Digital Object Identifier

ClearPath NYC: Enhancing Jogger and Walker Safety through Optimized Route Recommendations

ADAM. HERDMAN, ADAM. SOUTHWICK, DILLON. HENNESSY, MICHAEL. MCKEON, NIAMH. WALSH, AND YUHAN. HU.
Group 3. GitHub repository accessible at: https://github.com/Southwick-Adam/ClearPathNYC

ABSTRACT The urban challenges of New York City, such as overcrowded streets, littered streets and unsafe locations, highlight the need for an optimised navigation system for joggers and walkers. In this paper, we present ClearPath NYC, an innovative route recommendation system designed to enhance the outdoor experience for users by suggesting optimal routes that avoid the aforementioned undesirable areas. Utilising GIS technology and municipal data, routes are tailored based on advanced data modelling and user preferences. This paper explores the methodologies used in developing the system, including data collection, processing techniques and algorithmic route optimisation. Through evaluating this first iteration via usability questionnaires, this paper demonstrates the application's impact on enhancing user satisfaction and security. Accessible at **https://clearpath.info.gf/**, this application provides a practical tool for New Yorkers seeking a more enjoyable outdoor experience. ClearPath NYC stands as a testament to the potential of technology in solving urban navigation challenges.

I. INTRODUCTION

ClearPath NYC is a web application designed to generate routes for joggers and walkers within the Manhattan borough of New York City. Using predictive modelling based upon gathered data, the application presents the user with a route that is based around a quantifiable "Quiet Score". This aims to address the common issue of disturbances, crowds and noisy areas that many people encounter while walking.

At the core of the project lies the concept of busyness (specifically human-busyness), defined as "concerned with density and congestion of people in an area" [1]. Busyness affects the comfort and safety of pedestrians. Many existing route planners prioritise the quickest path, often disregarding the desirability of the route in terms of busyness. In contrast, ClearPath NYC focuses on providing routes that enhance the overall user experience by prioritising quieter, more pleasant paths. The application includes a feature that allows users to toggle between quiet and busy routes, giving flexibility of choice based on immediate context and preferences. For instance, a quieter route may be preferred by those seeking a quiet walk, while a busier route with more foot traffic may be preferred by women walking alone at night who have concerns about safety.

In this paper, we provide a comprehensive literature review focusing on navigation systems and route optimisation, particularly in urban environments. We then examine the design of our application ClearPath NYC, outlining the system architecture and key features, including the data integration and user customisation options, along with user feedback on the product. Finally, we discuss the overall outcomes of the project and potential directions for future development.

II. LITERATURE REVIEW

The concept of optimising urban routes for joggers and walkers by addressing disturbances is a relatively unexplored niche. This literature review aims to synthesise existing research on correlated topics, including but not limited to perceptions of crowding and the specific needs of joggers and pedestrians. By combining insights from these diverse fields, this review highlights key studies and theoretical frameworks, thus demonstrating the purpose and potential impact of ClearPath NYC.

As one of the most popular physical activities in the world [2], running is an easily accessible sport to all. While a diverse range of people engage in running, the vast majority (over 75%) utilise some form of wearable technology or running application. The work of [3] explores the commonalities of those individuals who utilise running applications, finding that casual individual runners were those most likely to utilise running applications and technologies. The paper argues that running applications can be used to combat dropout rates from running and ‘close the gap between runners’ expectations and actual experience’. This hypothesis is corroborated by [4], which demonstrates that the integration of route planning to training variables leads to increased commitment and consistency in urban runners.

Several studies have explored the factors that create a pleasant environment for runners and walkers. However, the literature is sparse when it comes to exploring these two user bases as one, with a tendency for research to focus solely on pedestrians or joggers. However, it is unanimously found that safety is a core concern for both demographics [5], [6], [7]. In particular, [6] finds that safety and security concerns override any other positive environmental factors. While the authors note a myriad of environmental factors and the positive psychological impacts they have on pedestrians, they do not deeply explore the interplay between the differing characteristics.

While there is brief reference in the literature to urban trash, it is typically in the context of urban planning and not how it impacts on the public’s experience of the urban environment. [8] explores the long-term impact of trash on an area and finds that, when litter is left in an area for a period of time and then removed, people will continue to litter it to the former degree of uncleanliness despite cleaning efforts. When combined with the work of Lovasi et. al [9], who identify street cleanliness as a key factor in New Yorkers’ uptake of walking and bicycling, it can be concluded that the presence of garbage in Manhattan streets severely hampers pedestrians and joggers in the city. However, more evidence is needed in this area in order to make concrete recommendations.

It is impossible to ignore the impact of crowds in urban areas on the choices of pedestrians and joggers, particularly in the context of Manhattan which has the highest population density of the entire United States [10]. According to [11], crowding is often a subjective psychological experience rather than a quantifiable density. As such, the authors note that determining how crowded is too crowded is difficult. This is echoed by [12], which also adds that access to features such as parks and clean streets can favourably influence perceptions of crowdedness. When compiled with [13], which argues that the perception of crowded or busy places is heavily influenced by cultural and societal norms, it is clear that there is no unanimous definition of what constitutes a crowded location.

However, while the literature struggles to find consensus on the definition of human-busyness, it is clear on the impact that said busyness has on runners and pedestrians. [14] finds that pedestrians will typically opt to avoid congestion, although vulnerable persons such as lone women or people unfamiliar with their environments will prefer congested routes. The paper also finds that when pre-planned routes are utilised, pedestrian adherence to the planned travel is raised, even in the face of overly busy areas. The work of [15] concurred with this in finding and utilised their understanding of busyness to develop a points-weighted Safety and Accessibility Index (SAI). Through analysis of their SAI, it was revealed that while busyness was a consistent negative for cyclists, for pedestrians and joggers it offered an element of security when other negative factors such as low safety levels are present.

In conclusion, while we have encountered no application that provides the same service as our product, the exploration of the literature reveals a clear gap in the market, underscored by solid, research-backed logic. The reviewed studies demonstrate that our product would be effective, marketable and useful by addressing critical factors such as cleanliness, safety and route optimisation. By integrating these elements, the urban experience can be significantly enhanced for joggers and walkers, promoting higher public engagement and increased physical health.

III. METHODOLOGY

A. ARCHITECTURE - MONOLITHIC TO MICROSERVICES

The application was initially designed as a monolithic architecture, combining frontend logic, database update programs, and backend services into a single module. This approach was logical, given ASP.NET has JavaScript React integration templates, and development was smooth. However, a mentor suggested incorporating an API microservice, which led to a project restructuring to enhance functionality and development capabilities of both the frontend and backend teams. The app’s routing capabilities are now handled through API calls, allowing other projects to implement its functionality by calling the ClearPath NYC endpoints. As the backend team developed the API endpoints, it was decided that separating all service interactions into microservices would reduce the complexity issues arising from the monolithic architecture’s interactions with the database. Microservices also appealed to our maintenance lead, who recognized the potential for improved scalability and security through the use of separated containers and reverse proxying. Most importantly, shifting to individual microservices would grant each team greater freedom to expand their own services without conflicting with one anther, thanks to the abstraction offered by this change.

Our maintenance lead restructured the application into four distinct microservices which can be seen in Figure 1. A React based frontend, an ASP.NET backend for serving the API endpoints and querying the database, a Flask based service for interacting with our pickle formatted data models, and a second ASP.NET service for pulling data from the model service and updating our database, which our application does every hour.

All of these services were containerized using Docker for easy deployment, as was the database itself. The team chose to use a containerized database for increased scalability, and allowing for swift deployment of different versions of the database as querying optimisations were deployed. This reduced the load on the server which multiple database instances would have incurred. Using Docker containers streamlined deployment testing, allowed updates to individual services without disrupting the application, and provided flexibility to add new services and pinpoint specific errors without redeploying the entire app.

Our maintenance lead created a final reverse proxy service with Nginx to provide load balancing, facilitate easy communication between the containers, and add a layer of security between the users and the application itself by enabling and enforcing HTTPS protocols.

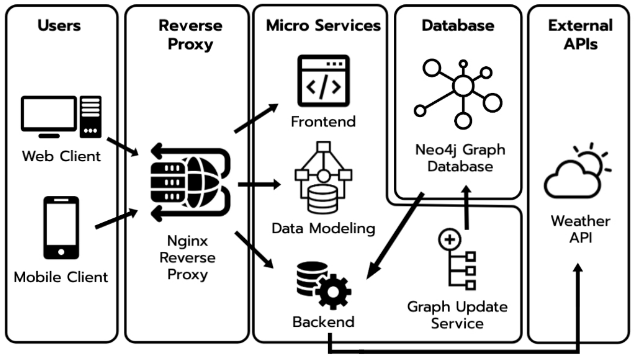

FIGURE 1. Architectural Diagram

B. GIT VERSION CONTROL AND DEPLOYMENT

Initially designed for a monolithic architecture, Git protocols were adapted to separate the teams for a concise codebase. These protocols were retained even after transitioning to distinct services, ensuring simple work differentiation and ease of deployment and testing. Each team (frontend, backend, data) managed a dedicated branch for their features. Upon completing a feature block, the code would be pushed to the 'dev' branch, which enabled compatibility testing with the other microservices, allowing identification and resolution of cross-service issues. At the end of each sprint, functioning code was merged into the 'main' branch and automatically deployed using GitHub Actions. Strict branch separation and mandatory testing and reviews maintained a functional codebase, enabling simultaneous team work without interference.

C. ASP.NET / NEO4J CHOICE RATIONALE

1) ASP.NET rationale and libraries used

The backend lead chose to use ASP.NET as the backend framework. ASP.NET can be used via many languages such as Visual Basic, C# or J#. C# was chosen as it has similarities to a language the team used last semester, Java, and due to its wide use in industry. ASP.NET was chosen for its performance, cross platform abilities, and security. ASP.NET also provided support for a wide array of packages and Neo4j, which appealed to the team. ASP.NET supports cross-platform developing, and with the team using a range of systems and aiming to deploy to Linux this was a key factor in the choice. We made use of the built-in security features, like the CORS policies, finding them intuitive to update for development environments and production. Lastly, C# and ASP.NET both have extensive documentation and community support, which the backend team frequently consulted to solve issues and optimize performance throughout the project.

C has many libraries available for use, and the backend team made use of several throughout the project. Some libraries include:

- NetTopologySuite: This library was used during the database construction phase to handle GeoJSON objects, a format for encoding and creating geographic data structures. NetTopologySuite facilitated node selection based on taxi zones and the Manhattan area.
- TinyCsvParser: This lightweight library was employed for parsing CSV data. It enabled the importing of data from various sources, such as 311 data and metro data.
- DotNetEnv: In the AspRun and AspBuild project folders, which were used for querying and updating the database, DotNetEnv was used for hidden key storage. This library helps manage environment variables securely.
- Swashbuckle: To facilitate API endpoint testing, Swashbuckle was employed, which integrates Swagger into the project. Swagger provides a user-friendly interface for testing and documenting API endpoints.

2) Neo4j rationale

Neo4j was chosen instead of a traditional relational database like MySQL or PostgreSQL. Neo4j is a graphical database, which suited the goals of the assignment. It enabled handling of the database of nodes and relationships, as well as parsing and traversing these quickly. Neo4j has additional libraries to support a range of path finding between nodes, of which we used A* shortest path and Yen's shortest path in the Graph Data Science Library. This was crucial to the team as it meant fast, reliable functions which reduced the development time and complexity.

D. NEO4J DATABASE CREATION AND DATABASE UPDATE

1) Database creation

The Neo4j Database was created using Open Source Maps data. The program first parsed a series of spatial 'nodes' carrying data such as coordinates and various tags. Certain tags, such as 'freeway', were excluded from the selection, based on testing results. Other data, such as metro and taxi zones, and whether the node was in a park, were added based on external datasets. These 'nodes' were added to the database, along with 'PATH' relationships between the 'nodes' that held the distance between them, as well as the quiet and loud scores. The extensive and user generated nature of the tags system posed challenges, as there is no definitive database of tags. There are also combinations of tags which signify different meanings, further complicating the system. A detailed examination of tags before starting was beneficial in streamlining the database setup early in the process and reducing development issues later on. However, many lessons on what to exclude were learned from examining nodes that raised errors, requiring continuous fine tuning of the database as the project developed.

2) Database update

The database update process first requests up-to-date predictions from the models for both the taxi and metro zones, storing this information in dictionaries to be referenced during each node update. The process then queries the Neo4j database to retrieve all nodes within a taxi zone. It calculates the 'quietscores' and 'loudscores' for each node, updating it in the database.

E. OPTIMISATION FOR NEO4J DATABASE UPDATES AND CYPHER QUERIES

The objective was to optimise the database update process for loud and quiet scores, used for predicting the path between nodes. Initially, the process took approximately 12 hours and 30 minutes, which was not feasible for the target update frequency of once an hour. Through the use of the cypher query 'PROFILE' and code base analysis, a series of optimizations were identified and implemented. The following optimizations were tested on a sample of 2891 nodes. The first optimization was to add in a constraint on the nodeid to ensure its uniqueness, allowing for less database hits per nodeid lookup, the next was to change the cypher query structure, and the last in this series of optimisations was to add a composite index. This reduced the segment update time from 174s to 26s. The end result was a predicted database update of 90 minutes. Further database examination identified that the taxi zone and metro zones were strings and may be more efficient as integers. This was initially set to string types as metro zones had tram1 and tram2, which were entered as strings. These were changed to 1 and 2, as there was no conflicting metro zone. Both metro and taxi zones were now entered as integers, lowering the update time to 20 minutes for the database. Lastly, the team examined reducing the relationships from 2 between nodes to 1. This was feasible as direction was no longer used and minimal updates to the code were required. This reduction further reduced the update time to 15 minutes.

F. LOOP AND POINT TO POINT FUNCTIONALITY ON THE BACKEND

1) Loop Functionality

The loop endpoint accepts a start coordinate as a list of doubles, a boolean indicating either the quiet or loud option, and the distance as a double. It then passes this to the loop function, creating a shape of between 3 to 10 sides and divides the total distance into segments of the specified length. The points, created by plotting the lengths at set angles, are called as the inputs to the A* path finding algorithm. The results are passed to the GeoJSON function, which returns a GeoJSON string and is returned to the initial endpoint request.

2) Point to point functionality

The 'Point to Point' functionality requires two lists of doubles, which are the required longitude and latitudes, and the boolean indicating the quiet or loud option. There are two endpoints for this with the identical inputs, allowing users to choose whether to use the single or multi path route options. The two lists of doubles enable scaling the number of coordinates sent by frontend or API users. The single path endpoint uses A* path finding between the coordinates, while the multi path route uses Yen's algorithm for finding the k-shortest paths, an adaptation of breadth-first search. Due to server constraints, k is set to 2, and the first two routes are returned. Different values were tested and the trade offs resulted in choosing k as 2. Both endpoints return a GeoJSON to the initial endpoint call.

G. FRONTEND RATIONALE

1) Frontend Technology Stack

React was chosen over other JavaScript frameworks for its component-based architecture and virtual DOM. ClearPath NYC reuses many UI elements, like the 'busy toggle switch' and location input bar in both the 'Loop' and 'PointToPoint' path finders. React's hierarchical structure improves maintainability and enhances performance by updating only components with state changes.

Moreover, React's extensive ecosystem offers various inbuilt libraries and external resources. Key frontend libraries for ClearPathNYC include React-bootstrap, Zustand and Mapbox GL JS. React-bootstrap provides adaptable UI elements for different screen sizes, enhancing responsiveness. For example, the sidebar, buttons, and toggle switches use customized bootstrap elements. Zustand manages global states, offering a simple, centralized state storage accessible by all React components, reducing complexity compared to libraries like Redux. This is crucial for conveying state changes across different hierarchies in the React component tree. Mapbox GL JS provides geocoding services and customizable map styling. It enables ClearPath NYC to render map visuals, offer location-search suggestions, and integrate various geospatial and visual sources seamlessly. Examples include the start location input bar and the dynamically rendered 'night-mode' map based on the time of day.

2) User Interface Design

The frontend design focuses on hide-able elements for greater visual clarity, especially when the map is the user's main focus. The web app's visual data representation necessitates an unobstructed view. Vertical sidebars from the left and right sides of the page adapt to both desktop and mobile interface, resizing elements to fit the screen. The left sidebar contains the core user interface for route planning, consolidating controls in one spot to minimise user confusion. In the mobile layout, the sidebar occupies the entire screen width, presenting a narrow set of options to reduce misinterpretation. The legend sidebar has a slight transparency effect to allow users to see the underlying map, as the legend only contains simple toggle filters for map markers. The main sidebar is opaque so as to not distract from the primary controls.

To enhance adaptability, the web app includes visual options like a day and night mode toggle, offering two colour schemes to accommodate different lighting conditions. A green colour scheme was chosen to align with the focus on greener nodes when generating routes, with buttons and interactive elements in a consistent green hue.

A colourblind mode button was added based on user feedback about the green to red colour scale representing busyness. This mode changes green values to purple, improving accessibility for users with Deuteranopia, Protanopia, or Tritanopia. Adobe Color's accessibility design tool [16] confirmed the new scheme's effectiveness. Figure 2. provides an illustration of how the user interface elements integrate in ClearPathNYC.

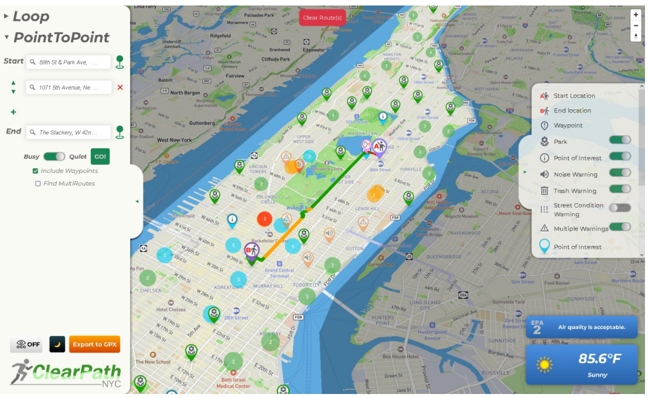

FIGURE 2. User Interface Layout

3) GPX route download

The team opted not to implement a user system, providing instead a button for users to download their generated routes in GPX format. GPX, an XML file format for storing coordinate data, is compatible with applications like Strava and Garmin Explore. This decision was made because the routes are designed for immediate use, as they are generated based on current busyness levels which will change over time. The download button also reduces server overhead and eliminates data storage requirements, such as those mandated by the European General Data Protection Regulation.

IV. DATA ANALYTICS AND VISUALISATION

A. DATASETS

1) Primary Datasets

The primary datasets were used to calculate the busyness of Manhattan. The baseline for this was the NYC Dot Pedestrian Mobility Plan Pedestrian Demand dataset, which categorises each street by pedestrian needs [17]. 'Baseline streets' have low pedestrian volumes and are typically residential, while 'global corridors', making up 0.5% of roads, are the busiest.

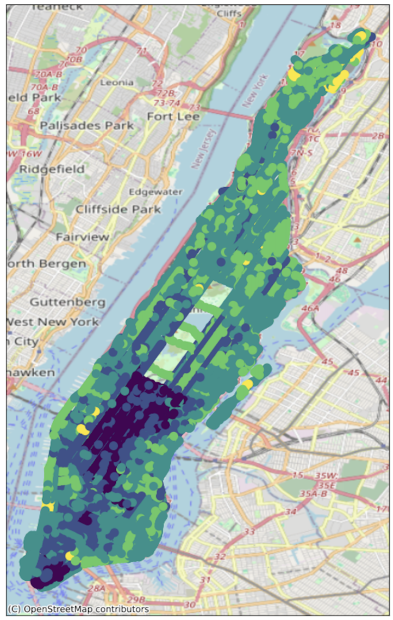

FIGURE 3. Pedestrian Mobility Plan Pedestrian Plan

The other primary datasets were the TLC Trip Record Data [18] and MTA Subway Hourly Ridership [19]. [18], divided into zones, indicated busy areas by day and time, while [19] measured station footfall. The combination of these comprehensive, current datasets enabled accurate predictions of Manhattan's busyness.

In reviewing the literature on these large datasets, a strong relationship between subway and taxi data in predicting city busyness was noted [20], [21]. These datasets are frequently used in urban planning, validating our choice to include them. Additionally, [22] offers a framework for using large public transport datasets, supporting the notion that combining multiple data sources enhances research accuracy.

2) Secondary Datasets

Secondary datasets refined the route-finding algorithm by incorporating factors not covered by the primary data, such as bad smells, street trash, parks, and construction. These considerations helped improve the application's route optimisation. The secondary datasets consist of the NYC 311 Service Requests [23] and NYC OpenData Points of Interest [24]. Given that both datasets are compiled by governmental authorities, they were exceedingly thorough and well labelled. These datasets were cleaned and utilised in both front end and back end work.

B. DATA CLEANING AND MANIPULATION

1) TLC Trip Record Data

The TLC Trip Record Data and MTA Subway Hourly Ridership datasets, with millions of rows, qualify as big data. However, quantity does not always correspond to quality (as per

[25]), which necessitates data cleaning and pre-processing. Duplicate entries and null values were removed, making data recovery unnecessary. Essential features were selected, and unnecessary ones were removed to save memory. The datetime feature was split into hour, day, month, and year, all crucial for prediction models.

The data was grouped by location and datetime, summing the passenger count. PickupId and DropoffID locations were merged into a single LocationID, with separate rows for different locations. This format allowed analysis of busyness by location, with insights on time of day, day of week, and time of year.

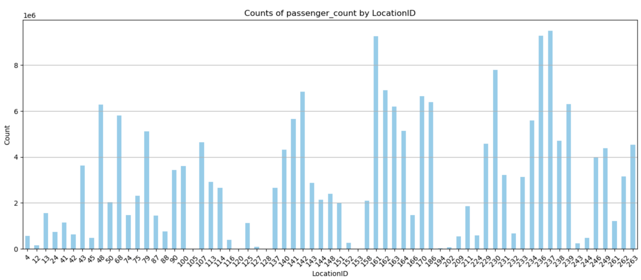

FIGURE 4. Passenger Count per LocationID (TLC Trip Record Data)

2) Subway Hourly Ridership

Unique longitude and latitude points differed from the number of stations, as stations had multiple coordinates. We rounded these coordinates and used the mode for each station. The data was then grouped by datetime and location, with ridership summed to evaluate subway busyness.

3) Busyness Rank

The busyness rank for the TLC Trip Record Data and MTA Subway Hourly Ridership was calculated using the same approach as the Pedestrian Mobility Plan. Quantiles were employed to bin the data based on passenger count and ridership. This busyness rank serves as the target feature for modelling and underpins our route planner.

C. MODELLING

1) Data Manipulation

The data was split 30/70 for testing and training, and further manipulated to enhance model accuracy. Weighted data based on feature importance was tested by creating duplicate LocationID features, but the accuracy improvements were minimal compared to the increased model size. Other manipulation methods, including balanced and encoded data, showed no significant accuracy gains. Consequently, the original dataset was used for modelling.

2) Model Selection

a: Logistic Regression and Ordinal Logistic Regression

Classification models were preferred over regression due to the categorical nature of busyness ranks. Various classification models were tested on the TLC Trip Record Data and applied to the MTA Subway Hourly Ridership, given their similar data formats. Logistic Regression was tried but struggled with intermediate classes, likely due to its simplicity, making it more suited for binary classification. Ordinal Logistic Regression, chosen for its ability to handle ordered labels, faced similar challenges with intermediate class accuracy, proving ineffective for our needs.

b: Random Forest and XGBoost

Due to the limitations of our logistic regression models, decision-tree models were explored to improve accuracy, especially for intermediate classes crucial to our application. The Random Forest Classifier showed significant accuracy improvements, though it still struggled with intermediate classes. XGBoost, a Gradient Boosting Decision Tree (GBDT) model known for its superior performance over Random Forest [26], was subsequently fitted. The XGBoost model significantly outperformed Random Forest, demonstrating higher accuracy and precision for each busyness rank.

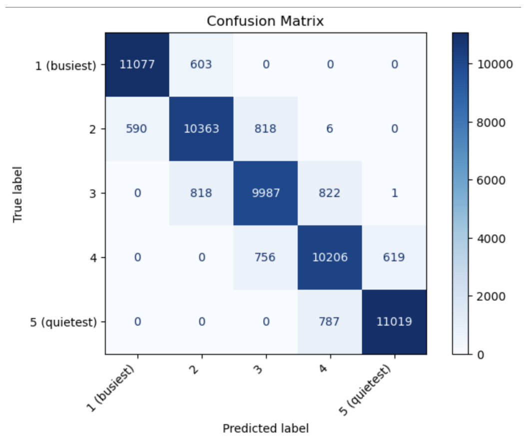

FIGURE 5. XGBoost Confusion Matrix

c: Ensemble and Stacking

Despite XGBoost's satisfactory performance, ensemble methods were explored to further enhance precision and accuracy, as supported by Tong et al. [27], who demonstrated that combining multiple independent decision tree models can improve accuracy. Combining XGBoost and Random Forest models led to a slight accuracy decrease. Stacking XGBoost with logistic regression as a meta-model achieved the highest accuracy but was more resource-intensive. Consequently, the slightly less accurate but more efficient XGBoost model was preferred to manage modelling overhead.

V. EVALUATION AND RESULTS

During initial development, user evaluation of ClearPath NYC primarily came from spontaneous feedback during the in-class presentations, the academic mentor meetings, and the UI-testing conducted by group members. Many suggestions for improvements were received, such as the inclusion

Model	Accuracy
Logistic Regression	0.33403
Ordinal Logistic Regression	0.31746
Random Forest	0.81048
XGBoost	0.90046
Ensemble (XGBoost, RF)	0.89713
Stacking (Logistic Regression)	0.90327

TABLE 1. Model accuracy table.

of a colour-blind mode for better accessibility and a route-loading popup for enhanced user experience. These features are included in the final product.

For a systematic evaluation, after careful consideration of all available options we elected to use a Computer System Usability Questionnaire (CSUQ). The CSUQ offers a comprehensive assessment through a wide range of questions and includes both positive and negative feedback, providing detailed insights and targeted ways to improve the application [28].

Survey feedback was positive regarding the ease of use and organisation of information. However, there were suggestions for additional functionalities during the development process. Users requested a mobile implementation, which was one of our stretch goals and was subsequently implemented.

Users appreciated features such as the day-to-night toggle, the "sleek" UI, exporting to GPX, and the grouping of information on the map. Suggestions for improvement included reformatting the legend, adding a clear route button, and enhancing the contrast between loud and quiet routes. These suggestions were all incorporated into the final version of the app. One feature we did not manage to include, as suggested by the CSUQ, was explanatory text showing users how to rotate the map view. The action for this improvement is discussed in detail in the next section.

VI. CONCLUSION AND FUTURE WORK

A. FUTURE WORK

The team identified a number of improvements that they would include given more time or with funding.

The current database is not optimised for producing multiple routes, as it counts each side of the road as separate paths, in line with the source, Open Source Maps. This results in numerous small changes available to the path finder. If the path is set as the 100th to reach the node, it is still a small difference compared to the 1st path. While database adjustments could improve this, time constraints prevented implementation in this project.

Users currently lack visual feedback during route generation. The team discussed adding a heat-map feature, but it was deferred for future work. This feature could be implemented by plotting each relationship on the map, although this would be cluttered and time-consuming. Alternatively, plotting taxi and metro zones (defined as a 400-metre radius around metro stops) and assigning colours based on busyness levels would offer a clearer visual representation.

We identified numerous datasets providing real-time updates and accurate forecasts of Manhattan street busyness. Companies like Google offer datasets with up-to-date street-level vehicle traffic information, and some datasets use mobile phone location data for precise foot traffic data. However, accessing these datasets is prohibitively expensive for this project. An expanded application could utilise live busyness updates for more accurate route creation. Accessing a few years of past data would still allow more effective model training.

To improve user experience, additional UI features could be included. Although ClearPath NYC has a straightforward UI design and pop-ups explaining most functions, some features, such as map rotation, may not be intuitive. An interactive tutorial could guide users through key functions.

Enhancing the route-marker drawing function could improve the user experience. Currently, it plots all 311 markers within the view-box of the route, instead of only the closest ones. A possible solution is to create a square based on every 10th coordinate in the provided route and link these squares into a polygon unique to each route. Testing different coordinate numbers will need to be conducted to finely balance the performance and user experience, as this solution may involve longer route-plotting time. Furthermore, an informational popup feature above the plotted routes may be implemented to show the exact length of the route.

Another future improvement would be to allow the users to set non-park or non-POI map locations as route waypoints. Finally, allowing the users to alter the route shape after route generation would also add to the user experience by offering more freedom to customise routes according to personal preferences.

B. CONCLUSION

In conclusion, this paper has explored the use case and motivation behind 'ClearPathNYC', a route generator designed to address the dual needs of avoiding or seeking busy areas for enhanced safety and convenience. By identifying a unique gap in the market, 'ClearPathNYC' offers innovative routing solutions tailored to user preferences. The development process, including the frameworks and libraries utilized, as well as the challenges encountered, has been comprehensively detailed, highlighting both the current capabilities and potential future enhancements of the system.

Insights on data analytics, from dataset choice to modeling techniques and results, were explored. User testing via CSUQ offered valuable feedback, emphasizing the importance of user-centered design. This iterative process led to refinements and identified areas for future development, showcasing the system's future direction.

Overall, 'ClearPathNYC' represents a significant step forward in personalized routing solutions, combining technological innovation with practical user feedback to create a safer and more efficient urban navigation tool.

REFERENCES

[1] A.J. Ashmore, “Estimating Spatial-Busyness Using Wi-Fi Traffic Filtering,” B.Sc dissertation, Dept. Comp. Sci, Maynooth Univ., Maynooth, Kildare, 2020.
[2] J. Andersen, "The State of Running 2019," runrepeat.com. https://runrepeat.com/state-of-running (accessed Jun. 30, 2024).
[3] M. Janssen et al., “Understanding Different Types of Recreational Runners and How They Use Running-Related Technology,”Int J Environ Res Public Health, vol. 17, no. 7, Mar, 2020. DOI: 10.3390/ijerph17072276.
[4] D. Parra-Camacho, M. de Santos, M. Gonzalez-Serrano, “Amateur Runners’ Commitment: An Analysis of Sociodemographic and Sports Habit Profiles,”Int J Environ Res Public Health, vol. 17, no. 3, Feb, 2020. DOI: 10.3390/ijerph17030925.
[5] I. Deelen, M. Janssen, S. Vos, C. Kamphuis, D. Ettema, “Attractive running environments for all? A cross-sectional study on physical environmental characteristics and runners’ motives and attitudes, in relation to the experience of the running environment,”BMC Public Health, vol. 19, no. 1, Feb, 2019. DOI: 10.1186/s12889-019-6676-6.
[6] C. Sundling, M. Jakobsson, “How Do Urban Walking Environments Impact Pedestrians’ Experience and Psychological Health? A Systematic Review,”Sustainability, vol. 15, no. 14, Jul, 2023. DOI: 10.3390/su151410817.
[7] D. Ettema, “Runnable Cities: How Does the Running Environment Influence Perceived Attractiveness, Restorativeness, and Running Frequency?,”Env and Behaviour, vol. 48, no. 9, Jul, 2015. DOI: 10.1177/0013916515596364.
[8] R. Dur, B. Vollaard, “The Power of a Bad Example: A Field Experiment in Household Garbage Disposal,”Env and Behaviour, vol. 47, no. 9, Jul, 2014. DOI: 10.1177/0013916514535085.
[9] G. Lovasi et.al, “Aesthetic Amenities and Safety Hazards Associated with Walking and Bicycling for Transportation in New York City,”ann. behav. med., vol. 45, no. 1, Feb, 2013. DOI: 10.1007/s12160-012-9416-z.
[10] United States Census Bureau, "Highest Density," census.gov. https://www.census.gov/popclock/embed.php?component=density (accessed Jun. 26, 2024).
[11] J. Vaske, L. Shelby, “Crowding as a Descriptive Indicator and an Evaluative Standard: Results from 30 Years of Research,”Leisure Sciences, vol. 30, no. 2, Feb, 2008. DOI: 10.1080/01490400701881341.
[12] B. Neuts, D. Vanneste, “Contextual Effects on Crowding Perception: an Analysis of Antwerp and Amsterdam”Royal Dutch Geo Soc, vol. 109, no. 3, Jul, 2018. DOI: 10.1111/tesg.12284.
[13] B. Neuts, P. Nijkamp, “Crowding Perception in a Tourist City: A Question of Preference”Tinbergen Institute Discussion Papers, Sep, 2011. DOI: 10.2139/ssrn.1938174.
[14] Y. Tong, N. Bode, “Higher investment levels into pre-planned routes increase the adherence of pedestrians to them”Psych and Behav, Jul, 2021. DOI: 10.1016/j.trf.2021.07.019.
[15] P. Koh, Y. Wong, “Influence of infrastructural compatibility factors on walking and cycling route choices”J of Env Psych, Aug, 2013. DOI: 10.1016/j.jenvp.2013.08.001.
[16] Adobe Color, "Color Checker," color.adobe.com. https://color.adobe.com/create/color-accessibility (accessed Jul. 24, 2024).
[17] New York City Department of Transport, 2023, "NYC DOT Pedestrian Mobility Plan", Department of Transport, New York. [Online]. Available: https://www.nyc.gov/html/dot/html/pedestrians/pedestrian-mobility.shtml
[18] NYC Open Data, 2023, "NYC DOT Pedestrian Mobility Plan Pedestrian Demand", Department of Transportation, New York. [Online]. Available: https://data.cityofnewyork.us/Transportation/Pedestrian-Mobility-Plan-Pedestrian-Demand/fwpa-qxaf/about$_data$.
[19] Metropolitan Transportation Authority, 2024, "MTA Subway Hourly Ridership: Beginning February 2022", Department of Transport, New York. [Online]. Available: https://data.ny.gov/Transportation/MTA-Subway-Hourly-Ridership-Beginning-February-202/wujg-7c2s/data$_preview$
[20] C. Yang, E. Gonzales, “Modeling Taxi Trip Demand by Time of Day in New York City”Trans Research Rec, Jul, 2014. DOI: 10.3141/2429-12.
[21] D. Correa, K. Xie, K. Ozbaay,“Exploring the Taxi and Uber Demands in New York City: An Empirical Analysis and Spatial Modeling”Trans Research Board Ann, Jan, 2017. DOI: 10.6084/m9.figshare.14503002.
[22] K. Zhannat, C. Choudhury, “Emerging Big Data Sources for Public Transport Planning: A Systematic Review on Current State of Art and Future Research Directions”Indian Inst Sci, vol. 99, no. 4, Sep, 2019. DOI: 10.1007/s41745-019-00125-9.
[23] NYC Open Data, 2024, "311 Service Requests from 2011 to present", Office of Technology and Innovation, New York. [Online]. Available: https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9/about$_data$
[24] NYC Open Data, 2024, "Points of Interest", Office of Technology and Innovation, New York. [Online]. Available: https://data.cityofnewyork.us/City-Government/Points-Of-Interest/rxuy-2muj
[25] L. Stice-Lawrence, “Practical issues to consider when working with big data,”Rev of Acc Studies, vol. 27, no. 3, Sep, 2022. DOI: 1117-1124.
[26] O. Sagi, L. Rokach, “Approximating XGBoost with an interpretable decision tree,”Information Sciences, vol. 572, Sep, 2021. DOI: 10.1016/j.ins.2021.05.055.
[27] W, Tong et al, “Decision Forest:Combining the Predictions of Multiple Independent Decision Tree Models,”J of Chem Inf and Comp Sci, vol. 43, no. 2, Feb, 2003. DOI:10.1021/ci020058s.
[28] C. Lung, "Measuring perceived usability: The CSUQ, SUS, and UMUX." chuniversiteit.nl. https://chuniversiteit.nl/papers/measuring-perceived-usability-the-csuq-sus-and-umux (accessed Jul. 25, 2024).

ADAM HERDMAN Adam Herdman was born in Dublin, Ireland in 1999 and achieved his Bachelor's degree in Creative Media Technologies in 2021, graduating from the Dún Laoghaire Institute of Art, Design and Technology. Having worked in primarily customer-facing roles, he has taken an interest in bridging the gap between user and service through UI design and infrastructure. Always interested in visuals and the art behind a project, he hopes to be able to assist in making visually pleasing and usable applications. He is currently studying a Master's degree in Computer Science at University College Dublin.

DILLON HENNESSY Dillon Hennessy was born in 2001 in Dublin, Ireland. He received his B.Sc. in Sports and Exercise Management in University College Dublin. He currently works as a performance analyst for UCD FC while pursuing his M.Sc. in Computer Science at University College Dublin. His role in ClearPath NYC was data lead.

MICHAEL MCKEON was born in Rotterdam, Netherlands in 1996. He received his B.Soc.Sci. in Social Science with joint majors in Economics and Social Policy and a specialisation in Human and Organisational Admin from University College Dublin. He received his M.Sc. in Management from University College Dublin before pursing Project Management in Dun Laoghaire Rathdown County Council. While there he received his Diploma in Project Management from the Institute of Public Administration. He is currently completing a M.Sc. in Computer Science at University College Dublin. His role in ClearPath NYC was backend lead.

ADAM SOUTHWICK Received his Bachelors degree in Economics as well as a BFA in theatre from New York University. He is currently pursuing a Masters degree in computer science at University College Dublin. His role in ClearPath NYC was maintenance lead.

NIAMH WALSH Niamh Walsh was born in Dublin, Ireland in 1993. She received the B.Ed. in primary education from University College of Ireland Maynooth, studying within the Froebel discipline. From 2016 to 2022, she was a primary teacher and became a State accredited Droichead induction mentor, along with piloting intercultural European Schoolnet education programmes. From 2022 - 2023, she was acting Head of Alexandra College's Hermione Residence, one of Ireland's premier boarding schools. Currently pursuing her M.Sc. in Computer Science at University College Dublin, Niamh's role in ClearPath NYC was coordination lead.

YUHAN HU Yuhan Hu received her B.Sc. degree in Psychology from Durham University, UK in 2023. From 2021 to 2022, she was a research assistant at Solitude Research Lab, Durham University. Yuhan is currently pursuing a M.Sc in Computer Science at University College Dublin. Her role in the ClearPath NYC Project is the frontend lead.