

Report Date: 11/11/2022

To: ematson@purdue.edu, ahsmith@purdue.edu and lee3450@purdue.edu

From: Chasing FOX

- Daye Kim (rlack4793@khu.ac.kr)
- Hyun Roh (yesyun@khu.ac.kr)
- Heewon Jeong (jhjmo0719h@chungbuk.ac.kr)
- Jihu Yang (zihooy@handong.ac.kr)
- Juwon Baek (21700340@handong.ac.kr)

Summary

- In the Data analysis part, five new urban factor datasets were found. After training the machine learning model on various factors, the results were compared. Through the model, two types of risk were combined into one risk score. The risk score was assigned to each node. The range of longitude and latitude was reset. The Chicago map was divided by a 1000 by 1000 matrix.
- In the Algorithm part, the risk score of each edge was calculated considering the (u, v) in each edge.
- In the Back-end part, in the K-SW server, the API that runs python code was integrated with apache2. And the API for creating users in the DB was updated.
- In the Front-end part, the registration page, profile page and scroll view function were implemented in conjunction with the server.

What Chasing FOX completed this week:

- **Data analysis**
 - Five new urban factor datasets have been found as follows.
 - Tree Debris Removal Requests [1]
 - Abandoned Vehicle Complaints [2]
 - Rodent Baiting Requests [3]
 - Tree Trim Requests [4]
 - Garbage Cart Requests [5]
 - The subject of the study was limited to pedestrians, and 20 types of crimes that could be dangerous to pedestrians were selected as follows.
 - 'THEFT', 'ASSAULT', 'BATTERY', 'NARCOTICS', 'WEAPONS VIOLATION', 'BURGLARY', 'ROBBERY', 'CRIM SEXUAL ASSAULT', 'CRIMINAL SEXUAL ASSAULT', 'PROSTITUTION', 'OFFENSE INVOLVING CHILDREN', 'SEX OFFENSE', 'STALKING', 'HOMICIDE', 'OBSCENITY', 'INTIMIDATION', 'KIDNAPPING', 'HUMAN TRAFFICKING', 'OTHER NARCOTIC VIOLATION', 'PUBLIC INDECENCY'
 - The results of machine learning models using only highly correlated urban factors for crime data were compared with those of models using all urban factors.
 - Using a machine learning model, urban factor risk and crime risk were combined into one risk score.
 - The calculated risk score was assigned to each node of the entire Chicago map graph.
 - The latitude and longitude range were reset as follows. (Zip Code 60666, which is the location for the airport, was extruded.).
 - longitude min = -87.9
 - longitude max = -87.5
 - longitude min = 41.6
 - longitude max = 42.05

- The Chicago map was divided by a 1000 by 1000 matrix and recalculated the 16 types of urban factor risk and the crime risk in both 2018 and 2019. (It was divided by a 100 by 100 matrix before.)

● Algorithm

- The riskiness of the edge was imputed by calculating the riskiness of the node connected to each edge. Each edge includes (u, v) which means starting at u and ending at v. So the riskiness of the edge was calculated with the mean of (u, v).

u	v	key	osmid	oneway	name	highway	length	lanes	bridge	tunnel	service	maxspeed	access	geometry	risk score_u	risk score_v	risk score
27446490	27446491	0	24308522	False	East Monroe Drive	secondary	17.036	NaN	NaN	NaN	NaN	NaN	NaN	LINESTRING (-87.62445 41.88083, -87.62424 41.8...	5	5	5.0
	2787305400	0	416548018	False	South Michigan Avenue	primary	10.679	3	NaN	NaN	NaN	NaN	NaN	LINESTRING (-87.62424 41.88093, -87.62424 41.8...	2	5	3.5
	2787305669	0	163148295	False	South Michigan Avenue	primary	11.835	3	NaN	NaN	NaN	NaN	NaN	LINESTRING (-87.62425 41.88073, -87.62424 41.8...	3	5	4.0
	2787305670	0	721925648	False	East Monroe Drive	secondary	9.036	NaN	NaN	NaN	NaN	NaN	NaN	LINESTRING (-87.62414 41.88084, -87.62424 41.8...	4	5	4.5
27446491	27446490	0	24308522	False	East Monroe Drive	secondary	17.036	NaN	NaN	NaN	NaN	NaN	NaN	LINESTRING (-87.62424 41.88084, -87.62445 41.8...	5	5	5.0

Fig 1. The edges including risk score

- For the implementation of the navigation, 9 nodes were selected to be used in the Google Map API. By considering the total length of the route, 9 nodes were selected with similar intervals.

● Back-end

- All the APIs for the application were organized. Since the server for Python and Java was different, Flask and Spring Boot were both used for developing APIs.
- The APIs that are related to handle user accounts are related to Java, so it was developed by Spring Boot. The APIs related to setting the riskiness of the road are developed with Python, so the main tool was Flask.
- In the K-SW server, the API made with Flask that runs python code was integrated with apache2 [6]. The port number is 80, and this test API can be connected at <http://192.168.2.117:80/api/test>.
- About the Spring server, the API for creating users in the DB was updated reflecting the modification in user objects. The port number is 8080, and this test API can be connected at <http://192.168.2.117:8080/api/user>.
- In the K-SW server, the additionally modified Spring code and Flask code were distributed. If the K-SW server is powered on all day, these API can be connected 24 hours.

● Front-end

- Since Google and Kakao Talk login functions were excluded, the UX page of the login page was modified and implemented.
- The registration page has been implemented. And when users signed up, they used Axios to store user information on the 'http://192.168.2.117:8080/api/user' server.
- The profile page was implemented, and information was retrieved from the server and stored in Encrypted storage.
- The scroll view function was implemented. It is used when the height of the application screen exceeds the size of the cell phone.

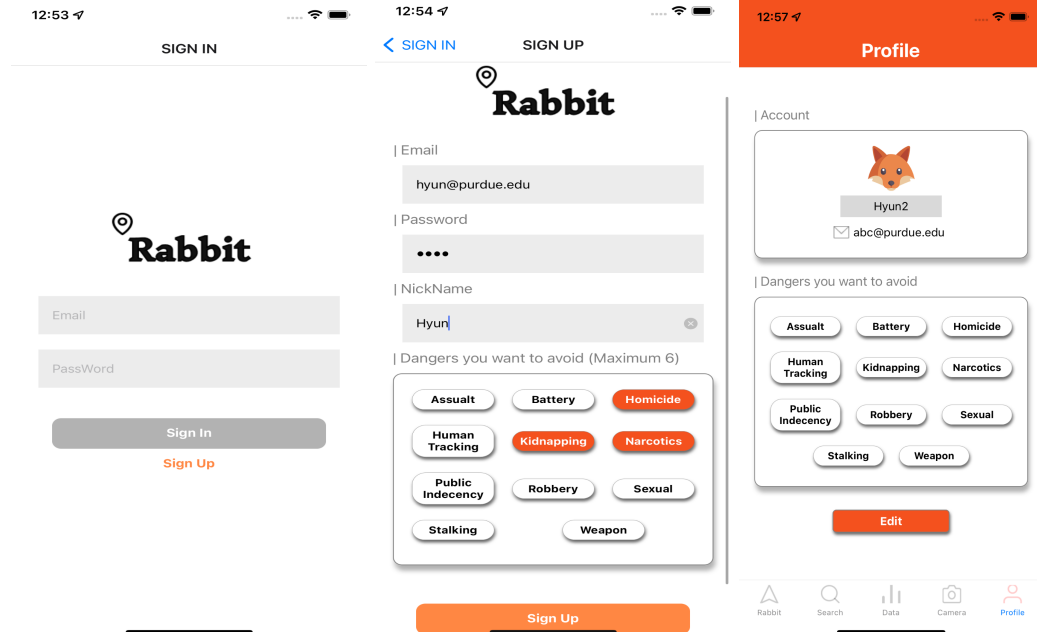


Fig 2. The Sign-In page

Fig 3. The Sign-Up page

Fig 4. The Profile page

Things to do by next week

- In the Back-end part, the function of the API that runs python code will be added to run the code about extracting each user's personalized nodes and edges and 9 way points.
- In the Front-end part, the path recommendation page will be implemented in connection with the server.
- In the Front-end part, the data Analysis page will be implemented. The task of obtaining the results of the data analysis from the server will be performed.
- The data analysis results from the 1000 by 1000 Chicago matrix will be summarized.
- The method of combining the risk score and distance of each node, the equation of the weight function, will be considered.
- The evaluation and verification method of the algorithm will be considered.

Problems or challenges:

- One of the problems is whether user information should be stored on the server or local storage. This is a security issue related to the token value, so it is being discussed in the Front-end and Back-end parts.
- As a method for combining risk score and distance, it is necessary to consider whether to use the weight function or to block only high-risk areas.

References

- [1] Chicago Data Portal, *311 Service Requests - Tree Debris - Historical*, (Mar 7, 2019). [Online]. Available: <https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Tree-Debris-Historical/mab8-y9h3>. [Accessed: Nov 9, 2022].
- [2] Chicago Data Portal, *311 Service Requests - Abandoned Vehicles - Historical*, (April 21, 2022). [Online]. Available:

<https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Abandoned-Vehicles-Historical/3c9v-pnva>. [Accessed: Nov 9, 2022].

[3] Chicago Data Portal, *311 Service Requests - Rodent Baiting - Historical*, (Mar 7, 2019). [Online]. Available:

<https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Rodent-Baiting-Historical/97t6-zrh>s. [Accessed: Nov 9, 2022].

[4] Chicago Data Portal, *311 Service Requests - Tree Trims - Historical*, (Mar 7, 2019). [Online]. Available:

<https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Tree-Trims-Historical/uxic-zsuj>. [Accessed: Nov 9, 2022].

[5] Chicago Data Portal, *311 Service Requests - Garbage Cart - Historical*, (Mar 7, 2019). [Online]. Available:

<https://data.cityofchicago.org/Service-Requests/311-Service-Requests-Garbage-Carts-Historical/9ksk-na4q>. [Accessed: Nov 9, 2022].

[6] “mod_wsgi (Apache).” Pallets. https://flask.palletsprojects.com/en/2.0.x/deploying/mod_wsgi/ [Accessed: Nov 10, 2022].