# A Matching Model for Vehicle Sharing Based on User Characteristics and Tolerated-Time

**Govind P. Yatnalkar and Husnu S. Narman**
College of Information Technology and Engineering, Marshall University, Huntington, WV, USA

{yatnalkar}{narman}@marshall.edu

***Abstract*—In the present age, transportation is humankind's necessity. With the increasing population, it has produced adverse effects like rapid consumption of fuel resources, high carbon emissions, and global traffic issue. In such cases, vehicle sharing is gaining attraction as a possible candidate solution. We have implemented a vehicle sharing rider matching model which matches users reaching nearby destinations. The algorithm then undergoes another matching layer, which filters users based on user characteristics. Best-matched users are then added to a final itinerary forming the route for the commute. In our model, we have used New York City cab zone locations with real-time navigation using Google Maps. We have introduced the concept of "User Threshold Time (UTT)," the time riders are willing to spend to pick other riders. Our major motive is to complete the pool for the maximum number of trips based on user characteristics. On a global scale, our model aims at saving resources and improve overall global atmospheric conditions. Results show that our matching model can be achievable in a reasonable time constraint.**

***Index Terms*—vehicle sharing, carpooling, ride-sharing, characteristics, user feedback system, user threshold time**

## I. Introduction

Vehicle sharing previewed itself during World War II oil and energy crisis [1, 2]. With time, world conditions improved and people got financially stable, resulting in a downfall of ride-sharing. Indeed, automobiles provide many benefits, but they also provide myriad problems. In the past decade, there has been an immense growth in the overall number of vehicles which has directly impacted the present traffic conditions [3]. Solutions like High occupancy vehicle (HOV) lanes are proposed to address the traffic issue, but there has not been a significant improvement in current traffic scenarios [4].

Moreover, fuel consumption has increased exponentially, and eventually, there is a possibility of outrunning these natural resources [5]. Despite the government's efforts in renewable energy generation, the rate of fuel consumption is comparatively high than renewable energy consumption [6]. The byproduct, vehicle emissions has detrimental effects on the environment and human health [7].

In such cases, vehicle sharing is a possible solution. It is the process of ride-sharing among riders traversing a series of sources and destinations. Moreover, carpooling increases the number of HOV lanes, providing smoother transportation. Reduced vehicle count means reduced fuel consumption. However, ride-sharing with strangers can be a problem if a central controlling system is absent as seen at airports. Therefore, our *aim* is to design a model than matches users based on their characteristics which result in a joyful and stress-free ride.

Our *objective* is to consider five characteristics and tolerated threshold time while designing the matching model to increase rider count. Initially, we register users with characteristics like chatty, friendliness, safety, punctuality, and comfortability. We perform matching of users having similar or closer characteristics. In registration, the User Threshold Time (UTT) is taken. This is the possibility of maximum waiting time that the rider and driver agree on. UTT times are between 10 and 30 minutes. Based on minimal UTT of a rider in a trip, drivers pick other riders to respect tolerated time of other riders. If the time required to pick up other rider exceeds trip UTT, the commuter is not picked. Threshold time assures travelers do not wait long picking other riders during a journey.

The key *contributions* of this paper are as follows: (i) a carpooling system is created based on characteristics of users, (ii) a model that considers matching using tolerated time; and (iii) an extensive simulation to test the efficiency of the model using real-time data.

Results show that it is possible to allocate best-matched riders using characteristics and UTT. Our proposed model aims to increase vehicle sharing while respecting rider considerations and decrease consumer frustration.

The rest of the paper is organized as follows. In Section II, the existing techniques are explained. In Section III, the feedback-based carpooling model is discussed. In Section IV, the proposed model is explained in detail. In Section V, the simulation environment with obtained results are discussed. Finally, Section VI has the concluding remarks and our plans to extend the proposed work.

## II. Literature Survey

With good availability of the internet and advanced technologies, carpooling has observed immense development. Companies like Uber and Lyft are coming up with ideas to enhance vehicle sharing[8]. However, social barriers and lack of equipment discourage carpooling. State governments are proposing plans to reduce taxes on ride-sharing vehicles and use public transport with ride-sharing services, but the overall market for vehicle sharing remains low [4, 9].

### A. *Popular Commercial Applications*

We started our research with Lyft, Uber, Juno, and Waze [10]. States like New York, California, Florida, and Texas are most popular for these cab services [11]. California is home to many car-sharing companies; hence, vehicle sharing is used heavily in California. The New York City Cab [12] is working with Uber contributing notably to ride-sharing services. While researching this case, we came across a data repository which included real-time NYC taxi zone locations [13], later forming the data source for our simulation.

Our findings directed to several issues. Passengers do not possess any knowledge of other riders. Drivers learn the passenger count after reaching the pick-up location. Such events prove stressful to all commuters. A vital issue found was the unexpected longing of the journey due to the sudden addition of a rider resulting in disputes and distress [14]. Another critical issue is the model design, "Same-Source-Same-Destination" and "Many-Sources-One-Destination" [15] approach which does not meet rider expectations.

Noting the stated issues, we provide trip data to all riders at the end of every trip formation. We tend to get best-matched riders using their characteristics. Also, rider waiting time or UTT never surpasses the registered UTT.

### B. Modern Technologies with Vehicle Sharing

Internet of Things and Cloud Computing are speeding up the building of smart cities. Indeed, car-sharing is a part of such smart systems.

IoT allows efficient device connectivity and communication for data broadcasting. A published notification can be sent to a billion connected devices. With carpooling, every vehicle can be connected to a data hub logging every minor update [3, 16]. Vehicle status can be notified to broadcasting riders, facilitating faster decisions for road traversing, vehicle tracking, and location-based requests clustering. These features can be used for frequent status updations, quicker rider-driver associations, and faster trip formation.

Cloud services bring numerous benefits to any computing system [17, 18]. Enabling cloud services enables better scalability, availability, plus efficient load management [17]. Using the cloud decreases the overall costs of any system. Albeit, time decides the fate of an application. In the cloud environment, requests from a client device travells and interact with servers and travel back to client devices to render server data introducing a latency. To reduce this delay, we learned about Fog Computing. A small group of servers is placed near the client location. Computations take place at this small cloud reducing the travel time. [19]. We utilized this idea of Fog computing for our carpooling technology. Currently, the computation is processed at the client machine.

To conclude, modern technologies play a crucial role in application design and resource management. Also, factors like load balancing, timeliness of result, user experience, and quality of service are equally vital.

### C. Multiple Sources Multiple Destinations (MSMD)

MSMD includes same source-same destination, same source-different destinations, different sources-same destination, and the most vital, different sources-different destination. It utilizes models like star networks, Dijiktras many sources one destination problem, and greedy algorithms [9, 20, 21].

One methodology stated the formation of multiple routes using star networks until finding an optimized one. The drawback is the computation time for developing multiple routes until finding the best route [9]. A similar approach is completing the journey through different transportation systems [22] like buses, bikes [4, 23] or even simply walking. Indeed, this again adds up extra time in the entire journey but follows the model of multiple sources and multiple destinations.

## III. System Model

The system model reflects an entire framework of a system. The heart of our designed algorithm is the procedure utilized for matching between riders.

### A. Problem Statement

Post scrutiny of many articles and reference papers, we found the major issue lies is in the matching of riders and time management [9, 18, 21]. Vehicle sharing can be encouraged if there are good matching rates and trip formation time. Also, users should be provided with meta-data of the trip. Indeed user locations and sensitive information are encapsulated for security purposes. Also, user expectations are met using multiple sources and multiple destinations model, which is an excellent choice for carpooling with time management.

### B. Architecture

Vehicle sharing model starts with an association of a driver to a trip, followed by finding and filtering riders based on characteristics and UTT. The model's last step is saving the rider feedback. Figure 1 provides an architecture of our implemented matching model.

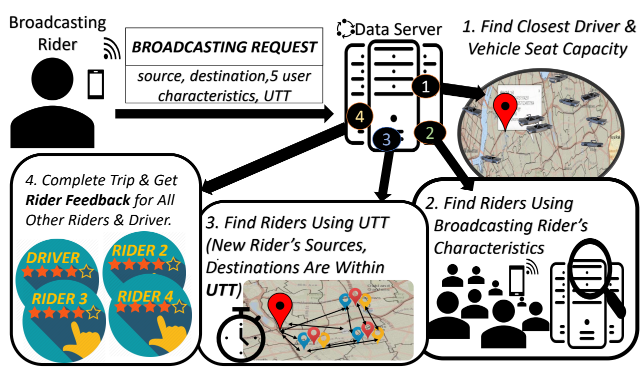


Fig. 1. Architecture for vehicle sharing rider matching model.

Throughout the implementation, we have maintained a client-server environment. Initially, a user broadcasts a rider request which includes user-id, source, and destination. Based on the user-id, we retrieve the characteristics and UTT. This data document forms the first stage of the trip. At the server-side, there is an active repository of all drivers. When drivers are active or awaiting broadcasting requests, their location and status are continuously updated for faster allotment to incoming requests. For an incoming request, all available drivers from the request originating zone are retrieved. The closest driver to user source location is selected. This adds the next vital data document of the driver in the trip.

The source zone is sent as a parameter to the rider matching functions. The first function retrieves best, or close characteristics matched rider list. The second function checks the traveling time of every rider's source and destination to broadcasting rider's source and destination. If the traveling time is less than trip UTT, the rider is accepted. These functions execute until the seating capacity of the vehicle is reached or until there are no riders in the rider list.

At epilogue, all riders rate the driver and other riders. The feedback system is a novel design to improve the matching rate. While rating, a rider selects a rating number for five characteristics on a scale of 1 to 5.

## IV. Proposed Model

The model inchoates five stages: The broadcasting rider, the closest driver, finding riders by characteristics and UTT matching, and saving user feedback. A brief description of every step is summarized in the following sub-sections.

### *A. The Broadcasting Rider*

The algorithm begins through a broadcasting rider which includes the broadcasting source zone and location forming starting point for the trip. Destination zone, location, five characteristics are also recorded. The source zone is referred for finding the closest available driver.

### *B. The Closest Driver*

Using the broadcasting rider source zone, a list of available drivers is retrieved. A driver is added to the list if the driver is active but the commuting vehicle has not reached seating capacity. The traveling time between the driver's current location and broadcasting rider's location are checked using Google Map Distance Matrix API. The driver with the shortest traveling time is selected and added in the trip.

### *C. Searching Riders with Characteristics Matching*

The trip data now consists of the broadcasting rider locations, five characteristics, and the closest driver. Other broadcasting riders are searched with similar or closer characteristics. Based on our several simulations, we concluded that the odds of finding broadcasting riders with similar characteristics is low. If riders are found, we add them in matched rider queue.

If the seating capacity is not reached more riders are searched by altering each characteristic. For example, if the characteristics are chatty:4, safety:3, punctuality:3, friendliness:1, comfortability:2, an alteration is done by adding or subtracting 1 to the chatty score resulting in either 5 or 3. This is defined as "closer" characteristic matching. Rider search continues until the seating capacity of the vehicle is reached. If the pool is still incomplete, all the broadcasting riders from source zone are selected and added in the list. This model is a default search model for Uber and Lyft. The rider list formed in this phase is given as the input in the next phase, matching with UTT.

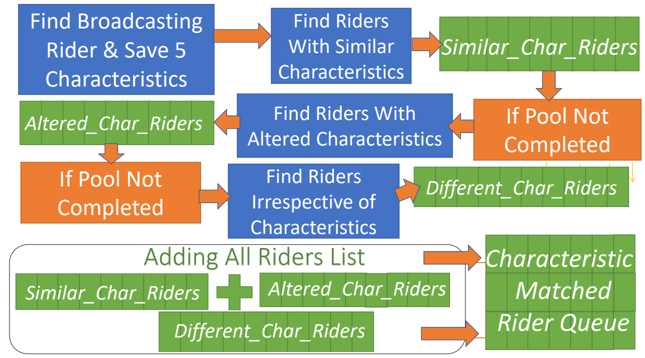


Fig. 2. Finding riders using characteristics matching. The output list or the riders queue is used as input for the next matching layer, matching using UTT.

### *D. Searching Riders with UTT Matching*

The broadcasting rider's UTT is referred to as the trip-UTT. For every rider in the list, a source and destination are selected from the NYC Zone file. Mostly, the source location originates from the same zone. Using the Google Map Distance Matrix API, the traveling time is calculated between the rider's source and broadcasting rider's source location. If the time is less than or equal to UTT, the algorithm proceeds to the second UTT check. The second UTT check includes calculating the traveling time between the rider's destination and broadcasting rider's destination and verifying if the time is less than or equal to UTT. If both UTT checks are satisfactory, the rider is added in the trip document. The UTT check is done for all riders in the queue until it reaches the seating capacity.

### *E. Final Trip Document*

The final trip document saves every rider's source, destination, characteristics, UTT, driver details, and vehicle seat capacity. Moreover, the overall time required for the journey is noted. The trip document is the final step and is added to the mongo trip collection.

## V. Simulation and Results

### *A. Experimentation*

At first, we selected a broadcasting rider with UTT 10 and a rider count of 100. The rider search begins based on characteristics and then by UTT. If a match occurs, riders are added in the trip else next rider is searched for matching. The traversing of riders continues till the rider count of 100 is reached. We ran the same code by increasing the UTT by 5 until it reached 30. Then, we increased the rider count by 100 until it reached 500. The complete simulation sequence can be given by (10,100), (15,100), (20,100), (25,100),(30,100)... (10,1000), (15,1000), (20,1000), (25,1000), (30,1000). The first digit denotes the UTT and the second denotes the rider count. We ran every simulation ten times.

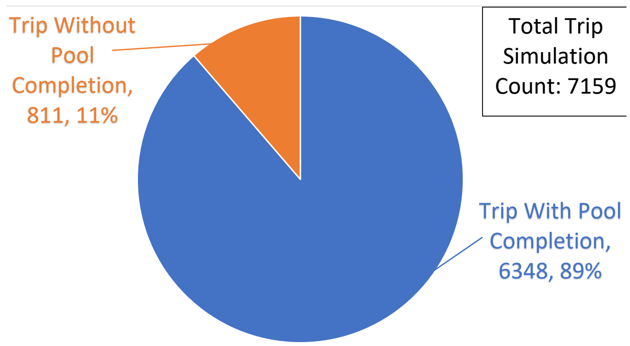


Fig. 3. Total trip count and percentage based on pool completion status.

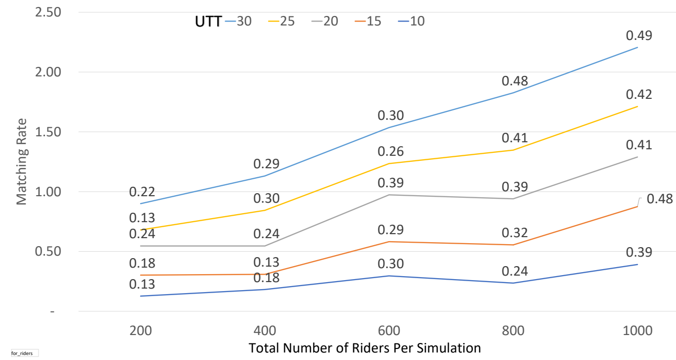


Fig. 5. Average matching rate per simulation event.

## B. Observations

The total trip count is 7159. Average trip formation time is 0.80 minutes which is less than a minute. Figure 3 states 6348 trips completed the pool, and 811 trips did not. From this analysis, our motive for pool completion for maximum trips is achieved. Also, total rider count checked in the complete simulation is 276400, out of which 93766 riders are in the pool. Figure 4 depicts the classification of accepted riders by exact or close characteristics match and alternative characteristics match.

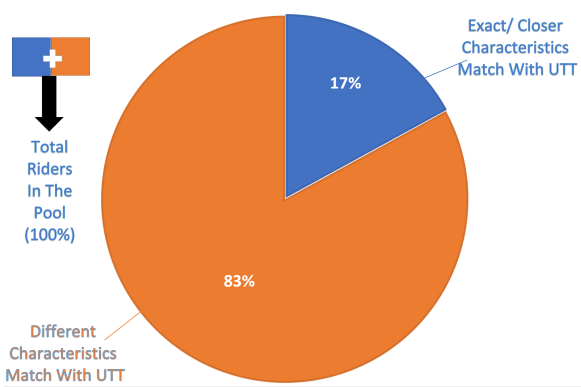


Fig. 4. Riders in the pool classified by the type of matching.

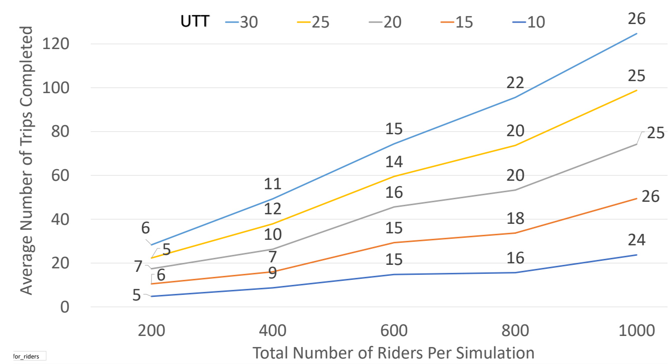


Fig. 6. The average number of trips completed per simulation.

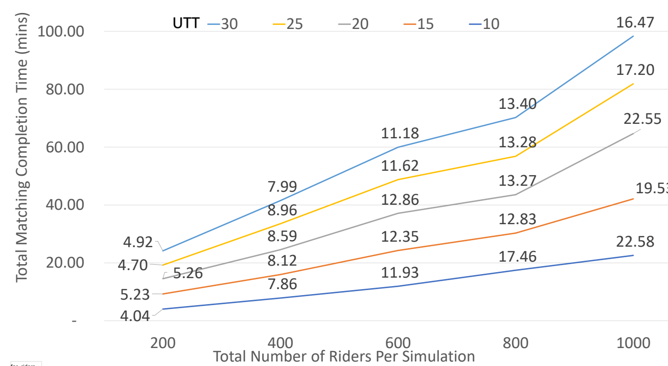


Fig. 7. The average time consumed for every simulation event.

Matching rate is the number of riders in the pool divided by the total rider count. If 10 riders are searched and 5 are accepted, the matching rate is 0.5. During the analysis, we drafted the matching rate, the average number of trips completed, and the average trip formation time. Figure 5 reflects as the rider count and UTT increases, the matching rate increases. As UTT and rider count increase, more riders are accepted at a faster rate, and more trips are completed. Therefore, as drafted in Figure 6, there is more room for riders which increases the total number of trips. Figure 7 states that the increased rider count correspondingly increases trip simulation time. In the end, we concluded, increase in riders, and UTT increases the matching rate, total trip count, and total trip simulation time.

## VI. Conclusion and Future Work

In this paper, we proposed a vehicle sharing matching model using user characteristics and User Threshold Time (UTT). To show the efficiency, an extensive simulation is developed. The performance is evaluated from 100 to 1000 riders with UTT from 10 to 30 minutes. It is observed as the rider count and UTT increase, the matching rate increases correspondingly. We also achieved the goal of maximum trip execution with pool completion. Also, the average trip formation time is less than a minute, which contributes to the quality of service and improved user experience.

Our future implementations include tracing the patterns in the rider feedback system using the machine learning algorithms. Matching will depend on the feedback score the rider focuses the most rating other riders. Also, an Android application with a pricing model may be developed for handling the transaction of requests for riders and drivers.

## REFERENCES


[1] T. Z. Jaime Netzer. (2014) A brief (illustrated) history of carpooling. [Online]. Available: https://www.thezebra.com/insurance-news/406/history-of-carpooling/

[2] L. Cook, "The energy-crisis-going, going, but not quite gone," *IEEE Transactions on Industry Applications*, vol. 17, no. 6, pp. 548–551, 1981.

[3] X. Wang, Z. Ning, X. Hu, L. Wang, B. Hu, J. Cheng, and V. C. Leung, "Optimizing content dissemination for real-time traffic management in large-scale internet of vehicle systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1093–1105, 2018.

[4] H. Huang, D. Bucher, J. Kissling, R. Weibel, and M. Raubal, "Multimodal route planning with public transport and carpooling," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[5] J. L. Helm, *Energy: Production, Consumption, and Consequences.* ERIC, 1990.

[6] C. T. Tugcu, I. Ozturk, and A. Aslan, "Renewable and non-renewable energy consumption and economic growth relationship revisited: evidence from g7 countries," *Energy economics*, vol. 34, no. 6, pp. 1942–1950, 2012.

[7] A. do Nascimento Rocha, L. S. Candido, J. G. Pereira, C. A. M. Silva, S. V. da Silva, and R. M. Mussury, "Evaluation of vehicular pollution using the trad-mcn mutagenic bioassay with tradescantia pallida (commelinaceae)," *Environmental pollution*, vol. 240, pp. 440–447, 2018.

[8] S. Li, F. Fei, D. Ruihan, S. Yu, and W. Dou, "A dynamic pricing method for carpooling service based on coalitional game analysis," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS).* IEEE, 2016, pp. 78–85.

[9] J. Jamal, A. Rizzoli, R. Montemanni, and D. Huber, "Tour planning and ride matching for an urban social carpooling service," in *MATEC Web of Conferences*, vol. 81. EDP Sciences, 2016, p. 04010.

[10] M. Tang, S. Ow, W. Chen, Y. Cao, K.-w. Lye, and Y. Pan, "The data and science behind grabshare carpooling," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA).* IEEE, 2017, pp. 405–411.

[11] WTOP. (2019) Top carpool percentages by city. [Online]. Available: https://wtop.com/council-of-governments-guaranteed-ride-home/2019/04/top-carpool-percentages-by-city/

[12] M. Samy and A. M. Elkorany, "Using semantic features for enhancing car pooling system," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).* IEEE, 2018, pp. 702–708.

[13] NYC, "Nyc open data," 2019. [Online]. Available: https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc

[14] J. Koebler, "Why everyone hates uberpool," May 2016. [Online]. Available: https\:\/\/www.vice.com\/en_us/article/4xaa5d/why\-drivers\-and\-riders\-hate\-uberpool\-and\-lyft\-line

[15] C. Huang, D. Zhang, Y.-W. Si, and S. C. Leung, "Tabu search for the real-world carpooling problem," *Journal of Combinatorial Optimization*, vol. 32, no. 2, pp. 492–512, 2016.

[16] I. Grgurevic, D. Perakovic, I. Forenbacher, and T. Milinović, *Application of the Internet of Things Concept in Carsharing System*, 10 2015, pp. 401–414.

[17] M. Zhu, X.-Y. Liu, and X. Wang, "An online ride-sharing path-planning strategy for public vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–12, 2018.

[18] C.-H. Lin, M.-K. Jiau, and S.-C. Huang, "A cloud computing framework for real-time carpooling services," in *2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012).* IEEE, 2012, pp. 266–271.

[19] M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing," *IEEE Internet of Things Journal*, 2018.

[20] R. Hasan, A. H. Bhatti, M. S. Hayat, H. M. Gebreyohannes, S. I. Ali, and A. J. Syed, "Smart peer car pooling system," in *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC).* IEEE, 2016, pp. 1–6.

[21] Y. Duan, T. Mosharraf, J. Wu, and H. Zheng, "Optimizing carpool scheduling algorithm through partition merging," in *2018 IEEE International Conference on Communications (ICC).* IEEE, 2018, pp. 1–6.

[22] V. M. de Lira, R. Perego, C. Renso, S. Rinzivillo, and V. C. Times, "Boosting ride sharing with alternative destinations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2290–2300, 2018.

[23] K. Kalogirou, N. Dimokas, M. Tsami, and D. Kehagias, "Smart mobility combining public transport with carpooling: An ios application paradigm," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS).* IEEE, 2018, pp. 1271–1278.