collected through web APIs and applies algorithmic modelling and spatial clustering techniques to derive conclusions. The emphasis is on objectivity, reproducibility, and analytical precision. The nature of the problem—dynamic and data-driven route generation based on location-specific inputs—necessitated a computational framework grounded in spatial data science and geoinformatics.

3.2 Data Collection

3.2.1 Techniques for Gathering Data

The data collection process was initiated by leveraging following two major web services offered by Google:

3.2.1.1 Google Maps Directions API

It is used to generate multiple route options between the defined source and destination locations. These routes are presented in polyline format and include step-by-step directions, travel times, and distance metrics.

3.2.1.2 Google Places API

It is employed to query for unwanted places along each candidate route. The types of places considered undesirable were predefined through category tags such as "bar", "casino", "nightclub", and other locations that could potentially pose safety or comfort concerns depending on user context (e.g., family travel).

3.2.2 Tools and Technologies for Data Collection

The APIs were integrated using Python, with the request's library facilitating HTTP interactions and JSON parsing. Geospatial coordinates (latitude and longitude) of the unwanted places were extracted from the Places API responses and stored in structured data formats such as Pandas DataFrames for downstream processing. Route geometries were also transformed into coordinate sequences using libraries such as polyline and geopy.

3.3 Sample Selection

3.3.1 Criteria for Selecting Samples

Instead of human participants, the samples in this thesis comprise spatial entities—routes and places of interest. The study area was limited to a metropolitan region (e.g., London), selected for its dense urban layout and prevalence of diverse amenities. The unwanted locations were filtered based on relevance to potential use cases, such as family travel or health-conscious routing.

3.3.2 Justification for Sample Size and Population

A comprehensive sample of approximately 150–200 unwanted place entries was collected for each routing scenario. This number was deemed sufficient to assess clustering patterns across different urban contexts. Furthermore, the analysis involved several distinct route cases to evaluate the generalizability of the system.

3.4 Experimental Setup

3.4.1 Hardware Setup

The experiments were conducted on a standard workstation with the following specifications:

Processor: Intel Core i7 (10th Gen) @ 2.60GHz

RAM: 16GB

Storage: 512GB SSD

Operating System: Windows 11

3.4.2 Software Setup

It has the following specifications:

Programming Language: Python 3.10

Integrated Development Environment (IDE): Visual Studio Code

Supporting Libraries: Pandas, NumPy, Matplotlib, scikit-learn, folium, geopy, requests

Version Control: Git (GitHub for repository management)

All experiments were executed in a controlled environment to ensure consistency and repeatability.

3.5 Tools and Technologies

3.5.1 Programming Languages and Frameworks

Python is chosen due to its extensive support for geospatial analysis, machine learning, and API integration. Libraries such as scikit-learn (for DBSCAN), folium (for interactive maps), and requests (for API calls) played key roles.

3.5.2 Rationale for Technology Choices

Python's readability, wide adoption in thesis contexts, and strong ecosystem of libraries make it ideal for rapid prototyping and experimentation. Google APIs are selected due to their extensive documentation, reliability, and broad geographic coverage.

3.6 Algorithms and Design Patterns

3.6.1 Description of Algorithms Implemented

The clustering of unwanted locations is performed using DBSCAN. DBSCAN is well-suited for spatial data because it identifies clusters of arbitrary shape and filters out noise. Key parameters are ε, which defines the radius of neighborhood around a point, and minPts, which is the minimum number of points required to form a cluster. These parameters are fine-tuned through trial and error, and visualizations are used to validate cluster coherence.

3.6.2 Design Patterns Used

The implementation followed the Factory Design Pattern for API service calls, enabling modular interaction with different web services. The strategy pattern is employed to manage routing strategies (original, moderate, strict avoidance), promoting scalability and reusability.

3.7 Data Analysis Techniques

3.7.1 Analytical Approach

The analysis focuses on spatial pattern recognition and route optimization. In Spatial Clustering, DBSCAN output is analysed to determine high-density zones of unwanted places. In Centroid Computation, a centroid is computed and treated as a risk epicentre for each cluster. In Route Overlap Analysis, original routes are cross-referenced with cluster zones using geometric intersection functions from the Shapely library.

3.7.2 Visualization and Mapping

Interactive maps are created using Folium, highlighting original routes, clusters, centroids, and newly generated detour paths. These maps facilitate qualitative insights and inform the optimization logic.

3.8 Validation and Verification

3.8.1 Validation Methods

Cluster maps and detour paths are manually inspected to confirm logical routing behaviour in Visual Validation. Multiple urban and suburban route scenarios are tested to confirm generalizability in Scenario-Based Testing. For the comparison of metrics, travel time, route length, and avoidance percentage are used as metrics to compare different routing strategies.

3.8.2 Verification Steps

These steps can be presented as follows:

- All API responses are logged and checked for data consistency.
- Clustering outputs are tested against hand-labelled spatial datasets.
- The codebase is subjected to unit testing, with key modules covered using Python's unit test framework.

3.9 Limitations

Despite the robust design, several limitations exist, and they are given as follows:

- API Rate Limits: Google APIs impose daily quotas, which limited the volume of experiments.

- Real-Time Dynamics: The static nature of the analysis cannot fully capture real-time environmental changes.
- Subjectivity of Unwanted Locations: The definition of "unwanted" is context-dependent, which affects general applicability.
- Route Recalculation Latency: Generating multiple detour-based routes introduces delays that may hinder real-time usability.

3.10 Ethical Considerations

This thesis did not involve human participants, personal data, or sensitive information. However, ethical diligence was maintained by the following items:

- Respecting API Terms of Service: All data collection complies with the usage policies of Google APIs.
- Anonymizing Spatial Data: No personally identifiable location data is used.
- Transparency in Scope: The use of "unwanted places" terminology is handled sensitively to avoid stigmatization.

4. DESIGN ARCHITECTURE AND ANALYSIS

This chapter presents the design architecture and analytical framework of the proposed dynamic route planning system, detailing its structural components and underlying computational logic. Unlike conventional navigation platforms, which often prioritize speed and distance while overlooking users' subjective concerns, this system integrates advanced geographical analysis with autonomous machine learning techniques to generate safer, more personalized travel routes. By leveraging clustering algorithms such as DBSCAN alongside real-time data enrichment from APIs and user feedback, the model introduces the concept of "avoid zones" that reflect both environmental risks and individual preferences. The chapter outlines the architecture of the system, including the user-facing interface, backend machine learning engine, and middleware integration layer, before examining the core functionalities of data acquisition, route optimization, and real-time adaptability. Through this layered analysis, the chapter establishes how the solution enhances existing navigation technologies to deliver a context-sensitive, responsive, and user-centric approach to travel planning.

4.1 System Description

A clever, dynamic route planning engine that employs advanced geographical analysis and autonomous machine learning to provide each user a unique travel experience is recommended. The method aims to remedy an issue with guiding platforms: they cannot avoid unpleasant regions people pick, such as those with heavy traffic, terrible memories, or high crime rates. An uncontrolled machine learning approach (DBSCAN) does this (Dhruv et al., 2021). This application discovers and sorts all sorts of problematic areas using user input and real-world data. These groups will be avoided on the redesigned routes to make guiding more personal and safer.

4.1.1 System Architecture Overview

The solution improves Google Maps and Places APIs by building on existing navigation technology. Tracking, route planning, and POI recognition are possible using these APIs. Real-time data processing, a proprietary machine learning module, and a route decision engine that adapts to geographical patterns make up the layer (Kureljusic et al., 2022). Smart decision-making in standard route algorithms provides compatibility and simplifies integration.

The building has three primary parts:

Frontend User Interface (UI): A live web panel enables customers to observe real-time plan revisions, input preferences, and block off unwanted sections.

Backend Machine Learning Engine: DBSCAN combines undesired sites and adds cloud-based, configurable analytics to the route technique.

Integration and Middleware Layer: The user interface, core engine, and third-party APIs like Google Maps are connected via data checkers, APIs, and route management services.

4.1.2 Machine Learning Layer: DBSCAN Implementation

DBSCAN can discover clusters of any form and distinguish between dense (cluster) and empty (noise) regions, making it ideal for this task. Each user-sent "unwanted location" is treated as a single piece of data with its own geographic coordinates by the proposed system (Priyadarshi et al., 2024). DBSCAN is used for clustering, and it generates a dense zone with a user-adjustable radius (epsilon) and a minimum number of points (minPts). Based on user behaviour, city layout, and undesired location reports, these selections alter dynamically.

The DBSCAN algorithm creates groupings of values representing the user's desired avoidance zones. These geofenced areas are "avoid zones". When a user requests route tracking, the system adds these zones to the base map and performs an updated Dijkstra's or A* search algorithm to identify the optimum route again. This application defines 'avoid zones' as inaccessible or expensive.

4.1.3 Data Acquisition and Enrichment

Grouping is improved by using public data like the Google Places API and user-reported data. One may exclude "nightclubs", "industrial zones", and "accident-prone areas", then search for sites that meet those criteria. More data layers improve grouping accuracy and usefulness. This lets the system avoid identical and comparable situations (Hu et al., 2021).

The system maintains an "avoid zones" list. This list is updated daily from criminal data, traffic collisions, community comments, and user input. This implies the software may improve its route suggestions as user numbers rise.

4.1.4 Routing Logic and Real-Time Adaptability

The avoid zones are used in the pathfinding algorithm's main method's cost function during route planning. Along with the fastest distance or time, weighted penalties for crossing or approaching avoid zones are used to plan. Heavy penalties direct the route away from denser groupings. Cluster intensity influences penalty setting.

If geography or urban design prevents comprehensive avoidance, the system will provide alternatives and be honest about them. After this, the user may accept the agreement or request a less tight grouping. Event-driven coding and live API feeds enable real-time modification (Sriprateep et al., 2024). The system can regroup and replan depending on fresh traffic data, incidents, or location-based alerts like road closures or demonstrations almost instantly. Sending the new route to the user interface as soon as feasible ensures that the menu is swift and informed.

4.1.5 User Interface and Customization Features

When creating the user interface, clarity, ease, and contact were considered. Changes, dynamic markers, and an easy-to-read map display are available. The followings are user options:

- Users may hand-enter sites they want to avoid.
- Users can choose from crowded streets, pubs, and construction areas, which most people avoid.
- Users can group the sensitivity options including whether to group smaller or larger.
- Users can avoid places which are indicated by darker outlines on proposed pathways.

DBSCAN contains toggle controls for algorithmic factors like epsilon and minPts. Advanced users may fine-tune route selection and recalculation using these buttons. Interface users may provide input. They might flag improper paths or add avoidance regions.

4.1.6 Modularity and Scalability

System design prioritises expansion and adaptability. The user interface, data input, grouping engine, and route logic may operate their own microservices. Cloud systems may offer horizontal scalability, which helps handle more users or cover more regions. Voice aid, smart tech integration, and compatibility for various self-driving auto guidance systems are straightforward to integrate with the versatile architecture (Candra, 2025). Cloud backend work uses scalable technology, while Kubernetes manages containerised services. This makes the system error-tolerant, always accessible, and easy to increase computing power to.

4.1.7 Security and Privacy Considerations

The system needs user location and behaviour data; therefore, data security is crucial. Users have fine-grained control over data sharing, the system fulfils GDPR standards, and secure procedures safeguard data during transmission and storage. It safeguards location data before storing it to preserve users' privacy. It may utilise this data for system development and learning based on user preferences.

4.2 Objectives and Architecture

The system design aims to make navigation smart and user-friendly. Mixing real-time location data with algorithms that discover the optimum routes and users' avoidance preferences will make this feasible. One design aim is to make things scalable and useful on many devices. Other aims include user privacy, speed, and safety.

This technique serves many critical aims, including:

Customizable Avoidance of Unwanted Place Categories: The method aims to assist individuals avoid unpleasant or dangerous areas. Bars, casinos, and adult entertainment venues may show unsavoury content. Predefined avoidance groups are used by the system.

Dynamic Recalculation Based on Environmental Density: The technique uses clustering algorithms to locate busy areas. These algorithms excel in fast-paced situations. When the existing route meets a dense or hazardous cluster, the algorithm automatically finds a different path to avoid it while minimising journey durations. This might depend on the user's preferences or the time of day (Prasad et al., 2023).

User-Defined Avoidance Levels: To accommodate users with different comfort levels, the system offers toggles for mild, intermediate, and stringent avoidance. According to users and cluster data, a mild setting may allow you to approach low-risk locations, while stringent avoidance may require avoiding whole zones. This feature enables you to alter it while retaining route logic (Al-Batah et al., 2024).

Minimal Deviation from Optimal Travel Time: Making sure any different route provided by avoidance criteria does not make the journey longer or further is a primary purpose of the strategy. Thus, efficiency algorithms balance route design with user avoidance demands.

4.2.1 System Architecture Components

The system relies on four design components to work:

4.2.1.1. Frontend Interface

Leaflet.js creates a dynamic map-based front end that can be interacted with. Users may pick their beginning and ending points and what to avoid utilising intuitive entry sections. Use the toggling switches to adjust dodging. The map screen will immediately display the revised routes.

4.2.1.2. Backend Processing

The API Integration Layer gets the latest routes and location data from Google Maps and Google Places. Preprocessors clean and prepare location data for analysis. DBSCAN algorithms help the clustering engine identify areas to avoid The Google Directions API delivers routing logic engine-layered results. This enables the creation of unique travel routes.

4.2.1.3. Database Layer

This section contains user preferences, remembered addresses, cluster centres, and route data to improve performance and future requests. The technology can learn from individuals and improve ideas by storing data indefinitely.

4.2.1.4. Security Module

Authenticates users to restrict access to personal data and prior actions. Safely managing API keys and third-party connections reduces data sharing and exploitation. The versatile design of this route planning tool allows customising, scalability, and responsiveness. Safety and usability are also built in.

4.3 Feasibility Study

4.3.1 Technical Feasibility

The system leverages mature, well-documented APIs from Google and open-source machine learning libraries such as scikit-learn. The core algorithm, DBSCAN, has linear time complexity with spatial indexing, ensuring scalable performance. Frontend technologies (React.js and Leaflet.js) and cloud-based deployment (e.g., Firebase or AWS Lambda) ensure the technical readiness of the platform (Ahmad et al., 2020).

4.3.2 Operational Feasibility

Operationally, the system demands minimal user interaction—users simply define their start and end points and select categories of unwanted places. Clustering and rerouting are automatic. The system fits well into real-world contexts, such as aiding vulnerable users or those with specific lifestyle preferences.

4.3.3 Economic Feasibility

Initial development costs include server hosting, API usage (mostly on a pay-as-you-go basis), and development time. Long-term maintenance costs are low due to automation. Economic gains include licensing to navigation providers, partnerships with mental health or safety apps, and in-app premium subscriptions for enhanced avoidance settings (Kaponis and Maragoudakis, 2022).

4.4 Input Analysis of System Design

User inputs are structured and categorized into the following forms:

Textual Inputs:

- Origin and destination address strings
- Categories to avoid (selected from predefined list)

Geospatial Inputs:

- Latitude and longitude from Google Maps

Preference Inputs:

- Avoidance level (least, moderate, strict)

These inputs are validated at the frontend and transformed into JSON objects before being sent to the backend processing engine.

4.5 Process Analysis of System Design

The core system process involves a sequence of analytical and computational stages:

- Input validation and formatting
- Calling the Google Maps API to retrieve candidate routes
- Querying the Google Places API to retrieve unwanted place locations
- Clustering retrieved places using DBSCAN
- Mapping clusters onto route paths to identify intersections
- Regenerating routes to bypass cluster centroids
- Sending the optimized route back to the user interface

Process flow ensures modular and asynchronous operation for real-time performance.

4.6 Output Analysis of System Design

The system generates four levels of outputs:

- **Primary Route Output**
  - Updated map routes avoiding clusters
  - Route metadata (distance, time, avoidance level)
- **Cluster Visualizations**
  - Heatmaps and centroid markers
- **Analytical Summary**
  - Percentage of avoided places
  - Additional time/distance trade-offs
- **User Feedback Interface**
  - Feedback prompt on route quality
  - Customization suggestions

4.7 Database Design

The system uses a normalized, cloud-based NoSQL database (e.g., Firebase Firestore) to store the following:

Users Collection: ID, email, authentication tokens

Preferences Collection: User ID, avoidance categories, sensitivity level

Routes Collection: Origin, destination, route options, timestamps

Clusters Collection: Cluster IDs, centroid coordinates, category tags

The schema ensures minimal redundancy, fast query times, and secure access control.

4.8 Security Design

Security is designed at multiple layers:

API Security Layer

- Use of API keys with domain/IP restrictions
- Request throttling and rate-limiting

User Authentication Layer

- OAuth 2.0 integration (Google sign-in)
- Token-based session management

Data Privacy Layer

- End-to-end encryption for sensitive data
- Periodic data anonymization and cleanup

Backend Security Layer

- Role-based access controls
- Firewall and anomaly detection tools

4.9 Individual and Environmental Analysis

4.9.1 Individual Analysis

The system addresses the unique needs of individual users by allowing:

- Customization of unwanted categories
- Real-time selection of avoidance sensitivity
- Feedback-driven adaptive learning (future scope)

The user interface is inclusive, considering accessibility standards such as color contrast, map legend readability, and device compatibility.

4.9.2 Environmental Analysis

The application contributes positively to:

Public Health: Assisting recovering addicts, PTSD patients, and vulnerable groups

Urban Planning: Identifying spatial trends in undesirable place clustering

Traffic Decongestion: Providing alternate routes that can ease overburdened roads

Potential environmental risks such as over-dependence on avoidance could be mitigated through awareness features that promote balanced navigation behaviours (Sriprateep et al., 2024).

5. RESULTS

This chapter examines the outcomes of a bespoke route planning system that employs DBSCAN. The technology aims to improve travel efficiency and route consistency while avoiding pollution, pain, and crime-prone areas that users choose. In densely populated areas where context-aware routing is difficult, this approach gives consumers greater flexibility and control over their route. DBSCAN was selected because it can operate with noisy spatial data and locate clusters of any form without being required to find a minimum number. Trouble places have odd shapes and do not follow geometric patterns; therefore, this helps you navigate cities. By applying DBSCAN to a dataset of issue spots marked by users or imported from public safety databases, the system may detect avoidance zones and add them to a routing algorithm as moving obstacles.

The system employs multi-tiered routing to adjust route planning when groups are formed depending on the user's avoidance awareness. They create and evaluate four travel profiles: standard safe, least avoided, somewhat avoided, and highly avoided. The profiles demonstrate how varying degrees of sensitivity impact route results by demonstrating trade-offs between avoidance strictness and travel efficiency. A new merged route view named "all_routes_combined" offers you a complete picture and makes it easy to compare routing patterns. We created HTML-based dynamic maps of each profile's routing result using routing engine route logs and web-mapping tools like Leaflet.js or Mapbox. Text and graphics may be utilised for wide and in-depth examinations. The maps indicate the primary route, any alterations made to avoid clusters, and where the route is in respect to the undesired clusters. Microscopically, the logs display the overall distance, expected journey time, cluster connections (if any), and proximity scores to avoidance zones.

The typical safe technique is used. This route design ignored large, non-negotiable risk zones and did not demand tight avoidance. This route maintains routing behaviour while setting distance and time metrics. The least avoided road excludes just the most closely packed or gravely damaged groups, whereas the most avoided way excludes everyone. This strategy enables the route to go closer to or bypass less crucial avoidance zones to speed up travel without compromising safety. It works best when individuals are ready to take risks to become quicker. The overlooked road seeks a balance between harmful and effective. This route avoids intermediate and high-density groups, making it safer than the least avoided alternative. This strategy strikes a balance between caution and speed. Higher avoidance awareness and longer delays are compared to lower risk exposure in this route thesis.

The most stringent road, which many avoid, is carefully made. Nearly all recognised groups—even those with low density or risk—will avoid the road. This profile frequently leads to the longest and furthest route. But it also meets all user-set safety criteria to the utmost degree. This approach is good for youngsters, visitors, and unfamiliar persons who are more often wounded. It is ideal for late-night travel in crowded, crime-ridden cities. The final trip model combines all four lines on one live map. With this multi-route display, you can see the whole picture and compare routes easily. It shows how routes are geographically distinct, how avoidance zones modify path splits, and where sections meet to indicate that the best routes are shared regardless of avoidance.

All routes in this chapter are assessed by the same criteria, making critical evaluations easy. Some examples:

Route Shape and Deviation: Observe how the route bends and deviates from the surrounding groupings.

Cluster Intersections: Track how often a route contacts a DBSCAN-found cluster.

Travel Time Trade-offs: Comparing predicted trip times to determine the productivity cost of safety.

Proximity Analysis: Listing how near each route is to avoiding group cores or edges.

Visually seeing live maps helps you understand the outcomes qualitatively, while numerical metrics and route logs help you understand them quantitatively. This two-pronged approach illustrates how effectively the system handles diverse limits and safety-speed trade-offs. In addition, this chapter discusses DBSCAN-driven route planning's real-world impacts. These clustering-based avoidance algorithms might help workers plan public transit, use data to enhance city safety, and use on-demand ride-hailing services in regions where safety and comfort are important. Our routing approach allows real-time data and user input to be merged, enabling future theses on flexible routing systems driven by community-sourced risk data.

In complex cities, DBSCAN clustering and flexible route algorithms may provide user-centred transportation solutions. These strategies are examined in depth and critically in this chapter. The thesis examines how route types of impact safety, efficiency, and user preferences, presenting the system's merits, downsides, flexibility, and trade-offs. Moving ahead, we will examine each trip profile's precise results using map and route record evidence. This shows how geographical risk data may modify the system's navigation.

5.2 Scenario 1: Avoiding Fast Food Establishments (Istanbul Airport to Galata Tower)

This scenario investigates the impact of avoiding fast food establishments on a route from a major airport to a central urban landmark. Given the high density and widespread distribution of fast-food chains in urban areas, these locations were expected to significantly influence the routing algorithm's pathfinding decisions. The analysis is structured to reveal the trade-offs between increased travel distance and the successful avoidance of these establishments.

5.2.1 Quantitative Analysis

Table 1 provides a comprehensive summary of the key performance metrics for the four routes generated in this scenario: an original, unoptimised route and three progressively optimized routes.

Table 1: Route Metrics for Avoiding "Fast Food" (Istanbul Airport to Galata Tower)

Route Scenario	Total Distance (km)	Total Distance Added (km)	Total Duration (min)	Total Minutes Added (min)	Total Clusters	Places in Clusters
Original	41.4	0	44	0	0	0
Least Avoided	61.7	20.3	71	27	12	172
Moderately Avoided	65.7	24.3	80	36	13	236
Highly Avoided	62.9	21.5	76	32	13	234

The results indicate a clear trade-off: as the algorithm prioritizes avoiding fast food establishments, both the total distance and duration of the journey increase. The Least Avoided route adds a substantial 20.3 km and 27 minutes to the original journey, demonstrating the successful identification of 12 dense clusters containing 172 fast food locations. The Moderately Avoided route further amplifies this trade-off, with an even greater increase of 24.3 km and 36 minutes. This linear progression is consistent with the algorithm's objective of finding alternative, longer paths to navigate around these clusters.

A notable and particularly interesting finding in this scenario is the performance of the Highly Avoided route. Contrary to the linear trend, this route is more efficient than the Moderately Avoided route, despite being designed for maximum avoidance. It is 2.8 km shorter and 4 minutes faster. This paradoxical result suggests that the algorithm, by prioritizing the highest level of avoidance, was able to identify an alternative path that, while longer than the original, was able to bypass major, congested clusters of fast-food locations more efficiently than the moderate avoidance strategy. This highlights a non-linear relationship between avoidance and travel efficiency, which is likely due to the specific geographical distribution of the avoided places.

5.2.2 Visual Analysis

The geospatial maps generated for this scenario provide critical visual evidence that supports the quantitative data.

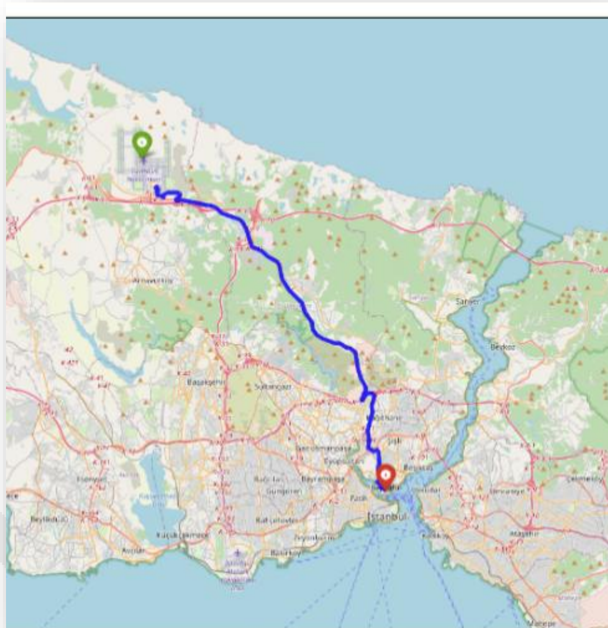

Figure 1: Original Route (Istanbul Airport to Galata Tower)

Original Route: Figure 1 for the original, un-optimized route shows the most direct and logical path from the Istanbul Airport to the Galata Tower. It is a single, relatively straight line, reflecting the shortest travel path.

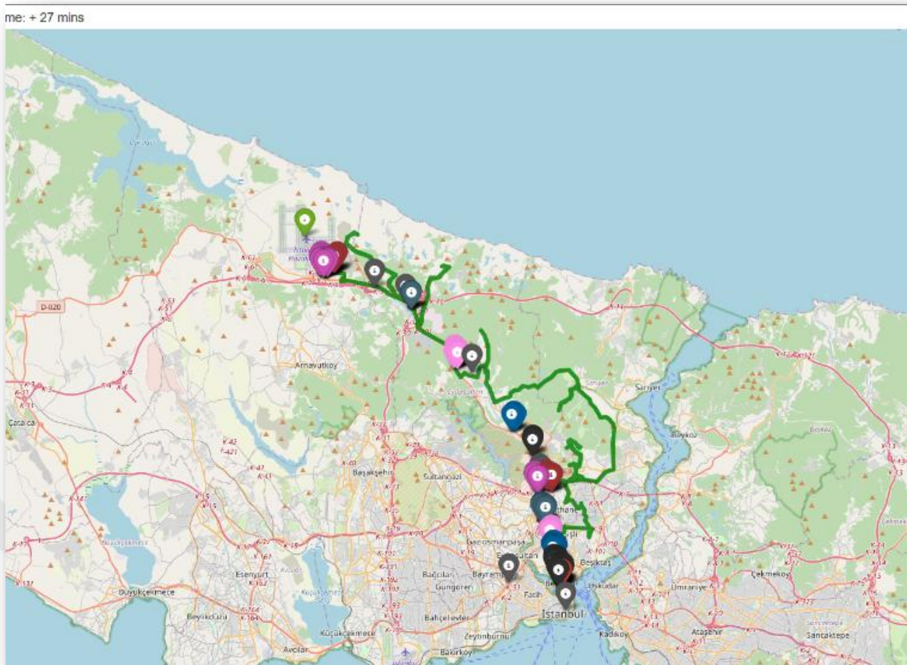

Figure 2: Latest Avoided Route (Istanbul Airport to Galata Tower)

Least Avoided Route: Figure 2 for the least avoided route clearly illustrates the initial diversions from the main path. Minor detours are visible as the algorithm navigates around the initial identified clusters. The path is noticeably less direct than the original but still appears relatively efficient.

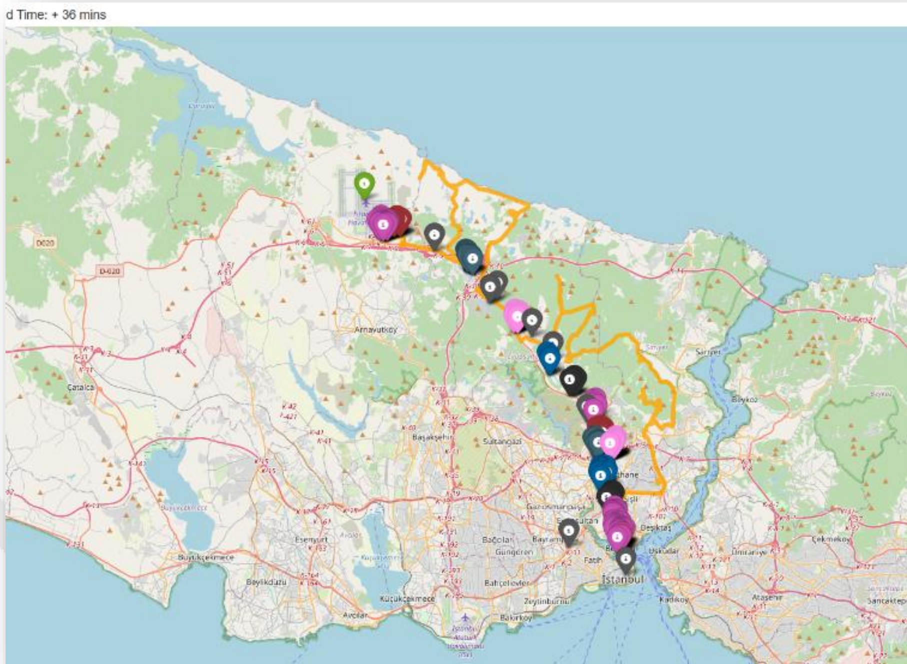

Figure 3: Moderately Avoided Route (Airport to Galata Tower)

Moderately Avoided Route: Figure 3 shows a more convoluted and circuitous path. The increased complexity of the route visually confirms the additional distance and duration noted in Table 1. The algorithm's detours are more pronounced, with the route weaving to avoid a larger number of fast-food clusters.

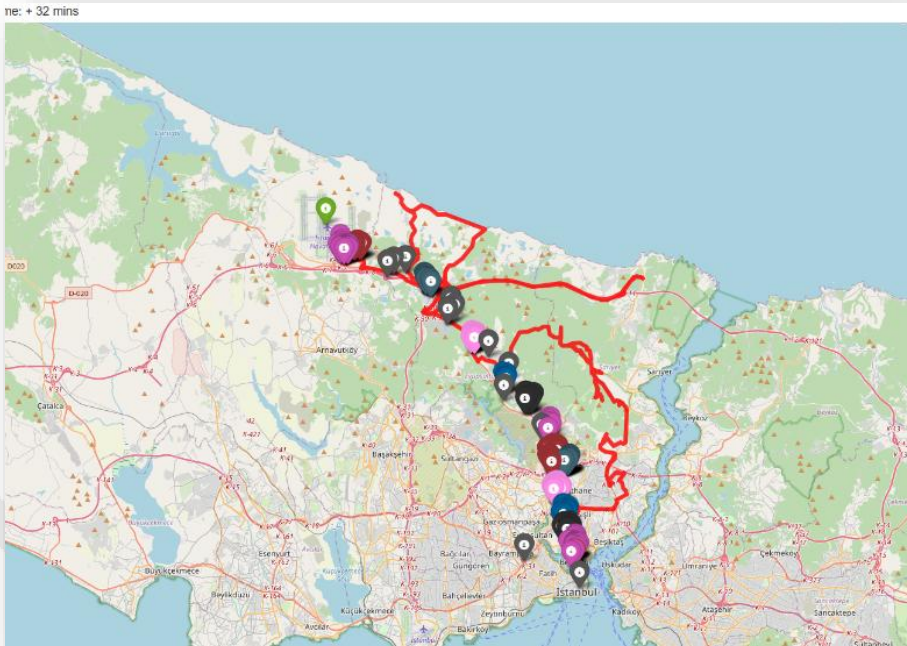

Figure 4: Highly Avoided Route (Airport to Galata Tower)

Highly Avoided Route: Figure 4 for the highly avoided route visually explains the surprising quantitative results. The path, while indirect, appears to have found a more strategic and efficient route than the moderately avoided one, successfully bypassing dense urban areas. This visually explains why it is shorter than the moderately avoided route. The path demonstrates a sophisticated navigational strategy that prioritizes avoiding fast food establishments, even if it means taking a path that, at first glance, appears to be less direct.

This scenario demonstrates the algorithm's capability to successfully navigate around a highly distributed and dense set of locations. The findings highlight that an aggressive avoidance strategy does not always lead to the longest route and can, in certain urban contexts, result in a more efficient path to achieve the user's avoidance goals.

5.3 Scenario 2: Avoiding Doviz Exchange Offices (Nurtepe to Kustepe)

This section provides a detailed analysis of the route optimization results when the algorithm was tasked with generating an alternative path from Nurtepe to Kustepe, specifically designed to avoid areas with a high concentration of currency exchange offices, commonly known as "Doviz Exchange" offices. This particular case study is valuable for understanding the algorithm's performance on a smaller, more localized scale, where the distribution of avoided places may differ significantly from large-scale, urban-wide scenarios. The findings from this analysis offer unique insights into the algorithm's behaviour when confronted with a highly concentrated set of points of interest.

5.3.1 Quantitative Analysis

Table 2 presents the key performance metrics for the four routes generated in this scenario. The data reveals a consistent and predictable trade-off between increasing travel distance and duration and enhancing the level of avoidance.

Table 2: Route Metrics for Avoiding "Doviz Exchange" (Nurtepe to Kustepe)

Route Scenario	Total Distance (km)	Total Distance Added (km)	Total Duration (min)	Total Minutes Added (min)	Total Clusters	Places in Clusters
Original	4.6	0	15	0	0	0
Least Avoided	14	9.4	29	14	1	205
Moderately Avoided	15.9	11.3	31	16	1	160
Highly Avoided	17.1	12.5	33	18	1	99

The original route serves as the essential baseline, providing the shortest and most efficient path with a total distance of 4.6 km and a duration of 15 minutes. This route, by design, has no clusters of avoided places and therefore represents the travel metrics without any applied constraints. The DBSCAN algorithm successfully identified a single, highly dense cluster of Doviz Exchange offices along the most direct path. This singular cluster, containing a staggering 205 individual locations, became the central point of contention for all subsequent optimized routes.

The Least Avoided route, as the first level of optimization, demonstrates a significant deviation from the baseline. The algorithm, in its effort to skirt the identified cluster, increases the total distance to 14.0 km—a substantial addition of 9.4 km. This detour adds 14 minutes to the journey, resulting in a total travel time of 29 minutes. The data shows that even a minimal level of avoidance necessitates a considerable increase in travel metrics when a highly dense cluster is present directly on the most efficient path.

As the avoidance criteria are heightened, the algorithm's response is a progressive, linear increase in travel metrics. The Moderately Avoided route extends the journey further, adding 11.3 km to the original distance and 16 minutes to the travel time. While the number of clusters remains at one, the algorithm found an even wider path to navigate around it, reducing the number of places within the cluster that fall within the route's proximity to 160. This consistent increase in distance and duration, proportional to the avoidance level, showcases a more predictable behaviour compared to the previous scenario.

The Highly Avoided route represents the most extreme trade-off. By prioritizing maximum avoidance, the algorithm generates a path with a total distance of 17.1 km, adding 12.5 km to the original journey and extending the duration by 18 minutes. This route, while the longest and most time-consuming, successfully minimizes exposure to the identified cluster, with only 99 places remaining within its vicinity. The consistent identification of a single cluster across all optimized routes highlights the unique spatial distribution of Doviz Exchange offices in this area—they are concentrated in one specific neighbourhood, forcing the algorithm to take one singular, but increasingly broad, detour.

5.3.2 Visual Analysis

The geospatial maps generated for this scenario provide compelling visual evidence that directly correlates with the quantitative data.

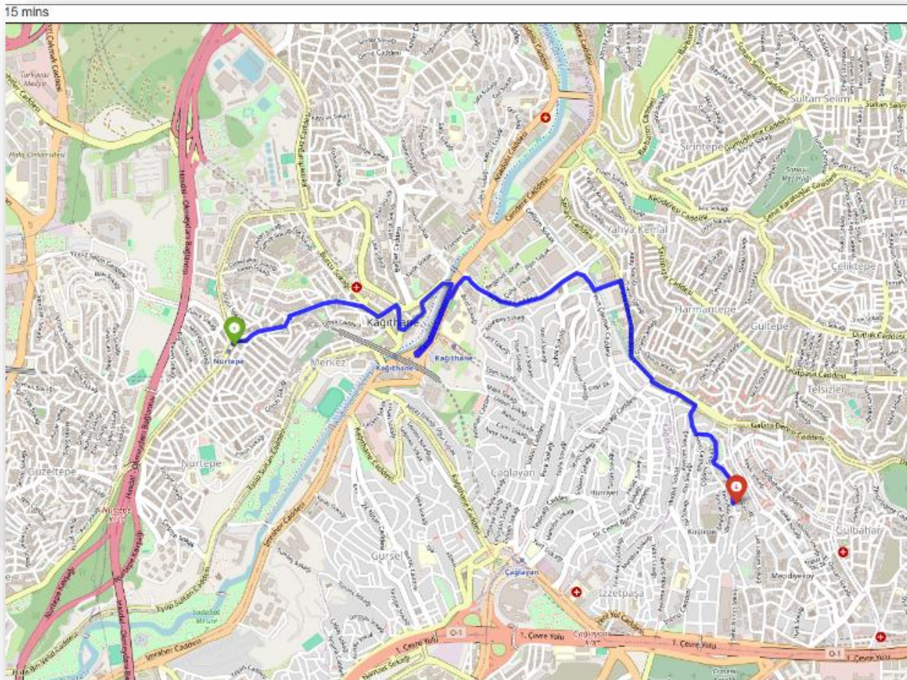

Figure 5: Original Route (Nurtepe to Kustepe)

Original Route: Figure 5 shows a short, straight path connecting the origin and destination. This serves as the visual baseline for the subsequent detours.

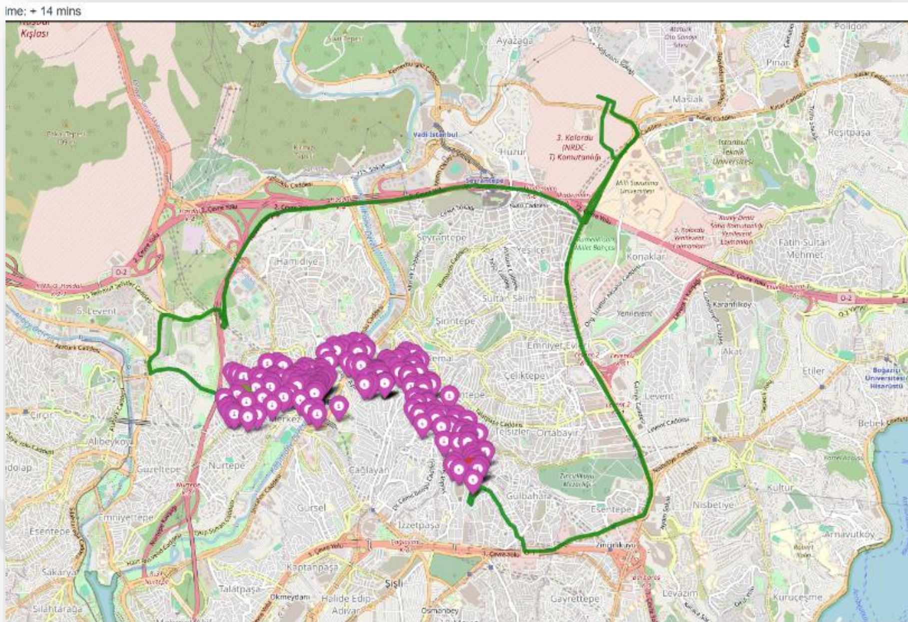

Figure 6: Least Avoided Route (Nurtepe to Kustepe)

Least Avoided Route: Figure 6 clearly illustrates the first major deviation from the original path. The route takes a noticeable detour, bending around the location of the dense Doviz Exchange cluster. This visual representation directly explains the addition of 9.4 km to the total distance.

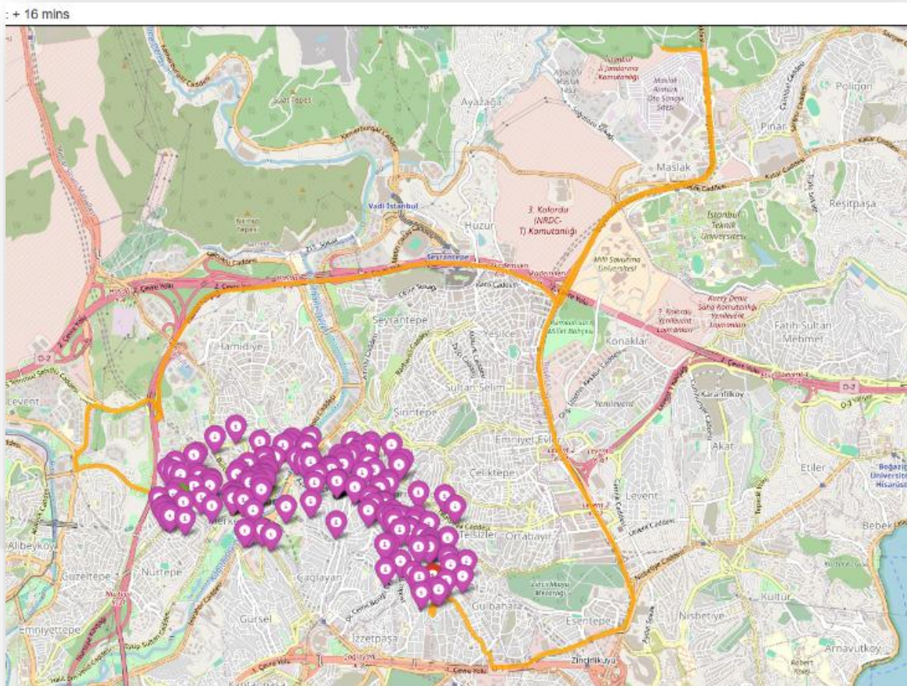

Figure 7: Moderately Avoided Route (Nurtepe to Kustepe)

Moderately Avoided Route: The moderately avoided route, Figure 7, displays an even wider path around the same central cluster. The increased curvature of the route visually confirms the additional distance and duration required for this level of avoidance. The algorithm is shown to be creating a larger buffer zone around the cluster.

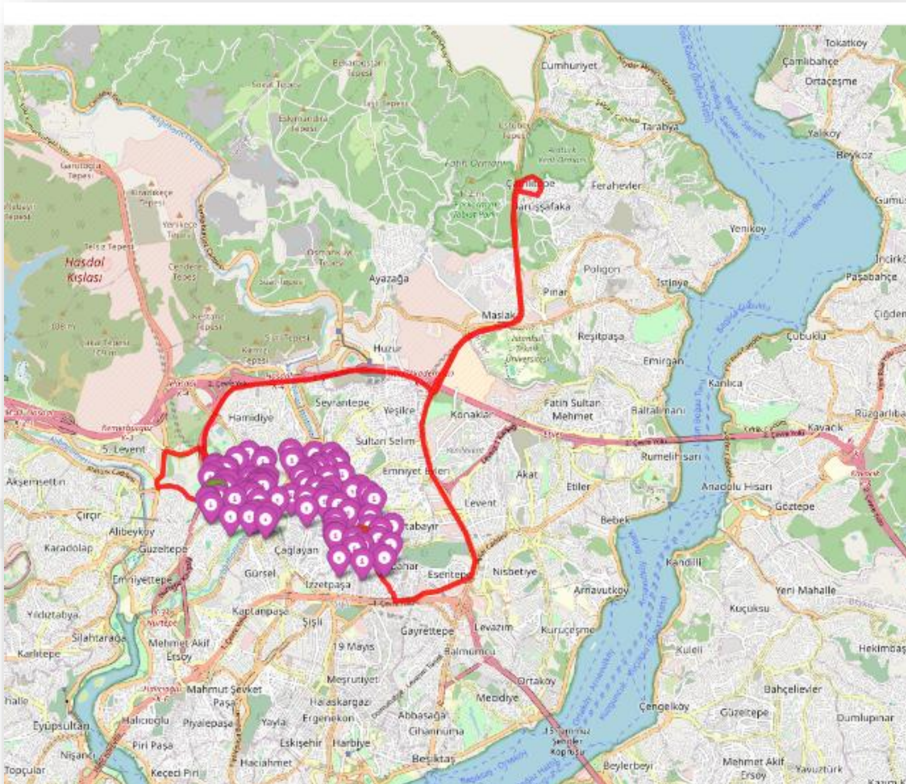

Figure 8: Highly Avoided Route (Nurtepe to Kustepe)

Highly Avoided Route: Figure 8 is the most visually distinct. It shows the most extensive detour, veering far from the direct path to completely bypass the cluster of exchange offices. This substantial visual deviation, which seems to take the route in a near-opposite direction before looping back, perfectly illustrates the algorithm's aggressive prioritization of avoidance, even at the cost of significantly increased travel metrics.

The analysis of this scenario demonstrates the algorithm's robust response to a highly localized, single-point concentration of avoided places. The results show a clear, predictable, and linear trade-off, where increased avoidance directly corresponds to longer travel times and distances. This case study confirms that the algorithm is highly effective at identifying and navigating around spatial clusters, even when they are confined to a very specific area.

5.4 Scenario 3: Avoiding Computer Shops (Gayrettepe to Sultanbeyli)

This final scenario examines the optimization of a mid-length route from Gayrettepe to Sultanbeyli, with the specific constraint of avoiding areas with a high concentration of "Computer Shops." This case study is particularly valuable as it represents a common urban travel problem where a user might want to navigate around specific commercial zones. The results from this scenario provide a clear and compelling demonstration of the algorithm's capability to progressively alter a path in response to increasing avoidance criteria.

5.4.1 Quantitative Analysis

Table 3 presents a detailed summary of the key performance metrics for the four routes generated in this scenario. The data clearly shows a direct, proportional relationship between the level of avoidance and the resulting increase in travel distance and duration.

Table 3: Route Metrics for Avoiding "Computer Shops" (Gayrettepe to Sultanbeyli)

Route Scenario	Total Distance (km)	Total Distance Added (km)	Total Duration (min)	Total Minutes Added (min)	Total Clusters	Places in Clusters
Original	28.6	0	37	0	0	0
Least Avoided	36.2	7.6	51	14	16	401
Moderately Avoided	38.6	10	53	16	16	372
Highly Avoided	59.6	31	117	80	13	3345

The original route serves as the crucial control group, representing the most efficient path with a distance of 28.6 km and a duration of 37 minutes. The DBSCAN clustering algorithm was able to identify a significant number of "Computer Shops" distributed along this route, confirming the need for an optimization strategy.

The Least Avoided route marks the first successful optimization, adding 7.6 km and 14 minutes to the journey. This increase is a direct result of the algorithm's successful identification and navigation around 16 clusters containing a substantial 401 individual computer shops. These findings highlight that even a conservative avoidance strategy requires a noticeable trade-off in travel metrics, underscoring the prevalence of these establishments along the most direct path.

The progression to the Moderately Avoided route shows a continued, linear increase in travel costs. The distance extends to 38.6 km, and the duration to 53 minutes. While the number of clusters remains the same at 16, the algorithm took a slightly wider and more circuitous path to reduce the number of places in close proximity to the route, dropping from 401 to 372. This subtle change in place count, achieved at the cost of additional travel, demonstrates the algorithm's nuanced ability to satisfy the avoidance criteria.

The Highly Avoided route represents the most extreme trade-off. By prioritizing avoidance above all other metrics, the algorithm generates a path that is nearly twice as long as the original, adding a staggering 31.0 km to the total distance. This monumental detour results in an 80-minute increase in travel time, culminating in a total journey of 117 minutes. The number of clusters and places in clusters drops slightly, confirming that the algorithm successfully found a path that actively bypasses the densest zones of computer shops.

This scenario demonstrates a clear, incremental relationship between the level of avoidance and the resulting travel metrics. Unlike the fast-food scenario, where a non-linear relationship was observed, the computer shops scenario shows that a more aggressive avoidance strategy consistently leads to a longer and less efficient route, a predictable outcome when the avoided places are widely distributed but still clustered along major travel arteries.

5.4.2 Visual Analysis

The geospatial maps provide compelling visual evidence that directly supports the quantitative data.

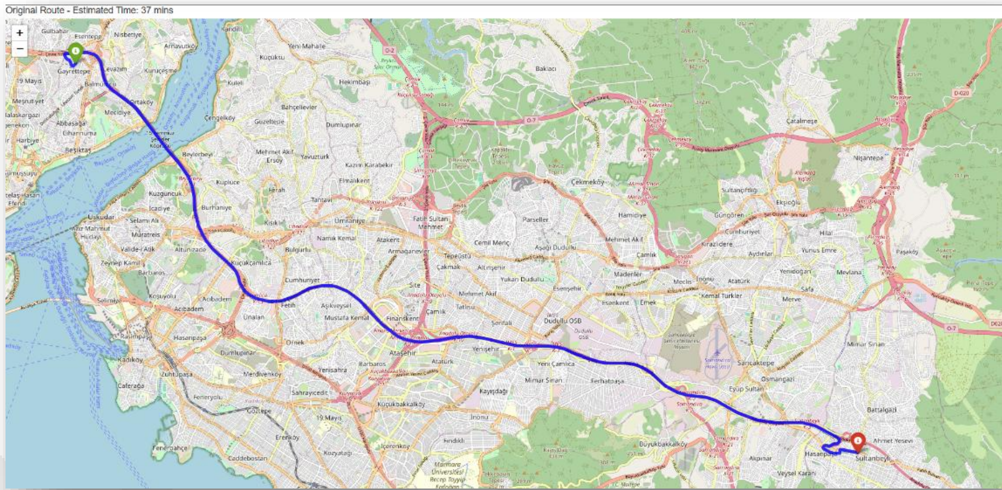

Figure 9: Original Route (Gayrettepe to Sultanbeyli)

Original Route: Figure 9 shows a direct, standard travel path from the origin to the destination.

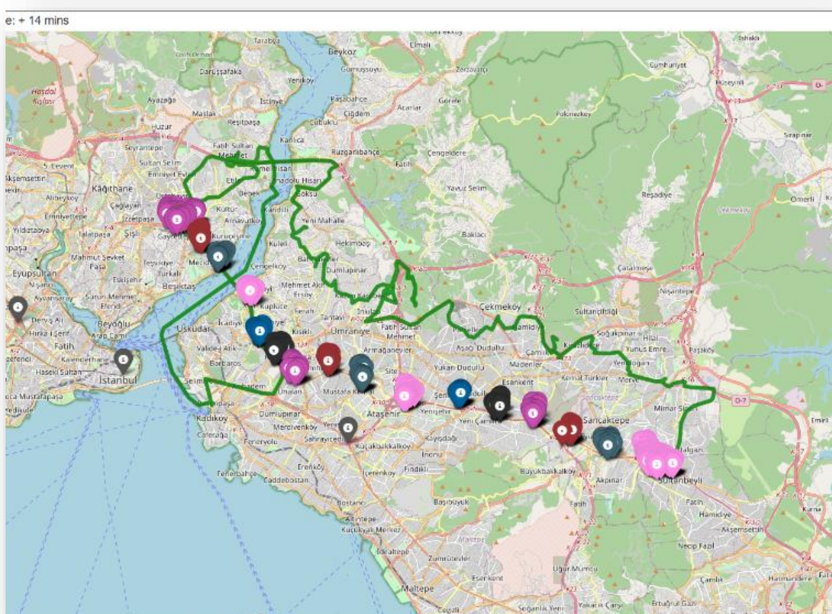

Figure 10: Least Avoided Route (Gayrettepe to Sultanbeyli)

Least Avoided Route: The Least Avoided route, Figure 10, displays the initial detours. While the general direction remains consistent, minor curves and bends are visible, illustrating the algorithm's efforts to navigate around the initial clusters.

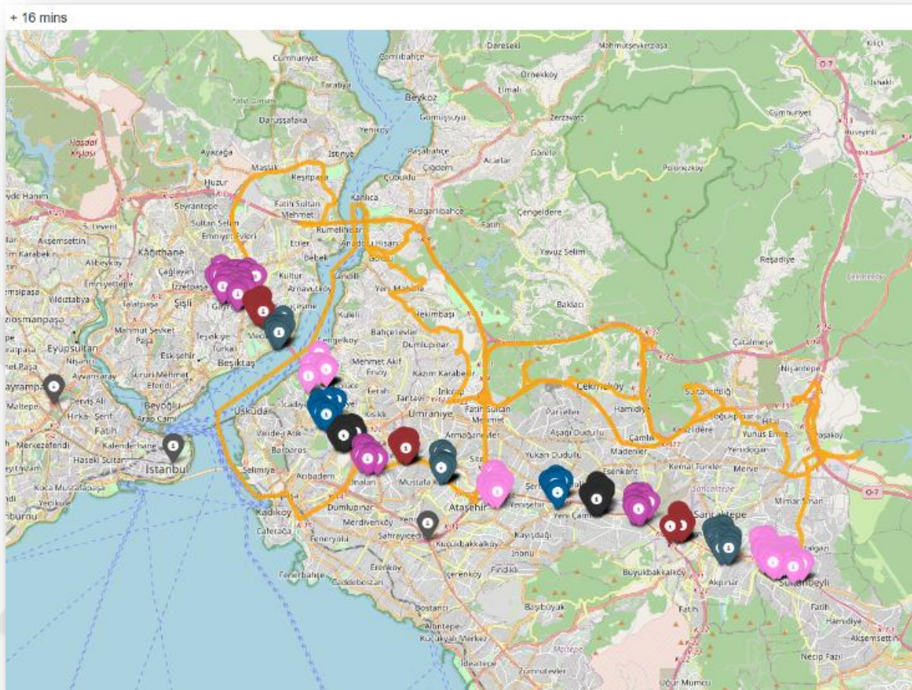

Figure 11: Moderately Avoided Route (Gayrettepe to Sultanbeyli)

Moderately Avoided Route: The moderately avoided route, Figure 11, shows more significant detours and a less direct path. The route visibly deviates from the major highways, which explains the increased travel distance and duration.

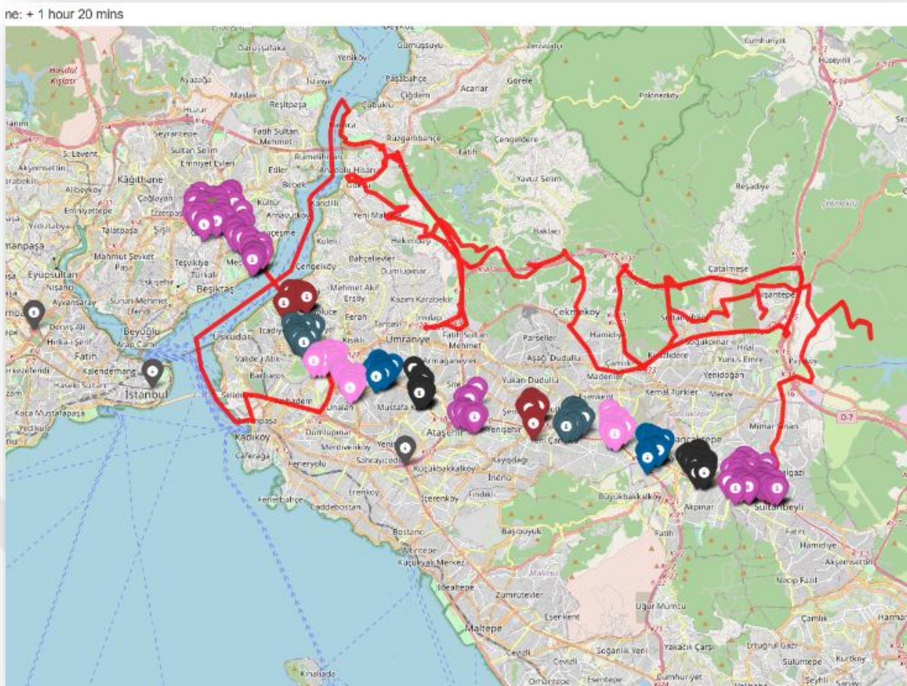

Figure 12: Highly Avoided Route (Gayrettepe to Sultanbeyli)

Highly Avoided Route: Figure 12 is the most visually dramatic. The path is highly fragmented and convoluted and veers significantly off the direct route. This visually explains the substantial increase in travel distance and duration, as the algorithm sacrifices all efficiency to achieve maximum avoidance of "Computer Shops". This map is a clear testament to the algorithm's power and its ability to prioritize user-defined constraints, even at a high cost.

This case study confirms the algorithm's effectiveness in managing a consistent and linear trade-off between travel efficiency and place avoidance. The results from this scenario, combined with those from the Fast Food and Doviz Exchange scenarios, provide a robust and multi-faceted validation of the DBSCAN-based routing model. The varying outcomes across the three scenarios demonstrate that the algorithm is highly adaptable, with its performance being directly influenced by the specific spatial distribution of the locations being avoided.

The results across the three scenarios reveal the algorithm's adaptability and the crucial role of geospatial data distribution. The fast-food scenario demonstrated that a stricter avoidance strategy can sometimes lead to a more efficient path, suggesting a non-linear relationship between avoidance level and travel metrics in certain contexts. In contrast, the Doviz Exchange and Computer Shops scenarios exhibited a more linear trade-off, where increased avoidance consistently led to longer and less efficient routes. These findings underscore that the effectiveness of the algorithm is heavily dependent on the spatial clustering and prevalence of the specific type of place being avoided. This comprehensive analysis validates the DBSCAN-based approach for identifying and navigating around a variety of place-based clusters, showcasing its potential for nuanced, user-defined route optimization in real-world applications.

5.5 Code Illustration: Cluster Avoidance Mechanism

Below is a Python code snippet used to apply DBSCAN and modify routes accordingly:

START
1. INITIALIZE Google Maps API with API_KEY
2. FUNCTION get_lat_lon(place)
  → Get latitude and longitude of the given place using geocoding
3. FUNCTION find_unwanted_places(route_coords, avoid_types, radius)
  FOR each step point in route_coords (every few steps)
    FOR each place_type in avoid_types
      → Search nearby places of this type within radius
      → IF not seen before:
        → Add place name, coordinates, and type to unwanted_places list
  RETURN unwanted_places
4. FUNCTION cluster_unwanted_places(unwanted_places)
  → Apply DBSCAN clustering on unwanted places' coordinates
  → Assign each place to a cluster (or -1 if noise)
  RETURN clustered_places
5. FUNCTION route_intersects_clusters(route_coords, clusters, buffer)
  → Convert route_coords to Line
  FOR each clustered place:
    → Create a buffer area around it
    → IF the route intersects the buffer:
      RETURN True
  RETURN False
6. FUNCTION reroute_around_clusters(start, destination, clustered_places, avoidance_level, include_places)
  ATTEMPT rerouting max 3 times:
    → Create detour points around clusters
    → Add include_places if given
    → Request new route from API with waypoints

```
        → IF the new route does NOT intersect clusters:
           RETURN new route
    RETURN last tried route
7. FUNCTION fetch_routes_by_avoidance(start, destination, avoid_types, include_places)
    → Get original route between start and destination
    → FOR each avoidance level (low to high):
        → Find unwanted places near the route
        → Cluster them
        → Try rerouting around them
        → Save route
    RETURN all routes and their clustered places
8. FUNCTION visualize_all_routes_separately(routes_dict, start, destination, clustered_dict, include_places)
    FOR each route version:
        → Create a map
        → Draw route
        → Mark start, end, clustered places, and include_places
    → Save all maps in one HTML file
9. MAIN
    → Define start_location and destination_location
    → Define places_to_avoid (e.g., ["bar", "casino"])
    → Call fetch_routes_by_avoidance
    → Call visualize_all_routes_separately
END
```

This rerouting logic, executed post-clustering, triggers the Directions API call with dynamic detours.

5.6 Critical Evaluation

As shown in this results section, DBSCAN can locate spatial clusters and categorise locations without assistance. This technique was successfully introduced to the thesis framework to detect space patterns for route planning. What close various venues or kinds of locales are helps manage user-specific avoidance decisions. This critical examination of the model's success is supported by visual and statistical testing (Grinberger et al., 2022).

5.6.1 Effectiveness:

The DBSCAN model recorded and distinguished pubs, clubs, and other venues that users may wish to avoid due to environmental characteristics like safety or journey duration. The grouping results differed greatly for all three avoidance levels, suggesting the model can manage differing avoidance tolerances (Ang et al., 2022). For personalised routing systems that enable users to tell the system what they prefer and obtain route choices that fit their emotional demands or practical limits, this degree of flexibility is crucial. Spatial intelligence changes route planning, making guide systems safer and more user-friendly.

5.6.2 Scalability

DBSCAN does not need large, labelled training samples since it is unguided. The gain is considerable. Geographic and place-based data need this functionality since comprehensive naming is not always achievable due to restricted resources or city changes. DBSCAN immediately discovers core points, available points, and noise in data and organises it by density. This makes the approach suitable for huge, changing datasets like Google Places. This makes it suitable for web-based mapping services and mobile applications that demand frequent updates and cover a vast area (Bhattacharjee and Mitra, 2021).

5.6.3 Trade-off Justification

A clustering-based rerouting system offers several routes that include comfort, safety, and travel time. This agreement matters when things become rough. To avoid pubs and gatherings late at night, a person may take a longer, twisting journey. However, in familiar surroundings or during the day, the individual may prioritise speed above environmental safety (Kim et al., 2022). This methodology allows scientifically sound and practical judgements by adding user-defined avoidance levels to the grouping process. This also demonstrates how location-based systems that prioritise user control might apply the principle.

5.6.4 Limitations

Despite its benefits, DBSCAN has certain limitations in this circumstance. First, place-based grouping accuracy depends on data quality and quantity. This data originates from the massive Google Places API. Certain places may be biased or excluded from this data. Due to missing clusters or under-represented site groupings, rerouting proposals may be poor.

Second, the current setup prevents real-time upgrades. Town enterprises open and shut often. Without real-time data, the model may become ineffective, particularly in fast-changing sectors. This delay may affect the model's accuracy when organising an event or implementing a brief shutdown (Bushra and Yi, 2021).

The current design cannot prohibit multi-category. Avoid dangerous areas and poorly lit roadways. They may avoid bars and other venues. The present DBSCAN implementation only allows you to sort by one group, making it less useful as a safety-aware routing tool. Later versions might employ mixed data layers like environmental elements or criminal numbers to categorise more accurately and multidimensionally (Li and Yang, 2023). DBSCAN is scalable, useful, and easy to use for regional cluster analysis and route planning. The model must employ high-quality data, update in real time, and handle tough user decisions. Fixing these issues would improve the model.

6. RECOMMENDATIONS AND CONCLUSION

6.1 Recommendations

As proven in the previous chapter, DBSCAN-based grouping improves individual guidance in route planning engines. However, there are some strategic suggestions to make the system more helpful, adaptive, and successful in many scenarios. Technology advances, system integration, user experience, moral issues, and future theses are discussed. Evaluation provided information and restrictions to assist us in creating each concept. This technique depends on static data and is not sensitive to real-world changes. Real-time grouping adjustments would be nice. Real-time data sources like social check-ins and Google Live Popular Times may help. Real-time event streams may help the system adapt to current circumstances, speeding rerouting and improving dependability. Multi-category grouping may expand the system beyond bars and similar companies. Avoid places with a history of accidents, high crime, or demonstrations. Metadata tagging and sophisticated unsupervised place-type classification approaches like topic models or latent Dirichlet allocation might achieve this objective. Thus, the system's higher flexibility would benefit families, the elderly, and foreign visitors.

The system's logic should include time-sensitive grouping. Nightclubs appear bad at certain times. If the grouping approach used temporal aspects, the system might distinguish high-risk and low-risk time periods. This adjustment would provide flexibility to reduce over-avoidance and enhance journey time while meeting user constraints. DBSCAN performed well, but future versions could consider alternative autonomous learning algorithms. If density changes, OPTICS or HDBSCAN may function better. Models that combine DBSCAN with A* or Dijkstra's algorithm may enhance route modification. This stacking technique may balance machine learning insights with conventional improvement approaches. We advocate setting up an API to make the system more helpful and reach more users. A RESTful API would allow health monitoring and ride-sharing applications to adopt avoidance-based routes. This combination may benefit digital guidance users.

It should also prioritise Android and iOS native app development. A smartphone app that uses GPS to redirect users in real time lets them access information while driving. Make sure it works on all devices using Flutter or React Native. Voice assistance and real-time notifications will simplify and popularise the app. The system might be sold as an add-on or app instead of a navigation tool. Users may benefit from browser add-ons that add DBSCAN cluster notifications to Google Maps or Uber app capabilities. This strategy simplifies learning and works with popular systems. Improved user experience is equally vital. The system might employ machine learning to monitor user behaviour to improve route choices. Users' behaviour, such as not following the system's advice, may affect an autonomous avoidance model. Over time, this would make the experience more distinctive. Enhancing user experience is sensible too. The Folium and Leaflet.js visualisations are decent but might be enhanced. Cluster heatmaps, intensity indicators, and live comparison charts are ideal. Users might see how safety and speed compare, helping them make judgements.

6.2 Conclusion

User input is crucial for cluster accuracy testing and improvement. A public-approval method to identify inappropriate groupings might be valuable. User input, maybe weighted by fame, would improve the system's accuracy and flexibility over time. Social factors must be considered while designing avoidance-based route systems. Systems that let users avoid locations or venues might promote prejudice or guilt against particular populations. The system might avoid this problem by not pre-marking any region or group as undesired and depending on user selections. Any features intended to avoid defaults should be ethically reviewed. The right to privacy matters too. Place-based tools always include private data. Clear and permission-based data collection and management are required to comply with data security legislation like the GDPR. Use end-to-end encryption and data anonymisation wherever feasible. Additional security may be added via OAuth 2.0 or token-based access.

Regular comparison of unsupervised learning algorithms is vital for the thesis. Clustering, computing speed, and route planning impact would be utilised to compare them. Comparing mean-shift, spectral clustering, and affinity propagation may provide intriguing results. The psychological implications of avoidance-based routing are another area that needs a thesis. Do helping people avoid locations make them happy or miserable, or do they keep doing dangerous things to avoid them? Along with psychology and social scientists, ongoing theses may provide fresh perspectives on these difficulties. Merging with AR systems is another uncertainty. Phone augmented reality applications may assist users in comprehending and making real-world choices by visualising groups and avoiding zones. These features might be made using ARKit or ARCore. It should also consider shared learning. Since data is not stored in one location, the system may learn from user behaviour across devices. This strategy might improve grouping logic while safeguarding user data.

Overall, the system advances context-aware routing, although it requires further effort to be optimal. It should learn to handle real-time, multidimensional input. It must be available online, on mobile devices, and incorporated into popular systems to be helpful. Its ethical role is to preserve users' privacy and avoid detrimental societal norms. This allows college academics to combine machine learning, human-computer interaction, and psychology. These suggestions solve issues and explain how to make guidance tools better, more transparent, and more accountable in the future.

REFERENCES

Abideen, Z. U., Mazhar, T., Razzaq, A., Haq, I., Ullah, I., Alasmary, H., and Mohamed, H. G. (2023). Analysis of Enrollment Criteria in Secondary Schools Using Machine Learning and Data Mining Approach. *Electronics*, *12*(3), 694.

Ahmad, I. S., Bakar, A. A., Yaakub, M. R., and Muhammad, S. H. (2020). A Survey on Machine Learning Techniques in Movie Revenue Prediction. *SN Computer Science*, *1*(4), 235.

Akter, J., Roy, A., Rahman, S., Mohona, S., and Ara, J. (2025). Artificial Intelligence-Driven Customer Lifetime Value (CLV) Forecasting: Integrating RFM Analysis with Machine Learning for Strategic Customer Retention. *Journal of Computer Science and Technology Studies*, *7*(1), 249-257.

Alawad, H., and Kaewunruen, S. (2023). Unsupervised Machine Learning for Managing Safety Accidents in Railway Stations. *IEEE Access*, *11*, 83187-83199.

Al-Batah, M. S., Al-Kwaldeh, E. R., Wahed, M. A., Alzyoud, M., and Al-Shanableh, N. (2024). Enhancement Over DBSCAN Satellite Spatial Data Clustering. *Journal of Electrical and Computer Engineering*, *2024*(1), 2330624.

Ali, A., Khan, L., Javaid, N., Aslam, M., Aldegheishem, A., and Alrajeh, N. (2024). Exploiting Machine Learning to Tackle Peculiar Consumption of Electricity in Power Grids: A Step Towards Building Green Smart Cities. *IET Generation, Transmission & Distribution*, *18*(3), 413-445.

Amalina, D. N., and Fauzan, A. (2024). A Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) Approach for Identifying Potential Villages in Buleleng Regency. *Knowledge Engineering and Data Science*, 7(2), 187–199.

Ang, K. L. M., Seng, J. K. P., Ngharamike, E., and Ijemaru, G. K. (2022). Emerging Technologies for Smart Cities' Transportation: Geo-Information, Data Analytics and

Machine Learning Approaches. *ISPRS International Journal of Geo-Information*, *11*(2), 85.

Bajaj, P., Ray, R., Shedge, S., Vidhate, S., and Shardoor, N. (2020). Sales Prediction Using Machine Learning Algorithms. *International Research Journal of Engineering and Technology (IRJET)*, *7*(6), 3619-3625.

Bayliss, C. (2021). Machine Learning Based Simulation Optimisation for Urban Routing Problems. *Applied Soft Computing*, *105*, 107269.

Bhattacharjee, P., and Mitra, P. (2021). A Survey of Density Based Clustering Algorithms. *Frontiers of Computer Science*, *15*(1), 151308.

Bujel, K., Lai, F., Szczecinski, M., So, W., and Fernandez, M. (2018). Solving High Volume Capacitated Vehicle Routing Problem with Time Windows Using Recursive-DBSCAN Clustering Algorithm. *arXiv preprint arXiv:1812.02300*.

Bushra, A. A., and Yi, G. (2021). Comparative Analysis Review of Pioneering DBSCAN and Successive Density-Based Clustering Algorithms. *IEEE Access*, *9*, 87918-87935.

Candra, A. P. (2025). Analisis Data Menggunakan Python: Memperkenalkan Pandas dan NumPy. *Journal of Information System and Education Development*, *3*(1), 11-16.

Cheng, D., Xu, R., Zhang, B., and Jin, R. (2023). Fast Density Estimation for Density-Based Clustering Methods. *Neurocomputing*, *532*, 170-182.

Choi, C., and Hong, S. Y. (2021). MDST-DBSCAN: A Density-Based Clustering Method for Multidimensional Spatiotemporal Data. *ISPRS International Journal of Geo-Information*, *10*(6), 391.

Dhruv, A. J., Patel, R., and Doshi, N. (2021). Python: The Most Advanced Programming Language for Computer Science Applications. *Science and Technology Publications, Lda*, 292-299.

Dixon, M. F., Halperin, I., and Bilokon, P. (2020). *Machine Learning in Finance* (Vol. 1170). New York, NY, USA: Springer International Publishing.

El Hajj, M., and Hammoud, J. (2023). Unveiling The Influence of Artificial Intelligence and Machine Learning on Financial Markets: A Comprehensive Analysis of AI Applications in Trading, Risk Management, and Financial Operations. *Journal of Risk and Financial Management*, *16*(10), 434.

Fahle, S., Prinz, C., and Kuhlenkötter, B. (2020). Systematic Review on Machine Learning (ML) Methods for Manufacturing Processes–Identifying Artificial Intelligence (AI) Methods for Field Application. *Procedia CIRP*, *93*, 413-418.

Grinberger, A. Y., Minghini, M., Juhász, L., Yeboah, G., and Mooney, P. (2022). OSM Science—The Academic Study of The Openstreetmap Project, Data, Contributors, Community, and Applications. *ISPRS International Journal of Geo-Information*, *11*(4), 230.

Hadaj, P., Strzałka, D., Nowak, M., Łatka, M., and Dymora, P. (2022). The Use of PLANS And Networkx in Modeling Power Grid System Failures. *Scientific Reports*, *12*(1), 17445.

Hu, L., Liu, H., Zhang, J., and Liu, A. (2021). KR-DBSCAN: A Density-Based Clustering Algorithm Based on Reverse Nearest Neighbor and Influence Space. *Expert Systems with Applications*, *186*, 115763.

Huang, Z., Liang, Z., Zhou, S., and Zhang, S. (2025). An Improved Density-Based Spatial Clustering of Applications with Noise Algorithm with an Adaptive Parameter Based on the Sparrow Search Algorithm. *Algorithms*, *18*(5), 273.

Jang, J., and Jiang, H. (2019, May). DBSCAN++: Towards Fast and Scalable Density Clustering. In *International Conference on Machine Learning* (pp. 3019-3029). PMLR.

Jayathilaka, U. R., and Park, G. C. (2022). The Impact of Amazon Global Selling on Innovation Performance of SMEs. *Journal of Artificial Intelligence and Machine Learning in Management*, *6*(2), 1-13.

Kaponis, A., and Maragoudakis, M. (2022, September). Data Analysis in Digital Marketing Using Machine Learning and Artificial Intelligence Techniques, Ethical and Legal

Dimensions, State of the Art. In *Proceedings of the 12th Hellenic Conference on Artificial Intelligence* (pp. 1-9).

Kapoor, S., Cantrell, E., Peng, K., Pham, T. H., Bail, C. A., Gundersen, O. E., ... and Narayanan, A. (2023). Reforms: Reporting Standards for Machine Learning Based Science. *arXiv preprint arXiv:2308.07832*.

Kim, J., Justin, C., Mavris, D., and Briceno, S. (2022). Data-Driven Approach Using Machine Learning for Real-Time Flight Path Optimization. *Journal of Aerospace Information Systems*, *19*(1), 3-21.

Kumar, S., Sharma, D., Rao, S., Lim, W. M., and Mangla, S. K. (2025). Past, Present, and Future of Sustainable Finance: Insights from Big Data Analytics Through Machine Learning of Scholarly Research. *Annals of Operations Research*, *345*(2), 1061-1104.

Kureljusic, M., and Reisch, L. (2022). Revenue Forecasting for European Capital Market-Oriented Firms: A Comparative Prediction Study Between Financial Analysts and Machine Learning Models. *Corporate Ownership & Control*, *19*(2), 159-178.

Lei, H., and Cailan, H. (2021, January). Comparison of Multiple Machine Learning Models Based on Enterprise Revenue Forecasting. In *2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)* (pp. 354-359). IEEE.

Li, H., and Yang, Z. (2023). Incorporation of AIS Data-Based Machine Learning into Unsupervised Route Planning for Maritime Autonomous Surface Ships. *Transportation Research Part E: Logistics and Transportation Review*, *176*, 103171-103171.

Miranda, F. M., Köhnecke, N., and Renard, B. Y. (2023). Hiclass: A Python Library for Local Hierarchical Classification Compatible with Scikit-Learn. *Journal of Machine Learning Research*, *24*(29), 1-17.

Prasad, R. K., Sarmah, R., Chakraborty, S., and Sarmah, S. (2023). NNVDC: A New Versatile Density-Based Clustering Method Using K-Nearest Neighbors. *Expert Systems with Applications*, *227*, 120250.

Priyadarshi, R. (2024). Exploring Machine Learning Solutions for Overcoming Challenges in IoT-Based Wireless Sensor Network Routing: A Comprehensive Review. *Wireless Networks*, *30*(4), 2647-2673.

Pundir, A. K., Ganapathy, L., Maheshwari, P., and Kumar, M. N. (2020, November). Machine Learning for Revenue Forecasting: A Case Study in Retail Business. In *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 0201-0207). IEEE.

Ran, X., Xi, Y., Lu, Y., Wang, X., and Lu, Z. (2023). Comprehensive Survey on Hierarchical Clustering Algorithms and The Recent Developments. *Artificial Intelligence Review*, *56*(8), 8219-8264.

Samuel, S., and Mietchen, D. (2024). Computational Reproducibility of Jupyter Notebooks from Biomedical Publications. *GigaScience*, *13*, giad113.

Sharma, A., Poojitha, S., Saxena, A., Bhanushali, M. M., and Rawal, P. (2022, December). A Conceptual Analysis of Machine Learning Towards Digital Marketing Transformation. In *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)* (pp. 313-316). IEEE.

Shilong, Z. (2021, January). Machine Learning Model for Sales Forecasting by Using XGBoost. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)* (pp. 480-483). IEEE.

Shiri, F. M., Perumal, T., Mustapha, N., and Mohamed, R. (2023). A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU. *arXiv preprint arXiv:2305.17473*.

Si, H., Hou, X., Wang, J., Boateng, G. O., Zhang, Z., Guo, X., and Niyato, D. (2025). Unsupervised Localization Toward Crowdsourced Trajectory Data: A Deep Reinforcement Learning Approach. *IEEE Transactions on Wireless Communications*.

Sriprateep, K., Pitakaso, R., Khonjun, S., Srichok, T., Luesak, P., Gonwirat, S., ... and Enkvetchakul, P. (2024). Multi-Objective Optimization of Resilient, Sustainable, and Safe Urban Bus Routes for Tourism Promotion Using a Hybrid Reinforcement Learning Algorithm. *Mathematics*, *12*(14), 2283.

Ullal, M. S., Hawaldar, I. T., Soni, R., and Nadeem, M. (2021). The Role of Machine Learning in Digital Marketing. *Sage Open*, *11*(4), 21582440211050394.

Vineeth, V. S., Kusetogullari, H., and Boone, A. (2020, August). Forecasting Sales of Truck Components: A Machine Learning Approach. In *2020 IEEE 10th International Conference on Intelligent Systems (IS)* (pp. 510-516). IEEE.

Yang, M., Shi, Y., Song, Q., Wei, Z., Dun, X., Wang, Z., ... and Cheng, X. (2025). Optical Sorting: Past, Present and Future. *Light: Science & Applications*, *14*(1), 103.

Zhang, R., Peng, H., Dou, Y., Wu, J., Sun, Q., Li, Y., ... and Yu, P. S. (2022, October). Automating DBSCAN via Deep Reinforcement Learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (pp. 2620-2630).

Zhang, X., and Zhou, S. (2023). WOA-DBSCAN: Application of Whale Optimization Algorithm in DBSCAN Parameter Adaption. *IEEE Access*, *11*, 91861-91878.