Prediction of the Crowdness in Subway Platform

Won-Ryeol Kwak. College of Liberal Studies,
Seoul National University, Seoul, Korea
Email: wrk9224@snu.ac.kr

***Abstract*—In this paper, we define the crowdness using the number of passengers getting on and off in subway platform. Then, we work with MLP, RF to forecast the crowdness.**

I. Introduction

Seoul, the capital of South Korea, is the most crowded city in this country. So, there exist highly developed public transport system carrying passengers to anywhere in Seoul. Especially, a lot of people take Seoul metro for daily routine such as commuting, meeting people. Considering only line 1 $\sim$ 8 and 2,3 section of line 9, which is managed by Seoul Transportation Corporation, there exist over 2.7 billion passengers who took Seoul metro in 2019. Among them, the majority of passenger data is recorded at rush hour, on transit station. Of course, such data directly reflects the crowdness in the station.
Therefore, it is necessary to predict the crowdness on platform and forecast it at right time. In this paper, we use machine learning to predict the crowdness in specific station/time from past passenger data.

II. Related Works

Professor Kim. suggested the definition of conjestion(crowdness) in [1].

III. Problem Definition

For simplification, We take only 10 stations of Seoul metro line 1 for samples According to [1], we define the crowdness in the platform as the following.

$$Crowdness = \frac{passengers}{capacity} \tag{1}$$

Then, we are going to estimate average number of passengers and predict the platform capacity to find crowdness.
First, average number of passengers is product of the number of people per hour and the time they stayed in the platform.

$$\begin{aligned} &\text{average number of passengers per hour} = \\ &K \times \text{number of people per hour} \times \text{platform staying time} \end{aligned} \tag{2}$$

(where K is constant.)

Then, we split the passengers into riding on passengers(RP) and getting off passengers(GP). Generally, GPs get out of the platform right after they getting off. Unlike GPs, RPs first wait for the next train before riding on. Assume that the number of waiting people in platform linearly increases before the next train arrives. By this assumption, average waiting time for RPs is half of the dispatch interval of train. So, the platform staying time is estimated to following.

$$t_{RP} = \frac{\text{dispatch interval}}{2} + \text{platform entering time} \tag{3}$$

$$t_{GP} = \text{platform entering time} \tag{4}$$

time(24hours)	5	6	7	8	9	10	11	12	13	14
weekday	8	11	17	18	15	15	12	12	12	12
Sat/Sun	8	9	13	13	15	14	13	11	13	12
time(24hours)	15	16	17	18	19	20	21	22	23	24
weekday	13	12	15	16	15	16	12	12	12	5
Sat/Sun	13	11	14	15	15	15	13	12	11	1

TABLE I
The number of train arrival per hour in Seoul Station.

Considering the timetable of Seoul Metro line 1, provided by Seoul Transportation Corporation, about 15 trains go through the platform every hours, so average $dispatch\ interval$ is 4 minutes. Plus, it takes about 1 minute to get out of/come into the platform. Therefore, $t_{RP} = 3$ minutes, $t_{GP} = 1$ minute. For some constant k,

$$\frac{Average\ passengers}{hour} = \frac{0.75RP}{hour} + \frac{0.25GP}{hour} \tag{5}$$

Second, platform capacity generally depends on the area of platform. But in most case, people in platform are not uniformly scattered throughout the platform. Instead, they stand in line near the screen doors. So, we count up the number of standing lines. According to Seoul Transportation Corporation, the train of line 1 is consist of 10 cars, which has 4 doors for each cars. In the platform, there exist 2 standing lines right and left of the door. In the result, the number of whole standing line of platform is 80. We call 100% crowdness when there are 4 people for all 80 standing lines. By the timetable above, 15 trains go through the station, we can say that the platform capacity is 4800 people per hour. To sum up,

$$Crowdness = \frac{passengers}{capacity} = \frac{0.75\frac{RP}{hour} + 0.25\frac{GP}{hour}}{4800} \tag{6}$$

IV. Solving Approach

We divide this crowdness into 10 level like following, then make levels to be a target(output) by one hot coding.

Inputs are month, weekdays(W)/Saterday(Sat)/Sunday(Sun), station number, and time. Encoding method for each inputs are like following.

Level	Lv.1	Lv.2	Lv.3	Lv.4	Lv.5
Crowdness	0∼ 0.1	0.1∼ 0.2	0.2∼ 0.3	0.3∼ 0.4	0.4∼ 0.5
Level	Lv.6	Lv.7	Lv.8	Lv.9	Lv.10
Crowdness	0.5∼ 0.6	0.6∼ 0.7	0.7∼ 0.8	0.8∼ 0.9	0.9∼

TABLE II
MATCHING OF LEVEL AND CROWDNESS

Input 1 : month(1∼ 12)
Input 2 : W(3), Sat(5), Sun(7)
Input 3 : station number
Seoul Station(1) - City Hall(2) - Jonggak(3) - Jongno 3ga(4) - Jongno 5ga(5) - Dongdaemun(6) - Dongmyo(7) - Sinseoldong(8) - Jegidong(9) - Cheongnyangi(10)
Input 4 : time(5∼ 24) to $(0.5 \sim 10.0)$ by
$t = 2x + 4$, where t is time, x is encoded time.

(time n means the interval between the n and n+1 o' clock)

We obtained about 0.9million data of passenger ride on, get off between from 2008 to 2019. First we preprocessed the data by the encoding method above, then let 2008 $\sim$ 2018 data to training data and 2019 data to test data. We trained 2-layer perceptron and random forest by the data. We used 2000 dimensional hidden-layer, 0.1 learning rate, 10 epoch, and initial weight randomizing by seed 0 for 2-LP, 100 trees for random forest.

V. EVALUATION

After training, we try to predict the data of 2019 using the model trained by past data.

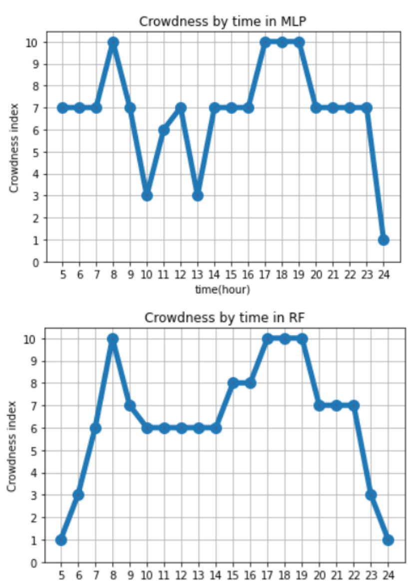

Fig. 1. Prediction of crowdness on June, weekdays, Seoul Station.

By the plot above, we can find the feature of crowdness that it reached peak at rush hours.
After that, we measure the correctness of target and prediction calculated by the whole inputs of test set. Target and prediction can be matched tally, in error range 1, or neither. For each cases we state that it is critically success, nearly success, failure. Results are as follows.

```
MLP error rate
success rate =  0.7487397260273972
failure rate =  0.25126027397260275
Critically success rate= 0.4629041095890411
nearly success rate= 0.28583561643835614
Total  18342 Error cases
```

```
RF error rate
success rate =  0.9478493150684931
failure rate =  0.05215068493150685
Critically success rate= 0.6658904109589041
nearly success rate= 0.281958904109589
Total  3807 Error cases
```

For 73,000 test data, MLP model hits 46.3% critically success, 74.9% success. RF model hits 66.6% critically success, 94.8% success. Consider the ambiguity of self-defined crowdness itself, we are going to look up the difference between target and prediction.

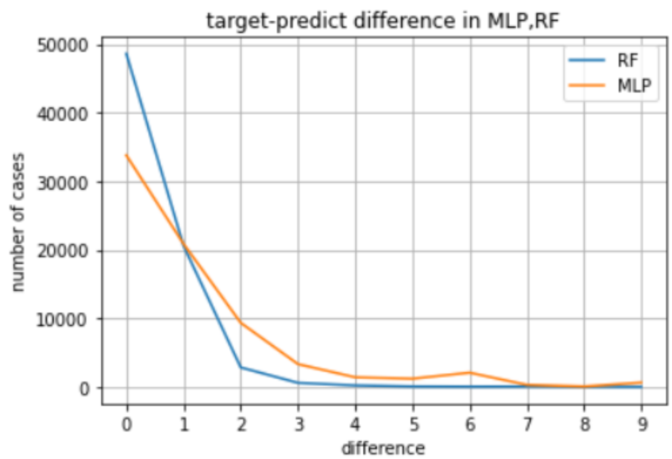

Fig. 2. Target-Prediction difference of 73,000 samples.

There exist 87.7% samples in error range 2 by MLP model. On the other side, there exist 98.7% samples in error range 2 by RF model. So, it can be make sense that the RF model successfully predict the crowdness.

VI. CONCLUSION

We trained MLP and RF by past passenger data and could predict the future platform crowdness along time. Also, we find that RF model could forecast the crowdness more accurately. Using this model, people can get crowdness forecast, handle with the conjestion problem wisely just like avoiding the crowding subway station. At the same time, we can put not only 3,5,7 but also 1,9 into input2 in the MLP model. Through this, perceptron can also predict the situation such as highly restful holiday or quiet and quiet platform by epidemic prevalence of COVID-19.
If further research make up, consideration of the up/down line and extension to line 2∼ 9 would be appropriate.

REFERENCES

[1] J.-s. Kim, "Subway congestion prediction and recommendation system using big data analysis," *Journal of digital Convergence*, vol. 14, no. 11, pp. 289–295, 2016.