IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

FIDDLE TOUR: FRAUDULENT TAXI TRIP DETECTION USING KNN MACHINE LEARNING ALGORITHM

[1]S.P. Ramya, [2]U. Ushekha, [3]R. Muthukkumar

[1,2,3] Dept. of Information Technology,

National Engineering College, Kovilpatti 628 503, Tamil Nadu, India.

Abstract: Taxi service is a very important part of public transportation in advanced cities, providing convenience for our lifestyle. Taxi services in trendy cities' area units are typically corrupted by fraud, and passenger area units are typically overcharged by taxi drivers. Existing trip detection models works believe the idea that the trip is properly recorded by the meter. However, there is a unit of several taxi drivers in Asian nations carrying passengers while not activating the meter, particularly once the taxi driver is attempting to overcharge the passengers. Thus, the present system predicts the unmetered taxi trips area unit detected in real-world situations, which describes the taxi trip that has been recorded as vacant but has similar driving behaviours to regular metered trips. It consists of a learning model that predicts the occupancy standing of taxis, but the prediction level is very low, and it is not correct. In this paper, K-Nearest Neighbour (KNN) machine learning algorithmic rule is proposed to determine tax fraud. Taxi fraud is determined by the cost per kilometre, if the driver overcharges the passenger the model predicts the fraud. In this model, first, the dataset has been trained for fraud detection. Second, the cost for the taxi trip is calculated based on the one-way, round trip, and including waiting time. Experimental results reveal that the proposed model detects taxi driver fraud within the calculation of trip sheets and enhances accuracy in identifying overcharging in fraud detection.

***Index Terms* - K-Nearest Neighbour (KNN), Machine learning, Trajectory, Distance, Price.**

I. INTRODUCTION

Taxi is major transportation within the populated area, giving nice advantages and convenience to our standard of living. However, one of the main business frauds in taxis is charging fraud, specifically overcharging for a particular distance. In application, it's hard to perpetually monitor taxis and find such fraud. Because of the worldwide global positioning system (GPS) embedded in taxis, we will collect the GPS reports from the taxis' locations. The data is used to construct taxis' trajectories and work out the service distance on the town map and find dishonest behaviors. However, in the application, because of the very restricted reports, notable location errors, advanced town maps, and road networks, our task to find taxi fraud occurred to passengers that they face important challenges, and the previous ways cannot work well. Taxi services in trendy cities square measure typically corrupted by frauds, and passengers square measure typically overcharged by taxi drivers [1].

Common taxi frauds include 1) meter meddling, wherever the meter of a taxi is changed so that the shown driving distance is longer than the true driving distance; 2) detour, wherever the taxi takes associate degree irregular route to deliver a passenger and sometimes takes a lot of driving time and distance than usual; and 3) refusing of service, wherever the motive force refuses to hold a selected passenger or tries to seek out a passenger World Health Organization (WHO) is willing to pay a lot of fares. These dishonorable behaviors sometimes have evident properties that take issue with traditional taxi journeys. For instance, the space of a detour is typically longer than traditional ways between a combination of supply and destination [2]; the rumored speed of the taxi with a tampered meter tends to be over its actual speed [3]; and the taxi drivers with a better financial gain square measure a lot of possible to refuse passengers traveling to less-traveled areas [4, 5]. There are several existing strategies that attempt to find these forms of behaviors.

This proposed model aims to find taxi fraud through unmetered taxi passages as bearing passengers while not driving the measure. During this approach, taxi drivers may " carry out" and now and besides overcharge passengers with better victuals than usual. Unmetered taxi trips may be a significant issue in trendy cosmopolises. For illustration, 146 taxi automobilists were caught for illicitly refusing or at helter-skelter charging the passengers for solely 3 days in Canton, China [6]. Unmetered taxi passages are doubtable for society because of 3 major reasons. First, they hurt the standard of taxi service, particularly for trippers that are unknown the megalopolis transportation, or throughout rush hours once the vacuity of a taxi is brief. Second, they cause illegal competition between taxis and produce hassle for business stewardship. Third, they're perturbing to be tracked or operated because of that there aren't any measure records for the unmetered transportations. Ultimately, it's necessary to nominate a reserve fraud discovery algorithmic methodology for unmetered taxi travels.

However, previous ways aren't appropriate for the detection of unmetered taxi trips. First, existing approaches sometimes believe the taximeters, which isn't any longer true for unmetered taxi frauds. Their area unit has several ways to search out whether the taxi is occupied or not, as the knowledge from seat sensors, however, such data isn't correct. For instance, the device cannot distinguish whether the seat is occupied by a passenger or a bit bag. Moreover, the sensors of many taxis aren't functioning at all due to aging or deliberate damage. Supported our observation, twelve-tone music of the taxis in our knowledge set have rides with metered records nonetheless the occupancy standing showed vacantly. Hence, the occupancy data isn't reliable and can't be directly accustomed to noticing unmetered taxi frauds. Second, existing approaches assume that fraud trips exhibit abnormal behaviors from traditional metered trajectories, which is additionally not true for unmetered taxi trips.

The driving conduct or dynamics of unmetered fraud rides area unit smart like impudent metered trips than to vacant taxi circles. Hence, it's incorrect to perform anomaly spotting on metered circles. Third, living approaches treat every metered flight as one object and aim to seek out abnormal objects. Still, a compatriot in Nursing unmetered taxi trip is generally a partial section of a whole unmetered flight together with formerly the taxi was vacant. accordingly, we tend to aim to observe abnormal flight pieces rather than complete flight objects. marginally, predating strategies don't feel to be possible for this task. Hence, we tend to make Fraud Trip, a fraud discovery system specifically designed for unmetered taxi trips. First, we tend to propose the Associate in Nursing possession discovery rule to spot the occupation of taxis supported by a fine-coarse set of options generated from taxi circles. Second, we tend to apply a most fraudulent flight clarification rule on the anticipated habitation standing of taxis. The Contribution Of the Proposed work is as follows.

- We consider the unmetered taxi trips and Outliers in a trajectory that is if the taxi driver is taking a detour by identifying the fraudulent taxi trip using the KNN algorithm.
- We propose a novel system that identifies Fraud trips and whether the passenger is overcharged by the taxi driver or not is found by learning the behavior of taxis.
- We exhibit the efficiency and accuracy of the KNN algorithm in terms of predicting fraud trips.

II. RELATED WORKS

Li et al. [1] discovered the passenger-finding strategies in a Time Location-Strategy feature triplet and constructed a train/test dataset containing both top- and ordinary-performance taxi features. Belhadi et al. [2]'s approach allows for to identification of both individual and group outliers and is based on a two-phase-based algorithm. Carlo et al. [3] proposed the state of the art in the analysis of route choice behaviour within the discrete choice modelling framework. In [4], the author's method is used to detect anomalous trajectories "on-the-fly" and to identify which parts of the trajectory are responsible for their anomalousness. Zhang et al. [5] demonstrated the potential of iBAT by using it for taxi driving fraud detection and road network change detection. Nagy et al. [6] have achieved heuristic routines taken from VRP methodology to render our procedures efficient when checking feasibility, and the appropriate mathematical relationships to describe changes in the maximum load of routes. Rong et al. [7] method provides time-honoured insights into the dynamics of taxicab offerings to maximize the earnings margins for the involved parties. Lee et al. [8] proposed a partition-and-detect framework for trajectory outlier detection, to detect outlying sub-trajectories from a trajectory database. Li et al [9] suggested fully learning experienced drivers' routing decisions which are based on their implicitly estimated traffic trends. Li et al. [10] system the proactive load balancing approach that enables efficient cooperation among end-to-end load balancers which schedules the cached data among MESs based on the predicted road traffic situation. Joseph at el. [11] proposed to connect the logit and probit models, with recently developed models and discussed several extensions to the simple logit model, as well as the choice set generation problem. Powell et al. [12] system to reduce the number of cruising miles while increasing the number of live miles, thus increasing profitability, without systematic routing.

Xie et al. [13] proposed to identify the requirement and the development of machine learning-based mobile big data (MBD) analysis by discussing the insights of challenges in mobile big data. Rong et al. [14] approach present generic insights into the dynamics of taxicab services with the objective of maximizing the profit margins for the concerned parties. Yuan et al. [15] introduced smart driving directions from the historical GPS trajectories of many taxis and provide a user with the practically fastest route to a given destination at a given departure time. Yuan et al. [16] proposed that a recommender provides taxi drivers which are more likely to pick up passengers quickly and maximize the profit. Yamamoto et al. [17] proposed an adaptive routing method in cruising taxis. The pathways where many customers are expected to exist are assigned to drivers that adapt dynamically to carrying customers. In [18], the authors must add route choices for taxi drivers regarding frequently mentioned cost-based route choice rules: pursuing the shortest time, or distance, avoiding passing signalized intersections, or making left/right turnings. Zhu et al. [19] proposed a framework for conducting big data analytics in intelligent transportation systems (ITS). Masabumi et al. [20] proposed a framework to foster the development of effective ridesharing mechanisms that would promote massification. Meng et al. [21] approached a cost-effective recommender system for taxi drivers. The design goal is to maximize their profits when following the recommended routes for finding passengers. Siyuan et al. [22] designed the Speed-based Fraud Detection System, to model taxi behaviours and detect taxi fraud which is robust to location errors and independent of the map information and road networks. Zhang et al. [23] demonstrated the big data analytics application in revealing novel insights from massive taxi trace data.

Tu et al. [24]'s system for electric taxi (ET) drivers. Taxi travel knowledge, including the probability of picking up passengers and the distribution of destinations, is learned from the raw GPS trajectories. Kong et al. [25] proposed a two-stage approach, which is composed of travel requirement prediction and dynamic route planning. Zhou et al. [26] introduced a system that uses a learning model to detect unmetered trajectory segments and introduces a heuristic algorithm to construct maximum fraudulent trajectories from the trajectory dataset. Bu et al. [27] introduced a framework for monitoring anomalies over continuous trajectory streams. Ding et al. [28]'s system detects "unmetered" taxi trips based on a novel fraud detection algorithm and a heuristic maximum fraudulent trajectory construction algorithm. Lai et al. [29] proposed the idea of a unique class of "electric powered field" through isomorphic passengers and taxis as expenses with specific signs. Lyu et al. [30] consider the nearby alternative pick-up/drop-off locations to schedule a flexible route with which the detour distance and travel cost can be greatly reduced by letting some passengers walk a short and acceptable distance.

III. PROPOSED MODEL

The KNN algorithm is used to detect taxi fraud. In figure 1, for instance, the input characteristics and replies are provided in the dataset. The data from the input dataset is initially imported using the JSON library to the model. The data is then preprocessed by tokenization and stemming, which helps the data to be classified into integer categories and converted to numerical values by the bag of words to train the model using the KNN algorithm. Finally, Feature selection is done by selecting input features and converting them to bags of words to load them into the KNN module. The KNN model provides high accuracy while training the machine. By entering test data into our application after training. KNN is utilized as a classifier to classify the dataset`s input data. The model is trained to calculate and display the outcome based on the data. Tokenization and stemming are used to pre-process data before the model gets trained. Following that, the passenger's input is selected and placed in a bag of words before being fed to the KNN module. Training the data to know how accurate the model is with different K values. The K value is picked that provides the highest accuracy for both training and testing data. The KNN technique is used to train the machine by utilizing a dataset to determine what the response will be for a certain query given by the passenger, such as the total distance and price based on the passenger's feature selection and for a specific path. It will classify the input data to forecast the output based on the dataset replies, and the result will be displayed whether the taxi driver is overcharging the passenger or not is known. By this, the passenger can negotiate with the driver to reduce the fare.

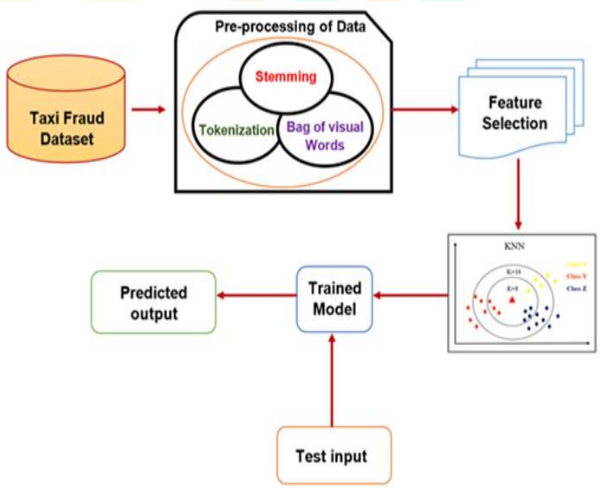


Figure 1. Proposed fraud detection model

IV. KNN MACHINE LEARNING ALGORITHM

A machine learning (ML) algorithm used for detecting the fraud trip is the KNN algorithm. It is a supervised machine learning technique that can be used to address both classification and regression issues and is straightforward to construct. It's a straightforward algorithm to comprehend and apply. Because this algorithm requires no assumption about data, it is particularly useful for nonlinear data. This algorithm is based on the idea that every data point that is close to another belongs to the same class. Classification and regression both can be performed. It has a high degree of accuracy, requires no data assumptions, and requires no extra assumptions, tuning of several parameters, or the development of a model.

When given recent untagged information, a supervised metric capacity unit rule uses a labelled input file to coach an operation that offers an appropriate output. The result of a classification and regression drawback task are separate worth and complex numbers, severally. This rule believes that objects that are like close. Because the name implies, purpose, datum, and information are accustomed to predicting the category or continuous worth for a replacement information point. An expected category for the new datum is generated by allocating a category label to the bulk of this rule from the coaching dataset. For regression, a projected continuous worth for our new datum is the Mean or median of continuous values allotted to KNN from the coaching dataset. Its many main advantages are simplicity, effectiveness, intuitiveness, and competitive classification performance in several domains. It's strong to cry coaching information and is effective if the coaching information is massive.

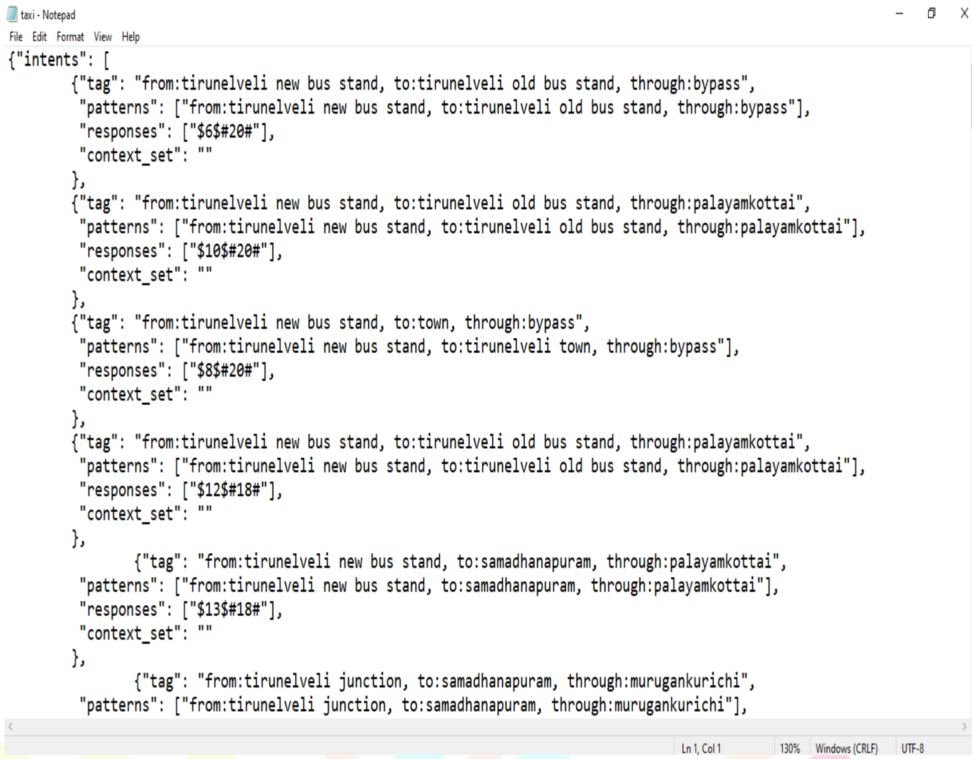
Algorithm 1 K-Nearest Neighbor

Input: A *n* x *n* distance matrix *D[1…n,1….n]* and an index x of the starting city.
Output: A list path of the vertices containing the tour is obtained.
begin
for *k* ← *l* **to** n do
    Visited [*k*]← false
    Initialize the list Path with *x*
    Visited [*x*]← true
    Current *x*
end for
for *k* ← 2 **to** *n* do
    Find the lowest elements in roe current and unmarked column j containing the element.
    Current ← l
    Visited[*l*]← true
    Add *l* to the end list Path
    Add *x* to the end list Path
end for
return Path

This algorithm evaluates the proximity of two data points using a variety of distance metrics. The closer two points are to each other, the more related and similar they are, according to this algorithm. Correlation and similarity are determined using a variety of distance metrics. Although there are many different distance functions to pick from, we should always use the ones that best fit the type of our data. This rule runs multiple times with completely different K values to seek out the worth of K that most closely fits your information. For evaluating K's performance, we'll contemplate accuracy as a data point. It's a superb candidate for our K worth if the worth of accuracy will increase fittingly to the amendment in K. once deciding the suitable worth for K, we tend to should contemplate the number of characteristics yet because of the sample size per cluster. The larger the choice we'd like to form to find the social applicable worth of K, the additional characteristics and teams we've got in our information assortment. Our predictions abate stable because the worth of K is reduced to at least one.

A. Taxi Trip Dataset

Taxi. Json is a dataset including information from the passenger's input fields as well as responses to those inputs. The data in this dataset file is in a human-readable text format that may be used to process data. JSON files are tiny text files that can be edited in a text editor. The intentions array contains all the data in the dataset. The distance from the point where the passenger boarded to the passenger's destination, as well as the passenger's path to the destination, are fields in the dataset. The passenger's input data is recorded in the tag array, while the patterns array holds the values of machine-understandable details. The tag and patterns provide fields like from, to, and through. The total number of kilometers between the boarding point and the destination is recorded in the responses, and the price per kilometer is determined. This dataset oversees pre-processing and feature selection.


taxi - Notepad

File Edit Format View Help


```
{"intents": [
        {"tag": "from:tirunelveli new bus stand, to:tirunelveli old bus stand, through:bypass",
         "patterns": ["from:tirunelveli new bus stand, to:tirunelveli old bus stand, through:bypass"],
         "responses": ["$6$#20#"],
         "context_set": ""
        },
        {"tag": "from:tirunelveli new bus stand, to:tirunelveli old bus stand, through:palayamkottai",
         "patterns": ["from:tirunelveli new bus stand, to:tirunelveli old bus stand, through:palayamkottai"],
         "responses": ["$10$#20#"],
         "context_set": ""
        },
        {"tag": "from:tirunelveli new bus stand, to:town, through:bypass",
         "patterns": ["from:tirunelveli new bus stand, to:tirunelveli town, through:bypass"],
         "responses": ["$8$#20#"],
         "context_set": ""
        },
        {"tag": "from:tirunelveli new bus stand, to:tirunelveli old bus stand, through:palayamkottai",
         "patterns": ["from:tirunelveli new bus stand, to:tirunelveli old bus stand, through:palayamkottai"],
         "responses": ["$12$#18#"],
         "context_set": ""
        },
                {"tag": "from:tirunelveli new bus stand, to:samadhanapuram, through:palayamkottai",
         "patterns": ["from:tirunelveli new bus stand, to:samadhanapuram, through:palayamkottai"],
         "responses": ["$13$#18#"],
         "context_set": ""
        },
                {"tag": "from:tirunelveli junction, to:samadhanapuram, through:murugankurichi",
         "patterns": ["from:tirunelveli junction, to:samadhanapuram, through:murugankurichi"],
```


Ln 1, Col 1 130% Windows (CRLF) UTF-8

Figure 2. The dataset provided on top of includes the computer file values, machine-readable values, and answers. The boarding purpose (from), destination (to), and route to the destination (through) information area unit are provided within the tag and patterns. The whole range of kilometers and the value per kilometer area unit hold on within the answer.

B. Preprocessing

Many packages area unit foreign throughout the coaching part to convert the info into a machine-readable format. Packages like linguistic communication area unit accustomed produce Python programs that employment with human language information for applied math linguistic communication process with Natural Language Processing (NLP). It includes tokenization, parsing, categorization, stemming, tagging, and linguistics reasoning libraries. During this project, we tend to use the tokenization procedure and stemming.

Tokenization is the method of breaking down a large body of data into smaller lines, words or maybe inventing new terms for a language aside from English. Once it involves operating with text information, tokenization is one of the foremost common activities. Tokenization is the method of breaking down a phrase, sentence, paragraph, or perhaps a whole text document into smaller parts like individual words or phrases. Tokens area unit the names given to every of those smaller units. Words, numerals, or punctuation marks may be used as tokens. By finding word boundaries, tokenization creates smaller units. This area unit the place wherever the first word ends, and the following word begins. These tokens' area units are utilized in the stemming and lemmatization method as a preliminary step. The terms in the dataset area unit are separated by utilizing these tokens.

To train the info, we tend to utilize a KNN classifier. This may be accomplished by importing the libraries into the classifier category. A Python library for scientific notation uses a method that knowledge sets array data. Importation of the OS package permits victimization of AN external file for coaching, like a JSON dataset file it helps to feature a JSON file to the coaching. This package is foreign to the coaching module victimization of packages related to machine learning. Once the import all the packages into the coaching module, the info is trained for the check method step by step. The locales provided within the dataset area unit are divided by items of words victimization the tokenization technique. The tokenized words area unit is then placed in another array for additional process, separating, and storing all the data within the dataset. Following that, the data is stemmed, the separated words' area unit additional classified, and therefore the root words for places area unit separated and placed during a separate array. The method of developing morphological variants of a root/base word is understood as stemming. Stemming algorithms or stemmers area unit terms accustomed to describing stemming programs. Data retrieval systems, like search engines, use stemming. It's employed in domain analysis to work out domain vocabularies. Stemming is advantageous as a result it reduces duplication by making certain that the word stem and it is inflected/derived terms have an equivalent which means. These arrays have currently been sorted and saved within the labels array.

KNN algorithms take knowledge and apply similarity metrics to classify the data. Each data word is converted into numerical words to feed into the KNN model because the KNN classifier works with only numerical data. The bag of words (BoW) process is used to convert words into binary knowledge (0s and 1s), with every word classed as zero or one and placed in a very new pickle file. The strategy is simple and labile. A BoW could be a text illustration that describes the frequency with which those words seem in the document. It consists of 2 parts one is a lexicon of acknowledged words and life of the presence of acknowledged words. As a result, all data concerning the document`s ordering or structure is deleted, it's dubbed a BoW. The model is simply involved with whether recognized terms occur within the document, not with the situation of these phrases. The BoW may be as straightforward or as subtle as you would like. The challenge is deciding the way to build a vocabulary of acknowledged words (or tokens) and the way to get the presence of acknowledged terms. The whole variety of records within the words is stored by another array. These stored records are currently obtained associated placed in an empty pickle file, so this algorithmic program is also trained on them. The KNN algorithmic program is employed to train the system by utilizing a dataset to predict the response to a question. Once the coaching is finished, 2 files are created: a pickle file that contains all the values and a KNN pickle file that's trained from the dataset. Once this file is ready, we don't need to train the model once more. The complete dataset is going to be trained during this module only once.

C. Feature Extraction

Once the dataset has been trained, the KNN technique permits the machine to interpret the info a lot merely and runs quicker, leading to an end in seconds. A package is employed during this testing module to assist in developing an interface within the program and building an efficient graphical interface. The text entry for the worth is additionally offered within the dialogue window as well as a label with editable text with its dimension and height that offer the labels and texts within the grid type that the row and columns area unit set to get the output, and that we give the submit button to receive the output. The module's main objective is to see for fraud mistreatment the calculable worth that the taxi driver can charge for the passenger. To do so, a taxi fraud operation is utilized. To create a pickle file a Python package is used that permits you to set up a Python object structure. Pickling an associate degree object in Python permits it to be saved on a disc. Pickle works by 1st "serializing" the item before writing it to a file. Pickling could be a Python operation that converts an inventory or different Python objects into a personality stream. The idea is that this character stream provides all the info needed to recreate the item in another Python operation.

The result response looks for similar data in the dataset and returns the results. Please fill in all fields in the window if the input is not properly given to display the message. When the testing module is initiated, a window will appear with fields such as from, to, though, one way or round trip, AC or Non-AC, and the price requested by the taxi driver. When the data is entered into the window and the submit button is pressed, a pop-up window appears with the specifics of the boarding location, destination, and transit location. The entire distance between the source and destination, the price per kilometer, the total amount for the trip, the total amount with and without air conditioning, and the total amount for up and down are then displayed. The result of whether fraud has been identified will be displayed at the end.

V. EXPERIMENTAL RESULTS

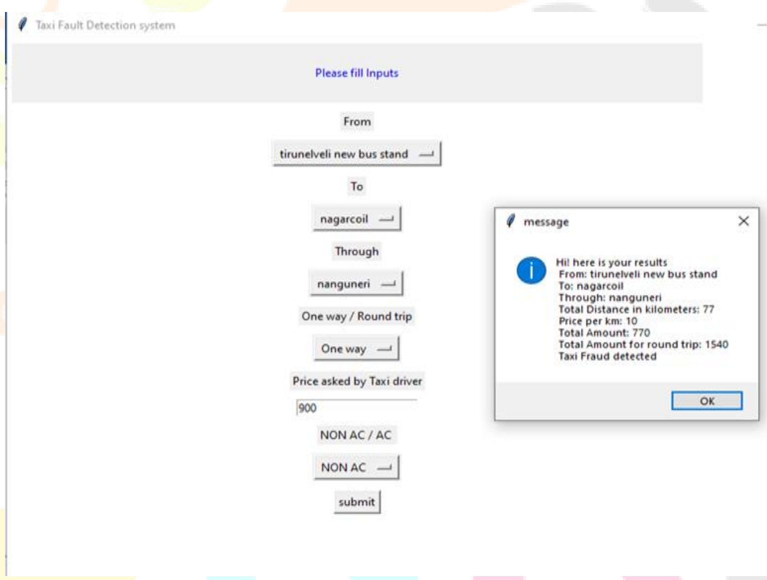


Figure 3. Fraud detection for the price offered by the driver who does not have air conditioning.

The system verifies whether the charge requested by the taxi driver is larger than the expected amount or not when the price requested by the taxi driver is given. If it is high, fraud has been detected. Because the requested amount for non-AC is higher, taxi fraud is detected in the above figure.

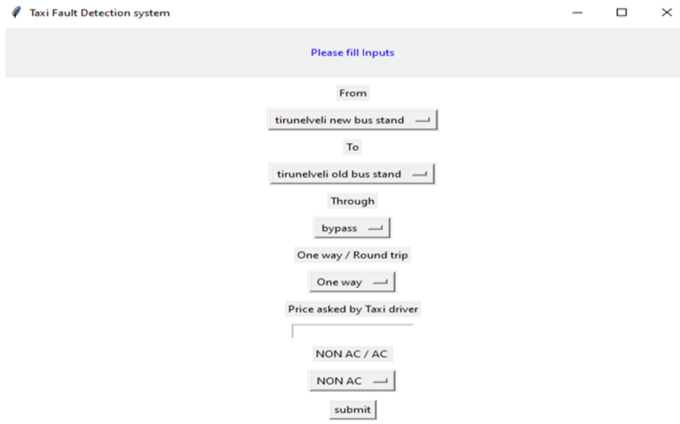


Figure. 4. Initial User interface Window

To access the input values, open the user interface window. Before the passenger offers the input, this window appears with the specified locations. It includes fields such as from, to, though, one-way, or round-trip, AC or non-AC, and so on. If the taxi driver requests a fare, enter that amount in the Fare Requested by the Taxi Driver field to detect fraud.

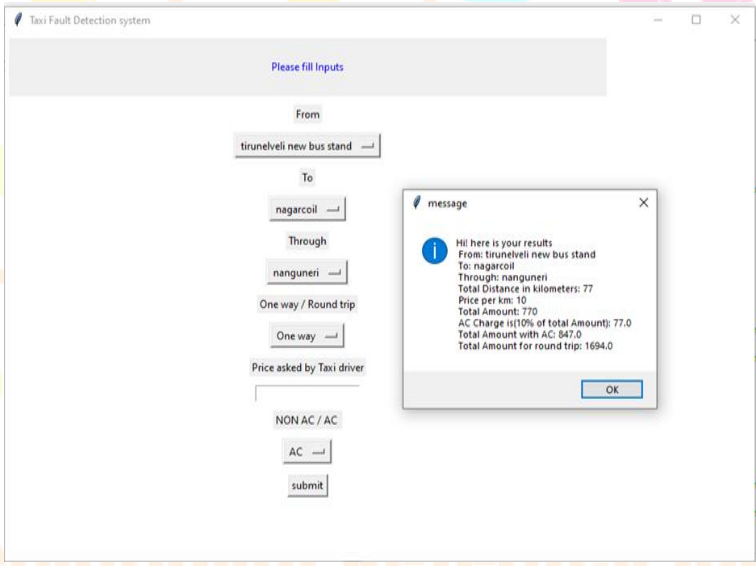


Figure. 5. Price Prediction for AC trip.

Following the user's input submission. A tiny dialogue box appears with the trip's details. Provides the trip's total mileage, price per kilometer, and air conditioning, AC charges 10% of the total price, plus the AC charge.

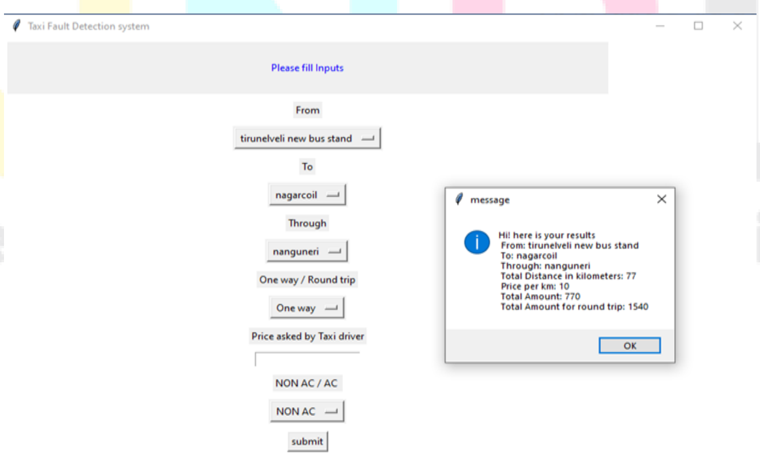


Figure. 6. Price prediction for the non-AC trip.

The cost of a one-way non-AC trip from the boarding point to the destination is computed and displayed in the above figure.

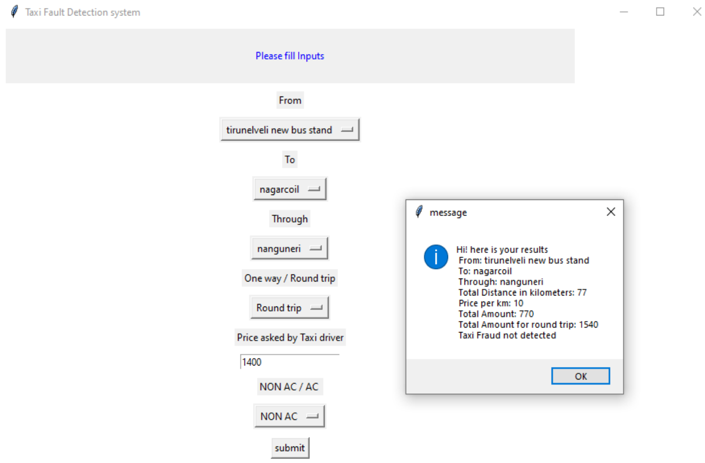


Figure. 7. Fraud detection with round trip and non-AC.

The round-trip price requested by the taxi driver is less than the estimated production. As a result, there was no detection of taxi fraud.

We looked at metered and unmetered taxi trips in real-world data, including distance and pricing. To determine whether the taxi driver's charges are reasonable. We developed the innovative system Fiddle Tour: Fraudulent Taxi Trip Detection Using KNN Machine Learning Algorithm, which predicts taxi fraud using distance and price for the path taken. The KNN algorithm is used to train the dataset to find the response for the exact Travel path specified by the passenger. To receive an answer based on the passenger's needs, such as one-way, round-trip, AC, or non-AC travel, as well as the trip's cost. The efficiency of fraud trip prediction is excellent, and it executes quickly when this Algorithm is used. The fraud trip prediction is only done in this project; we have not visualized the travel path, which will be explored and implemented in future works.

REFERENCES

[1] B. Li et al., "Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset," 2011 IEEE International Conference on Pervasive Computing and Communications Workshops 2011, pp.63-68, doi:10.1109/PERCOMW.2011.5766967.

[2] Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, A. Cano and J. C. -W. Lin, "A Two-Phase Anomaly Detection Model for Secure Intelligent Transportation Ride-Hailing Trajectories," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4496-4506, July 2021, doi: 10.1109/TITS.2020.3022612.

[3] C. G. Prato, "Route choice modeling: Past, present, and future research directions," J. Choice Model., vol. 2, no. 1, pp. 65–100, 2009, doi:10.1016/S1755-5345(13)70005-8.

[4] Chen et al., "iBOAT: Isolation-Based Online Anomalous Trajectory Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 2, pp. 806-818, June 2013, doi: 10.1109/TITS.2013.2238531.

[5] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "ibat: detecting anomalous taxi trajectories from GPS traces," in Proceedings of the 13th international conference on Ubiquitous computing. ACM, 2011, pp. 99–108, doi.org/10.1145/2030112.2030127.

[6] G. Nagy and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries," European journal of operational research, vol. 162, no. 1, pp. 126–141, 2005, https://doi.org/10.1016/j.ejor.2002.11.003.

[7] H. Rong et al., "Mining Efficient Taxi Operation Strategies from Large Scale Geo-Location Data," in IEEE Access, vol. 5, pp. 25623-25634, 2017, doi: 10.1109/ACCESS.2017.2732947.

[8] J. Lee, J. Han and X. Li, "Trajectory Outlier Detection: A Partition-and-Detect Framework," 2008 IEEE 24th International Conference on Data Engineering, 2008, pp. 140-149, doi: 10.1109/ICDE.2008.4497422.

[9] J. Li et al., "A Traffic Prediction Enabled Double Rewarded Value Iteration Network for Route Planning," in IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4170-4181, May 2019, doi: 10.1109/TVT.2019.2893173.

[10] J. Li et al., "An End-to-End Load Balancer Based on Deep Learning for Vehicular Network Traffic Control," in IEEE Internet of Things Journal, vol. 6, no. 1, pp. 953-966, Feb. 2019, doi: 10.1109/JIOT.2018.2866435.

[11] J. N. Prashker and S. Bekhor, "Route choice models used in the stochastic user equilibrium problem: A review," Transp. Rev., vol. 24, no. 4, pp. 437–463, 2004, doi:10.1080/0144164042000181707.

[12] J. W. Powell, Y. Huang, F. Bastani, and M. Ji, "Towards reducing taxi cruising time using Spatio-temporal profitability maps." in SSTD. Springer, 2011, pp. 242–260, https://doi.org/10.1007/978-3-642-22922-0_15.

[13] J. Xie, Z. Song, Y. Li, Y. Zhang, H. Yu, J. Zhan, Z. Ma, Y. Qiao, J. Zhang, and J. Guo, "A survey on machine learning-based mobile big data analysis: Challenges and applications," Wireless Commun. Mob. Comput., vol. 2018, 19 pages, 2018, https://doi.org/10.1155/2018/8738613.

[14] J. Yang, J. Dong, Z. Lin, and L. Hu, "Predicting market potential and environmental benefits of deploying electric taxis in Nanjing, China," Transport. Res. Part D-Transport. Environ., vol. 49, pp. 68–81, 2016, doi:10.1109/ACCESS.2017.2732947.

[15] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in Proceedings of the 18th SIGSPATIAL International conference on advances in Geographic Information systems. ACM, 2010, pp.99–108, https://dl.acm.org/doi/pdf/10.1145/1869790.1869807.

[16] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," in Proceedings of the 13th international conference on Ubiquitous computing. ACM, 2011, pp. 109–118, https://doi.org/10.1145/2030112.2030128.

[17] K. Yamamoto, K. Uesugi, and T. Watanabe, "Adaptive routing of cruising taxis by mutual exchange of pathways," in International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, 2008, pp. 559–566, doi:10.1504/IJKESDP.2010.030466.

[18] L. Li, S. Wang and F. -Y. Wang, "An Analysis of Taxi Driver's Route Choice Behavior Using the Trace Records," in IEEE Transactions on Computational Social Systems, vol. 5, no. 2, pp. 576-582, June 2018, doi: 10.1109/TCSS.2018.2831285.

[19] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, pp. 383-398, Jan. 2019, doi: 10.1109/TITS.2018.2815678.

[20] M. Furuhata, M. Dessouky, F. Ordo´nez, M.-E. Brunet, X. Wang, and ˜ S. Koenig, "Ridesharing: The state-of-the-art and future directions," Transportation Research Part B: Methodological, vol. 57, pp. 28–46, 2013, doi: 10.1016/j.trb.2013.08.012.

[21] M. Qu, H. Zhu, J. Liu, G. Liu, and H. Xiong, "A cost-effective recommender system for taxi drivers," in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014, pp. 45–54, doi.org/10.1145/2623330.2623668.

[22] S. Liu, L. M. Ni, and R. Krishnan, "Fraud Detection from Taxis' Driving Behaviors," in IEEE Transactions on Vehicular Technology, vol. 63, no. 1, pp. 464-472, Jan. 2014, doi: 10.1109/TVT.2013.2272792.

[23] S. Zhang and Z. Wang, "Inferring passenger denial behavior of taxi drivers from large-scale taxi traces," PloS one, vol. 11, no. 11, 2016, doi: 10.1371/journal.pone.0165597.

[24] W. Tu et al., "Real-Time Route Recommendations for E-Taxies Leveraging GPS Trajectories," in IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 3133-3142, May 2021, doi: 10.1109/TII.2020.2990206.

[25] X. Kong, M. Li, T. Tang, K. Tian, L. Moreira-Matias and F. Xia, "Shared Subway Shuttle Bus Route Planning Based on Transport Data Analytics," in IEEE Transactions on Automation Science and Engineering, vol. 15, no. 4, pp. 1507-1520, Oct. 2018, doi: 10.1109/TASE.2018.2865494. (10)

[26] X. Zhou, Y. Ding, F. Peng, Q. Luo, and L. M. Ni, "Detecting unmetered taxi rides from trajectory data," 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 530-535, doi: 10.1109/BigData.2017.8257968.

[27] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, "Efficient anomaly monitoring over moving object trajectory streams," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009, pp. 159–168, doi:10.1145/1557019.1557043.

[28] Y. Ding, W. Zhang, X. Zhou, Q. Liao, Q. Luo, and L. M. Ni, "Fraud Trip: Taxi Fraudulent Trip Detection from Corresponding Trajectories," in IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12505-12517, 15 Aug.15, 2021, doi: 10.1109/JIOT.2020.3019398.

[29] Y. Lai, Z. Lv, K. -C. Li and M. Liao, "Urban Traffic Coulomb's Law: A New Approach for Taxi Route Recommendation," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 8, pp. 3024-3037, Aug. 2019, doi: 10.1109/TITS.2018.2870990.

[30] Y. Lyu, V. C. S. Lee, J. K. Ng, B. Y. Lim, K. Liu, and C. Chen, "Flexi-Sharing: A Flexible and Personalized Taxi-Sharing System," in IEEE Transactions on Vehicular Technology, vol. 68, no. 10, pp. 9399-9413, Oct. 2019, doi: 10.1109/TVT.2019.2932869.