FAKE GPS DETECTION USING MACHINE LEARNING MODELS

LINH VUU

MSc Data Science

Thesis

AUGUST 2022

ACKNOWLEDGEMENTS

Words cannot express my gratitude to my thesis supervisor for his invaluable feedback. I also could not have undertaken this journey without the generous support from GeoComply Company, who financed my master study. I am also grateful to my teammates for their help at work when I was on leave to have time working on the thesis.

Finally, I would be negligent if I did not mention my parents and friends. Their belief in me has kept my spirits and motivation high during this process. I would like to thank my poodle for all the entertainment and emotional support as well.

ABSTRACT

GOJEK is a location-based mobile application providing transportation services using the Global Positioning System (GPS) to detect the drivers and the customers' position. Due to the high competition among the drivers and attractive bonuses paid from the company to the most active drivers, many drivers try to use Fake GPS location application to spoof their locations to receive more orders. As a result, the honest drivers would be impacted detrimentally as they lose their chances to approach customers. To deal with this problem, this proposal suggests using machine learning algorithms to detect spoofing cases using GOJEK public dataset. Several models, namely Logistic Regression, Decision Tree and Random Forest will be applied to the dataset and evaluated based on the specificity and accuracy to select the best one. This research would have a significant impact on GOJEK company as they can protect innocent drivers and reduce the waiting time for the customers by eliminated cases where drivers are too far away but fake location to be near them. Further research in other industry can also use this implementation as an example of how to detect location spoofing. This research needed Jupyter Notebook, Anaconda and took around four months to go from the research phase to the modelling and the final stage of reporting. The result indicated that Decision Tree would be the best model to deal with GOJEK dataset.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Definition
API	application programming interface
CSV	Comma-separated values
CV	Cross-validation
FN	False Negative
FP	False Positive
GIS	Geolocation Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning Systems
IoT	Internet of Things
KNN	K-nearest neighbors
NASA	National Aeronautics and Space Administration
PMUs	Phasor measurement units
RF	Radio Frequency
RFE	Recursive feature elimination
sqft	square feet
ST	space-time
TN	True Negative
TP	True Positive
XML	Extensible Markup Language

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background of the Study

More than 70% of people around the world have smart phones with Global Positioning Systems (GPS) and Wifi capabilities as a result of technology evolution (Haddad et al., 2014). Using cellular network stations, GPS and wifi positioning, a lot of mobile applications can acquire locational information to provide location-based services. One of cell phone applications applying location navigation features to transportation successfully is GOJEK (Mar'i and Pangestu, 2021).

Moslehpour et al. (2022) informed in their paper that GOJEK Indonesia is one of the largest technology companies in the mobile commerce. Established in 2010, GOJEK has evolved from a ride-hailing phone service into an advanced technology firm providing location-based services on mobile application, both transportation and delivery. 7.1 billion USD was contributed by GOJEK to economy in Indonesia in 2019 (Florene, 2020) and received investment from large companies such as Google and Mitsubishi (Chao, 2020). In 2021, Mar'i & Pangestu (2021) noted that the GOJEK application had almost 2 million driver partners and 200 million downloads, showing that this application has been known commonly.

According to Nandi (2019), in their online survey conducted between September and November of 2017, 96% of responses admitted that they used motorcycle application in their daily activities. 53% of them chose GOJEK over Grab and Uber, which demonstrated a huge demand for GOJEK in recent years. As a result, the number of GOJEK drivers also increased significantly. Unemployed people became GOJEK drivers because the job does not require having a university degree. Also, a lot of employees resigned from their jobs to be drivers due to the attractive income and the flexible working hours. In fact, according to a survey conducted by Hidajat et al. (2021) sixty per cent of drivers chose to work full time because they had no other job and forty per cent of them previously had jobs but then quit and became drivers.

Apart from the advantages, there are several issues in this industry, including frauds committed by drivers (Damaini et al., 2018). To take benefits from the applications without moving, drivers

use a fake GPS application to make their phones locating them in different places from their current positions. This would help them to reach the customers faster or even create fake trips, which is unfair for other drivers working honestly. Interestingly, in 2018, the Institute for Development of Economics and Finance surveyed 516 online drivers and found that 61% of drivers knew that their fellows were cheating (Hidajat et al., 2021).

To solve this issue, this paper applies machine learning, a computer science field that studies a given data, defines the patterns and makes predictions on the new dataset (Simon et al., 2016). According to Lim (2019), machine learning was leveraged to deal with problems in logistics, economics and transport, including dispatching the right drivers, estimating the correct prices, recommending the favourite food and building trust with their customers. Thanks to that, they had more than 20 products in 4 countries with hundreds of million orders per month (Lim, 2019). To categorise Fake GPS cases in GOJEK dataset, Decision tree, Random Forest Classifiers and Logistic Regression will be applied on GOJEK dataset in this research to see which one is suitable for fraud detection in the GOJEK case since they are common algorithms using for binary classification (Bahel et al., 2020).

1.2 Related Research

GPS, the only completely functional Global Navigation Satellite System (GNSS), was found and implemented by the United States Department of Defense. Twenty-four satellites orbit the earth, transmit signals to devices with GPS function, determining their location, direction, and speed (Maddison and Ni Mhurchu, 2009). The positions those devices are calculated by the distance between themselves and three or more GPS satellites, using trilateration, a mathematical technique. Nowadays, GPS plays a significant role in life with a wide range of applications, such as security, navigation system, commercial and scientific research (Rustamov, 2018). The more popular positioning technology is, the harder people try to manipulate it with different location spoofing techniques. To illustrate, the location was spoofed to spread fake news in society (Maheshwari, 2016), to prevent protestors from being attacked by Egyptian police (Lotan et al., 2011), and to mislocate ships in the Mediterranean Sea (Anon, 2013). Location spoofing was used increasingly in the Internet of Things (IoT) (Koh et al., 2016) and in cyber warfare (Lee and Buehrer, 2012). Players spoofed location in Pokemon Go,

a location-based game, to catch more pokemon and receive more in-game items (Zhao & Zhang, 2019).

1.3 Aim and Objectives

The principal purpose of this paper is to use the same GOJEK public dataset with different analysis approaches attempting to improve the algorithm, evaluate performance of other machine learning models and produce better results, increasing the rate of correctly defining fake GPS trips. The successful identification of spoofing location using trips' details allows technology transportation companies to treat their driver partners fairly and to reduce losses in ghost trips where their partners had the capabilities of teleportation.

This study aims for the three purposes:

- To find patterns by analysing relationships between the details of drivers' ping signals.
- To build different models to automatically decide whether an order is real or not.
- To compare and evaluate the performance of models, then select the most accurate model.

1.5 Scope of the Study

The scope of this study is to use GOJEK published dataset to predict fake location cases based on drivers' behaviours. The limitation of the scope is that the dataset only contains a few features describing the orders. Other data can also be collected from the application to define whether the location is spoofed or not, such as the devices' information, moving velocity and vertical accuracy. However, those details can be either very hard to collect by the software or so confidential that cannot be published to the community for researching purposes. Nevertheless, it is not in the scope of this dataset and this paper.

1.6 Significance of the Study

This study helps to define fake GPS rides, thus, it brings benefits not only to online transportation service providers, their customers, and all legal drivers but also to other location-based applications.

By recognising fraud driving activities, the companies can reduce their losses when not giving rewards to spoofers and all honest drivers can have opportunities to earn bonuses through performance incentives. For example, GOJEK drivers can earn Rp. 200,000.00 when collecting 30 points on GoRide service and earn Rp. 400,000.00 with 21 points on GoCar service (Jakarta, 2019). This leads to drivers using multiple phones, accounts and spoofing location applications to artificially create and deliver driving services (Khong, 2019). Recently, Grab suffered a case of payment fraud on Grabhitch, losing $41,000 (RM165,000) and arrested ten suspects while according to Istiardi & Setiawati (2022), GOJEK lost Rp. 500,000,000.00 paying to perpetrators committing fraud orders.

When stopping drivers to fake their locations to approach customers, the companies can ensure a true waiting time for the customers and a fair working environment for other drivers. Foo Wui Ngiap, head of trust, identity, safety and info security at Grab indicated that drivers used spoofing apps to be located near customers when they were actually not (Khong, 2019). Because of that, nearby drivers could not take that ride and the fraudsters took more time to come to the passengers compared to what Grab app estimated. As a result, other drivers would lose their income and customers would not be happy and think that Grab allocate rides inaccurately.

This research is also helpful for other location-based applications, like Pokémon Go, StepN, as a reference to detect location spoofing based on users' behaviours. Although other applications can collect different data and users can have different behaviours, some approaches to detect location spoofing and algorithm used in building and comparing models can be utilised to prevent spoofing in general.

1.7 Structure of the Study

This paper is organized into six chapters. Chapter one is an introduction of the study, containing background, objectives, scope and significant of the study. Chapter two investigates different location spoofing methods and reviews existed solutions to detect location spoofing. Section three describes research methodology used in this research to detect fake GPS location in GOJEK dataset. Chapter four analyses the data and the models in details. Chapter five discusses more the experiment results. Chapter six concludes the research with some recommendations, the limitation of this research and suggest possible future improvement.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The second chapter will discuss literature review on previous studies of location spoofing as figure 1. Methods to interfere locations will be examined. Then, geospatial data and suspicious signs of location spoofing will be introduced. Application of machine learning will come next and finally, the use of machine learning in location spoofing detection from other papers will be analysed.

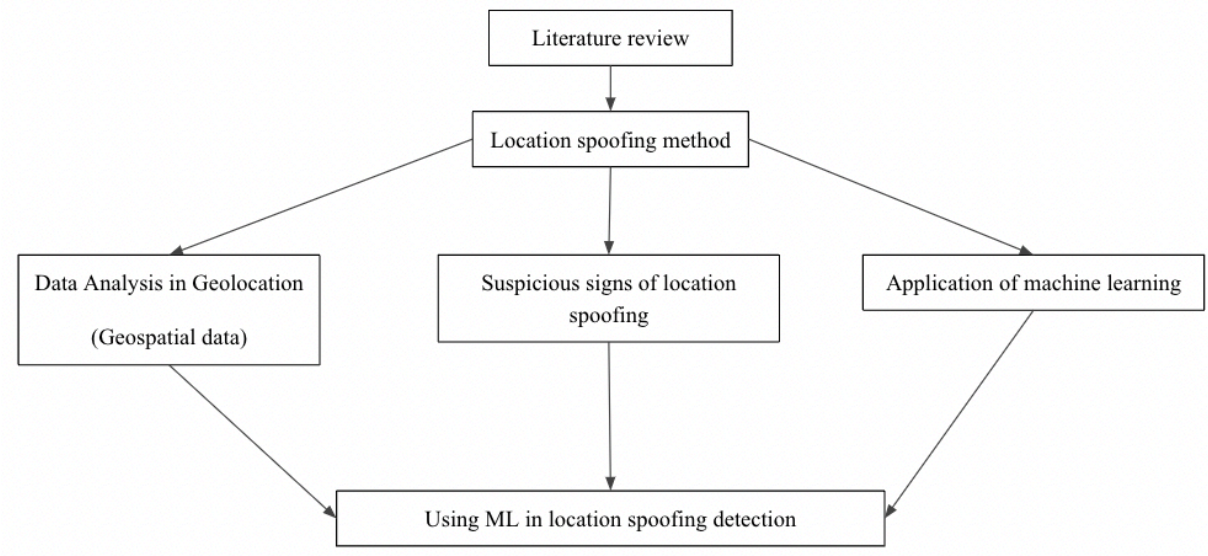

Figure 1. Flow of Literature Review

2.2 Location spoofing methods

Wong and Yiu (2020) mentioned that location spoofing is to change the location purposely. Traditionally, GPS spoofing attack is the act of altering the radio signal from the GPS satellites to transmit non-accurate coordinates to the target device as the left arrow of figure 2. Later, as the mobile phone is developed, location spoofing is to fake GPS location on location-based applications illustrated in the right arrow of the figure.

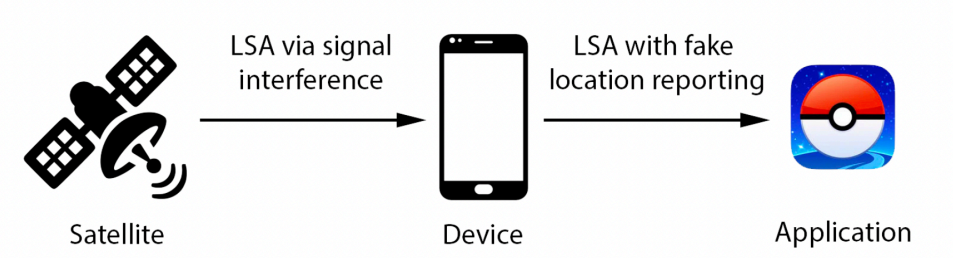

Figure 2. Location spoofing attacks (Wong and Yiu, 2020)

Similarly, Van Der Merwe et al. (2018) classified the spoofing attacks into four layers with radio frequency (RF) based and interface based (figure 3). The first level is the deployment architecture, including how many and where transmitters are used. In the second level, the take-over strategy forces a receiver to lock onto the fake signal. The level three is to control the position, auto-pilot and timing attack. The final level is the application which is the interface of the spoofing system to bypass the GNSS.

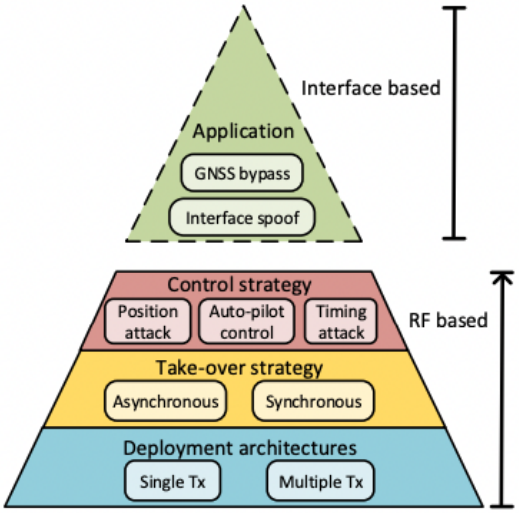

Figure 3. Spoofing classification layers (Van Der Merwe et al., 2018)

2.2.1 Signal interference

According to Chapman (2017), because the GPS's structure is known publicly, it is not difficult to build a system to broadcast false GPS signals to receivers, so that those devices display incorrect GPS locations. GPS spoofing attacks were used in warfare, as GPS spoofing can fake ships', airplanes' and vehicles' locations confusing the enemy, like the Iran war in 2011 (Psiaki and Humphreys, 2015). Later, several spoofing tests were conducted, such as the drone attack (Kerns et al., 2014), and yacht spoofing (Bhatti and Humphreys, 2017). Recently, it has become accessible to ordinary hackers by using a portable radio signal transmitter and open-source software. To illustrate, the device can be bought online under $500 on greatscottgadgets.com. The software can be downloaded from GitHub, a publicly accessible code repository. A number of researchers have worked on RF based location spoofing detection.

Ahmad et al. (2019) revealed how GPS infrastructure was vulnerable and at which stages a GPS receiver could be impacted by an attacker. Then, they delivered a detailed study of two main anti-spoofing techniques. Spoofing detection techniques were used to discriminate the spoofing signals so spoofing attacks could be detected. Mitigation techniques were used to perform countermeasures against the fake signals to eliminate the impact of the spoofing methods. Both techniques were very difficult to do because they were quite complex and required external equipment to monitor several powers and signals.

Siamak et al. (2019) presented a method to detect GPS spoofing in power grids by using a dynamic estimator to online estimate differences in phases' angles stemmed from different spoofing attacks. Using the proposed technique, multiple spoofing attacks were detected quickly, and helped to reduce attacks' effects. The performance of proposed approach was then compared with the results of other static techniques and was proved that the accuracy was increased, and the detection time was reduced significantly.

Liu et al. (2019) recommended a GPS spoofing detection technique, called GPS-Probe, which analysed continuously received signal strength indicator and timestamps at server of air traffic control messages broadcasted periodically by aerial vehicles and monitored by ground sensors. It used a machine learning algorithm to estimate the target's real position and detect whether it was controlled by GPS spoofing attacks. The model did not require any updates of GPS

infrastructure or receivers. Furthermore, it achieved the accuracy of 81.7% and precision 85.3% on average. In the best scenario, the accuracy and precision were 89.7% and 91.5% respectively.

Jansen et al. (2018) created a system to detect and localise GPS spoofing attacks on flying targets. It monitors the air traffic from GPS-derived position advertisements and localise spoofing attacks by an independent infrastructure located on the ground continuously monitoring the contents and the arrival times of advertisements. It was evaluated with simulation data and real-world data from a crowdsourced air traffic monitoring sensor network. It detected spoofing attacks in less than two seconds and localised the attacker up to 150 meters of accuracy after monitoring 15 minutes.

While exploring whether multiple prevention measures could be combined to an effective spoofing countermeasure, the paper of Haider and Khalid (2017) pointed out techniques to protect GPS receiver and examined proposed solutions to prevent GPS spoofed including "Multi-test Detection and Protection Algorithm against Spoofing Attacks on GNSS Receivers" (Jovanovic et al., 2014), "GPS Spoofing Countermeasures" (Warner et al., 2003), "An Asymmetric Security Mechanism for Navigation Signals" (Kuhn, 2004), "A Cross-Layer Defense Mechanism against GPS Spoofing Attacks on PMUs in Smart Grid" (Fan et al., 2015), "Detection and Mitigation of GPS Spoofing Based on Antenna Array Processing" (Magiera and Katulski, 2015); and "GPS Spoofing Detection via Dual-Receiver Correlation of Military Signals" (Psiaki and Humphreys, 2015). They concluded that almost all systems could prevent just simple spoofing attack, only Kuhn (2014)'s and Psiaki and Humphreys (2015)'s methods had the ability to defend sophisticated attack. However, they were not the most effective methods because their implementations took a lot of time, were so expensive and they also needed to satisfy the evaluation criteria in table 1. Thus, there was no perfect technique to effectively detect GPS spoofing and more techniques should be combined to provide better solutions.

Table 1. Non-Machine learning solutions to detect spoofing via signal interference

Source	Solution Technique	Evaluation
(Liu et al., 2019)	A GPS spoofing detection technique, called GPS-Probe, which analysed continuously received signal strength indicator and timestamps at server of air traffic control messages broadcasted periodically by aerial vehicles and monitored by ground sensors. It used a machine learning algorithm to estimate the target's real position and detect whether it was controlled by GPS spoofing attacks.	The model did not require any updates of GPS infrastructure or receivers. Furthermore, it achieved the accuracy of 81.7% and precision 85.3% on average. In the best scenario, the accuracy and precision were 89.7% and 91.5% respectively.
(Ahmad et al. 2019)	Two main anti-spoofing techniques: Spoofing detection techniques were used to discriminate the spoofing signals so spoofing attacks could be detected. Mitigation techniques were used to perform countermeasures against the fake signals to eliminate the impact of the spoofing methods.	Both techniques were very difficult to do because they were quite complex and required external equipment to monitor several powers and signals.
(Siamak et al., 2019)	Method to detect GPS spoofing in power grids by using a dynamic estimator to online estimate differences in phases' angles stemmed from different spoofing attacks.	Using the proposed technique, multiple spoofing attacks were detected quickly, and helped to reduce attacks' effects. The performance of proposed approach was then compared with the results of other two static

		methods and was proved that the accuracy was increased, and the detection time was reduced significantly.
(Jansen et al., 2018)	A system to detect and localise GPS spoofing attacks on flying targets. It monitors the air traffic from GPS-derived position advertisements and localise spoofing attacks by an independent infrastructure located on the ground continuously monitoring the contents and the arrival times of advertisements.	The system was evaluated with real-world data from a crowdsourced air traffic monitoring sensor network and by simulations. It was able to globally detect GPS spoofing attacks in less than two seconds and to localise the attacker up to an accuracy of 150 meters after 15 minutes of monitoring time.
(Haider and Khalid, 2017)	Multiple prevention techniques were examined: "Multi-test Detection and Protection Algorithm against Spoofing Attacks on GNSS Receivers" (Jovanovic et al., 2014), "GPS Spoofing Countermeasures" (Warner et al., 2003), "An Asymmetric Security Mechanism for Navigation Signals" (Kuhn, 2004), "A Cross-Layer Defense Mechanism against GPS Spoofing Attacks on PMUs in Smart Grid" (Fan et al., 2015), "Detection and Mitigation of GPS Spoofing Based on Antenna Array Processing" (Magiera and Katulski, 2015); and "GPS Spoofing	Those systems were evaluated using the following criteria: • Quick implementation • Cost effective • Be able to prevent simple / intermedia / sophisticated attack • Easy to test

	Detection via Dual-Receiver Correlation of Military Signals" (Psiaki and Humphreys, 2015)	• Require external hardware Almost all systems could prevent just simple spoofing attack, only Kuhn (2014)'s and Psiaki and Humphreys (2015)'s methods had the ability to defend sophisticated attack. However, they were not the most effective methods because their implementations took a lot of time, were so expensive and they also needed to satisfy the evaluation criteria in table 1. Thus, there was no perfect technique to effectively detect GPS spoofing and two or more techniques should be combined to provide better solutions.

2.2.2 Location spoofing applications

On Android, to spoof location, users can use mock location apps, one of the most popular applications with more than 500,000 downloads on the Google Play store, root devices or emulators (Qadri, 2021). With GPS-spoofing apps, like Lockito, users can easily modify their location by using the mock location on Android to set the spoofing app as a mock location application (ABIOLA, 2017). To prevent spoofing, mock location setting can be checked by using "Location.isFromMockProvider()" for each location detecting that the location was from a mock provider when the API returns true (Android Developers, 2022). However, this method can bypass by converting those applications into system applications (iSunshare, 2022). To hack into the system, root method is used. After being rooted, users have the right to gain privilege to access the core of the operating systems and modify any information including the GPS locations. Hence, to prevent spoofing, rooted devices should be prohibited using SU Binary Files And Packages, ROMs, Checking for Dangerous Props, checking Common Apps Found On Rooted Device, Unusual Permissions For Partitions And System Directories or Commonly Used Root Cloaking Apps (Pandita, 2022). However, there are third party applications to hide the root status of the devices and also hide suspicious processes, such as Magisk Manager (Magisk Manager, 2022). Another approach to fake location is to use emulators, such as Bluestack (Bluestacks, 2022) and Memu Play (Memuplay, 2022). They allow users to use Android applications on computers, and they also come with fake location and root features. GOJEK Driver application should have function to detect emulators as well as the suggestion in Android Emulator Detection article (Chong, 2021).

On iOS, as Apple Store does not allow mock location apps, jailbreaking is used similarly to root on Android, to access root of the system and any applications not available on App Store (George, 2022). This can be detected by checking paths that only exist on the jailbroken devices (Cataluna84, 2019). If user do not want to jailbreak their devices, they can spoof location by connecting their iPhones with Macbooks and use Xcode (Young, 2022), 3uTools (3UTools, 2022) or iTools (iTools, 2022). To deal with this method, GOJEK application needs to prevent users to connect their phones to laptops. However, there is no current method available.

Table 2. Non-Machine learning solutions to detect spoofing applications

Source	Spoofing Technique	Solution	Limitation
(Young, 2022)	iOS simulation: use Xcode (Young, 2022), 3uTools (3UTools, 2022) or iTools (iTools, 2022)	No current method available	Can be a risk to GOJEK
(Pandita, 2022)	Root method on Android	Rooted devices should be prohibited	There are third party applications to hide the root status of the devices and also hide suspicious processes, such as Magisk Manager (Magisk Manager, 2022)
(Qadri, 2021)	Mock location apps on Android	Using Location.isFromMockProvider() for each location detecting that the location was from a mock provider when the API returns true (Android Developers, 2022)	Can be bypassed by converting those applications into system applications (iSunshare, 2022)
(Chong, 2021)	Android emulators	Emulators should be prohibited	Can only be used when fake location is performed on emulators
(Cataluna84, 2019)	Jailbreak on iOS	Check paths that only exist on the jailbroken devices	Can only be used when fake location is performed on jailbroken devices

2.3 Data Analysis in Geolocation

Although GOJEK Driver app can apply the suggestions above to detect which movement is fake, there are spoofing techniques that cannot be detected. According to Arias-Pérez et al. (2022), in their research, they highlighted that data play a very crucial role in solving business problems. In the GOJEK's case, data can be used to determine if the drivers were spoofing locations or not. The data, representing location of a physical object in geographical coordinate system as latitude and longitude, is called geospatial data or geographic information (Zola and Fontecchio, 2021). Latitude and longitude can be referred to specific address on the earth's surface, like plats' numbers, streets, cities, states and countries using geocoding method.

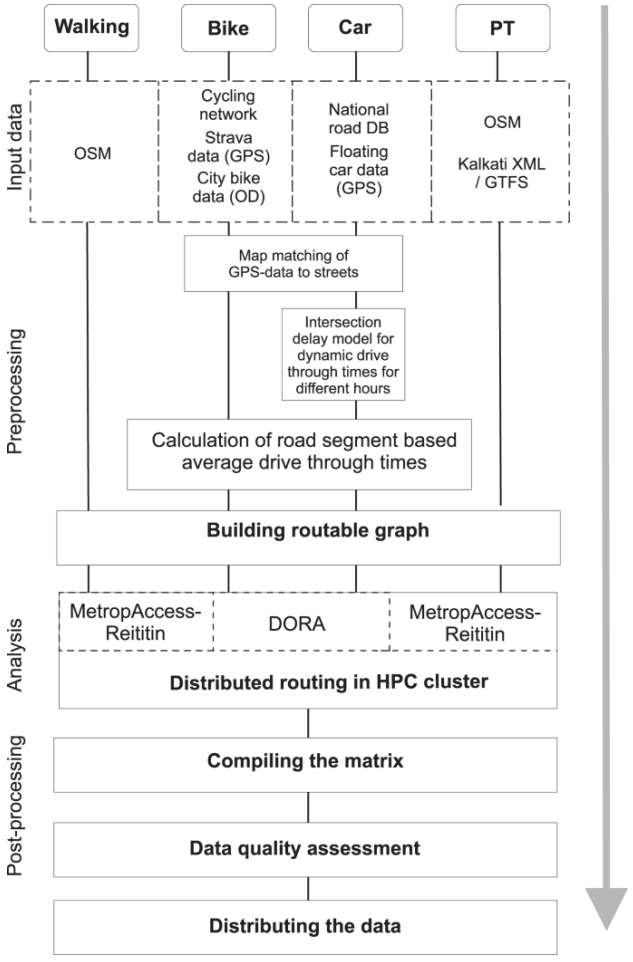

Figure 4. Generic workflow of the analytical steps that were taken to produce the travel time and distance data for Helsinki Region (Tenkanen and Toivonen, 2020)

Spatial data helped a lot to deal with emerging problems. To illustrate, through the Copernicus program, collected by satellites orbiting the Earth, over 10 petabytes/year data were analysed to generate indicators, maps, past's trends and future prediction, which all are freely access on the internet (Stanley and Gunn, 2018). Those helped to prevent natural disasters, highlight low resources areas, control renewable solar energy and improve air quality (Copernicus EU, 2017), such as the ASIMUTH project launched in 2009, aided farmers successfully to detect harmful algal bloom on the European Atlantic coast through satellites' data monitoring and analysis (Robinson and Scassa, 2022). From a wide range of datasets, namely OpenStreetMap, Strava cycling GPS data, Kalkati XML, Digiroad and City bike data of Helsinki, Tenkanen and Toivonen (2020) produced the Helsinki Region Travel Time Matrix as figure 4 above by comparing several travel modes, like public transport, private cars, cycling and walking. The matrix provided insights of the urban accessibility conditions and regional changes comparisons, to support decision making and geospatial planning in the study area.

Figure 5 defined a process of extracting insight from the geolocation datasets by finding common patterns in the data (Kotawadekar, 2021). This can be applied in different sectors and help many organisations to make decisions using spatial datasets, such as National Cancer Institute, National Aeronautics and Space Administration (NASA), National Geospatial-Intelligence Agency, the US Department of Transportation and National Institute of Justice. For instance, environmental technologies monitor tidal and temperature patterns via spatial data. From historical data of weather and geolocation, scientists can monitor high risk locations and predict natural disasters. Geospatial data was used in logistics in order to support companies know which delivery method would work well at a specific area, make correct estimations on time deliveries and track orders in real time. In ecology management, researchers use Geolocation Information System (GIS) tools to classify remote sensing images for mapping forest coverage. Regarding social safety, crime analysts can discover from the crime event maps hotspot patterns so the police resources can be effectively allocated. In transportation, analysts investigate historical GPS trajectories to make recommendation on fastest routes to travel from one place to another. Spatiotemporal data mining techniques can be used by epidemiologists to detect disease outbreak. The data can also be applied to domains like earth science, climatology, IoT and precision agriculture.

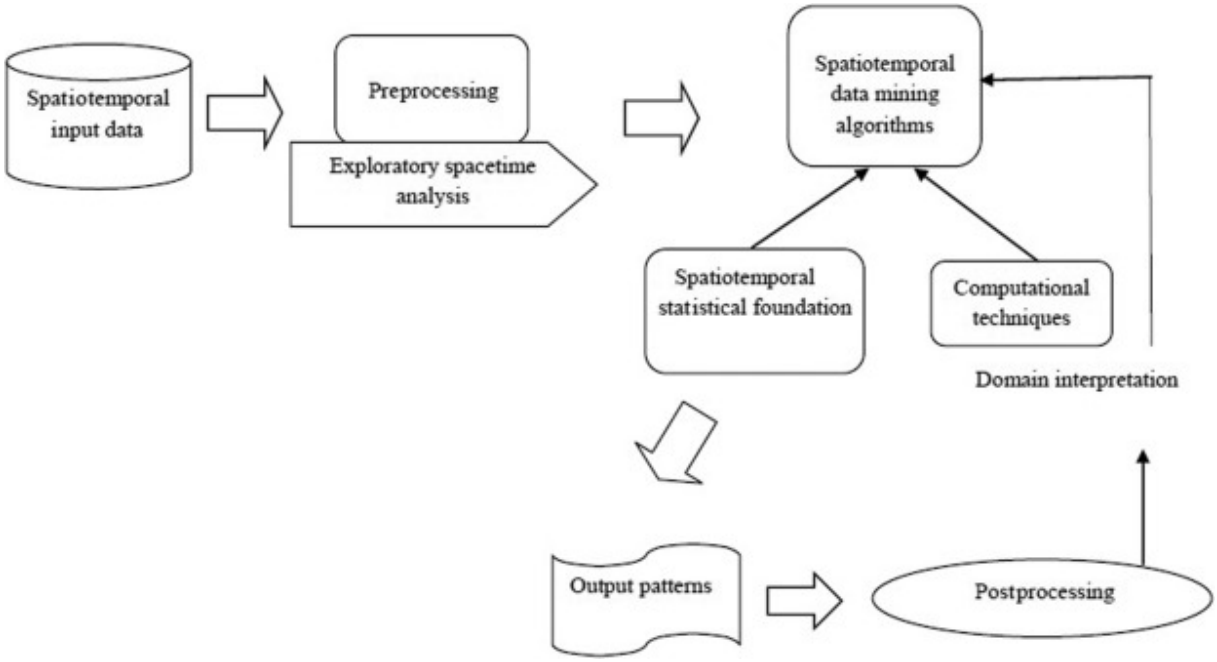

Figure 5. Spatial data analysis process (Kotawadekar, 2021)

Another example where geolocation data is helpful is that spatial learning was used in forest wildfire prediction (Subramanian and Crowley, 2018). In this case, fire was in the landscape at one place, while a set of fire actions might take from that position anytime, and could spread east, west, north, south or not spread at all. This framework was applied to invert dynamic usual setup using the Markov Decision Process for immediate wildfire spread. Furthermore, the predictive patterns were learned by complicated spatial processes. Additionally, the classification of correct data was compared with satellite and other related data. Then, the behaviour of five following approaches were used to examine this issue, including value iteration, policy iteration, Q-learning, Monte Carlo Tree Search, and the Asynchronous Advantage Actor-Critic.

In the next session, articles exploring common patterns of location spoofing data will be reviewed.

2.4 Suspicious signs of location spoofing

The most common variables to determine a location are latitude, longitude, altitude, speed, and accuracy. Latitude and longitude are the coordinates of the location. Accuracy means how accurate the location is. Altitude is the height of the location in relation to sea level. According

to (App Ninjas, 2017), the real GPS would continuously attempt to pinpoint a device's location. As a result, latitude, longitude, altitude, and accuracy change slightly almost every second as the figure 6 demonstrated. When the user is moving, speed will also be changed (figure 7). They revealed that on the other hand, most fake GPS software change only latitude and longitude values, while altitude, speed, and accuracy remain the same even in stationary moving mode (figure 8). This pattern can be used to detect when the gps is spoofed. However, so far, there is an app that can make those values look real as the figure 9 and figure 10, which is GPS JoyStick. Because of the rapid development of technology, more and more applications have the same functions will appear and the pattern is no longer effective enough to detect fake gps cases.

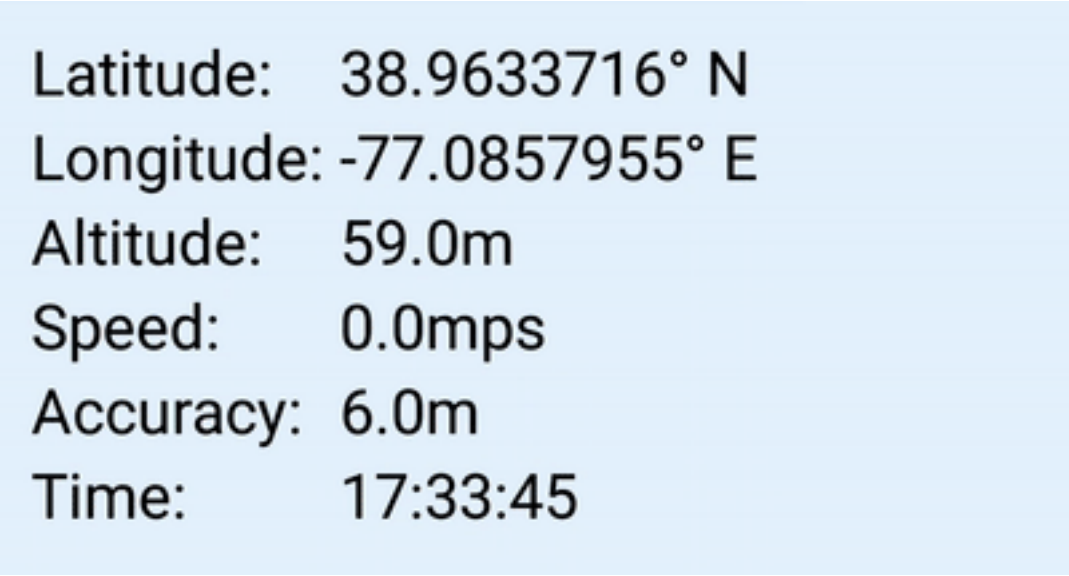

Figure 6. Device GPS While Stationary

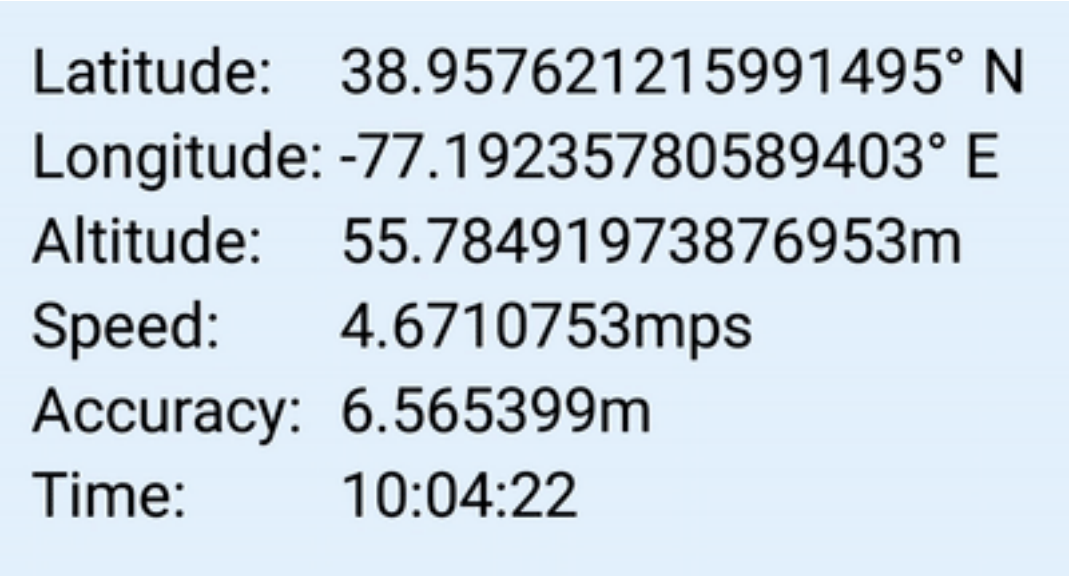

Figure 7. Device GPS While Moving

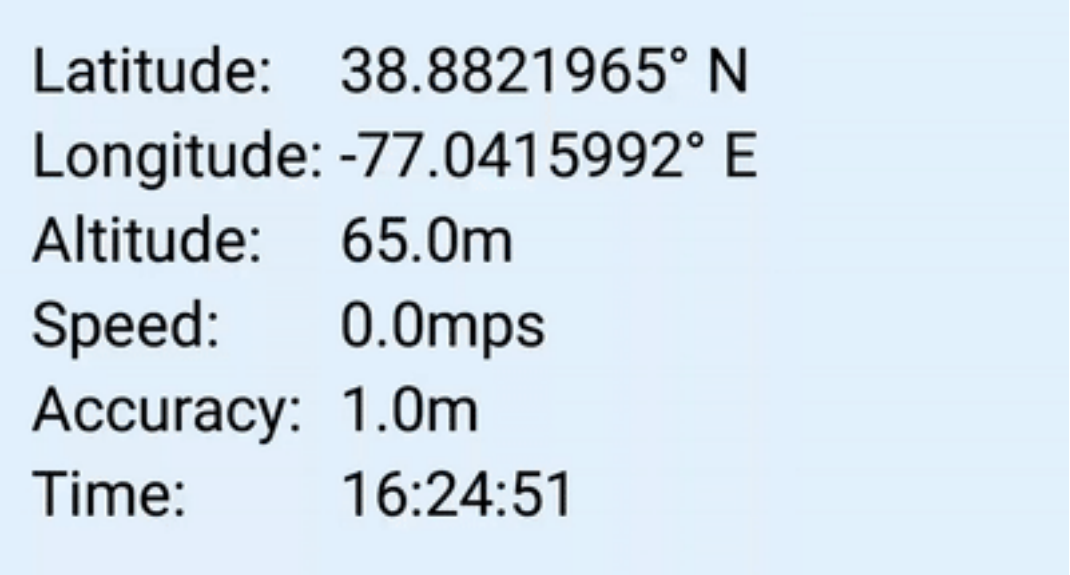

Figure 8. Typical Spoofing App While Stationary / Moving

Figure 9. GPS JoyStick While Stationary

Figure 10. GPS JoyStick While Moving

Similarly, other pieces of data were used in Wong & Yiu's investigation in 2020, a behavioral detection approach using the accelerometer and gyroscope commonly attached in most of the mobile devices. They verified whether the GPS location was real or not by comparing with the direction extracted from gyroscope data and examined the number of steps collected from accelerometer in order to confirm if the movement' step length aligns with the approximate length of adult's steps. However, this would only work when the GOJEK Driver app and users' devices are able to collect accelerometer and gyroscope data correctly.

Table 3. Non-Machine learning solutions using patterns to detect spoofing applications

Citation	Technique	Evaluation
(Wong and Yiu, 2020)	behavioral detection approach using the accelerometer and gyroscope commonly attached in most of the mobile devices	Only work when accelerometer and gyroscope data are available.
(App Ninjas, 2017)	Use pattern of latitude, longitude, altitude, speed, and accuracy changing	This method will not work when gps is spoofed by apps that can make latitude, longitude, altitude, speed, and accuracy look real.

2.5 Application of machine learning in geolocation

Thanks to the development of technology, computer systems can learn historical data to draw insights from patterns in data and predict the future (Shinde and Shah, 2018). That machine learning is now applied in many cases, including geolocation.

For example, in the research of Pandey et al. (2021), machine learning technology was used in a recommender software to suggest people nearby places that they might be interested in. The system detected the user's current location, did some calculation and gave results based on the historical data in the database, including the user's profile and their interest topics. Harvesine formula was used to calculate the distance between the user and places. Then, it showed people places near them that they may want to go to (as figure 11). Big companies providing services have used recommendation engines to encourage people using their services more, including Amazon, Netflix and eBay by providing them information and suggesting favourite products to customers who usually even do not know what they want or need. As the number of people using the internet increases, better systems are needed to do better analysis and provide useful recommendation so companies can serve their customers better and people will have more choices to go to.

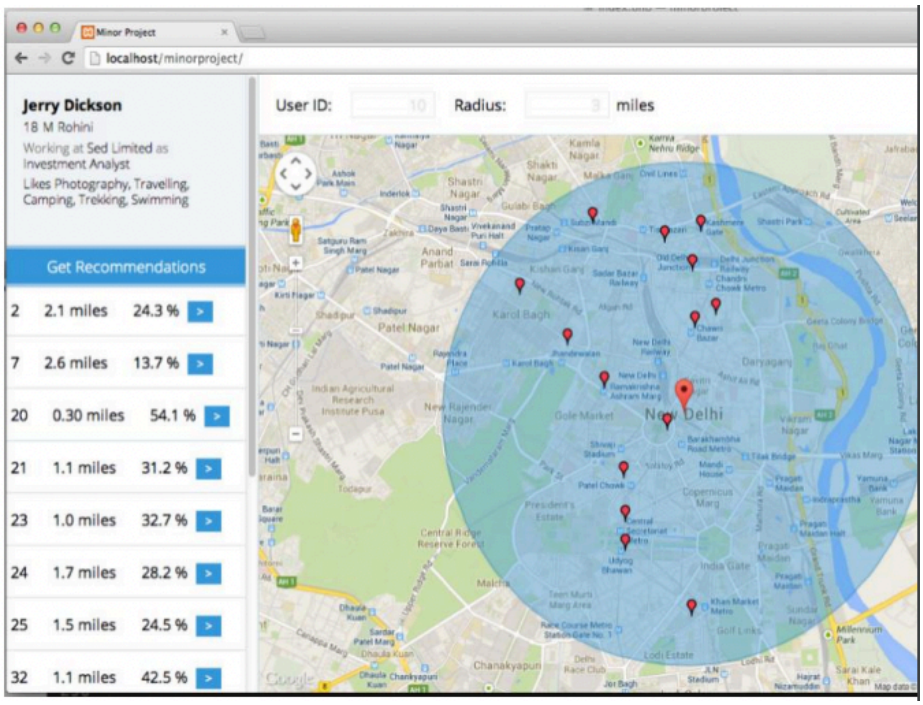

Figure 11. Geolocation Based Recommender System (Pandey et al.,2021)

Sudheer et al. (2022) introduced a new hybrid model using K-nearest neighbors (KNN) algorithm to predict house sales. The model was developed using popular machine learning regression models and determined the value of the land according to geographical features. KNN had better performance where non-linear relationship existed between features in figure 12 and target variable – the sale price.

Variable	Description
zipcode	The 5-digit zipcode of the property
lat	Latitude
long	Longitude
sqft_liv15	Average size of interior housing living space for the closest 15 houses, in square feet
sqft_lot15	Average size of lot area for the closest 15 houses, in square feet
waterfront	A binary coded feature on whether the property is alone the waterfront or not
view	A metric that gauges how good the view of the property is from 0 to 4

Figure 12. Geographical features of the dataset (Sudheer et al., 2022)

From their analysis, compared to the price variable, price per square feet is a better target because price/sqft (square feet) described the land value more accurately and thus, could help to compare different prices between locations easily. Just using price as the target would not effectively predict the value of the land because some areas are larger than others, while there were some small apartments located in city centers or near the beaches, which can cause them more expensive. Figure 13 shows how effective zipcode was used to predict price and price per square feet. The darker regions cost much more money compared to the lighter regions. In the left image, there are not many differences between lands in different areas. On the other hand, the right image presents that lands overlook the beaches (the blue color in the map) are more expensive than those not surrounded by the water, which is more accurate to predict the land value.

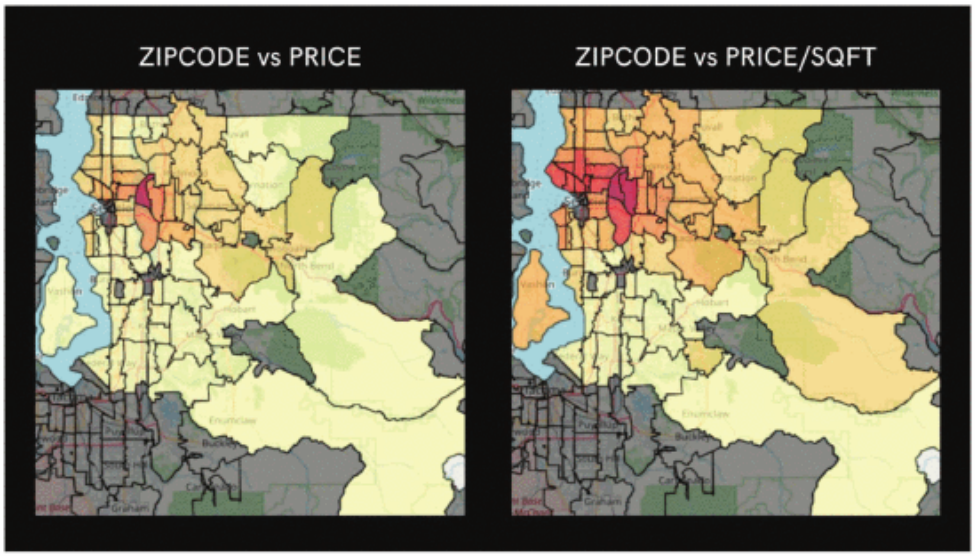

Figure 13. Zipcode vs price mapping

2.6 Using Machine Learning in location spoofing detection

There are several papers investigated how machine learning can be used to detect location spoofing.

To detect spoofing location in social media, Zhao & Sui (2017) developed a Bayesian time geographic method using millions of geotagged tweets. They collected about three million geolocation tweets from Twitter's application programming interface (API) within two days and stored information like who sent the tweet, where and when it was and other attributes of the user. To detect a fake geo-tweet, they adopted the filter and refine strategy (Wood, 2008) as illustrated in figure 14. To be specific, firstly, a space-time (ST) cone was used to detect the fake paths which exceeded the human moving speed. Then, they calculated the probability the user tended to visit places by a Bayesian time geography model. It is more likely that the fake location had a lower visiting probability. This method was tested with millions of geo-tweets, indicating that their hybrid approach was able to detect spoofed location on social media posts although fake location tweets only took a small percentage of the total tweets. The limitation is that it needs the user to continuously post their tweets sharing their location so it can detect those violating normal velocity constraints.

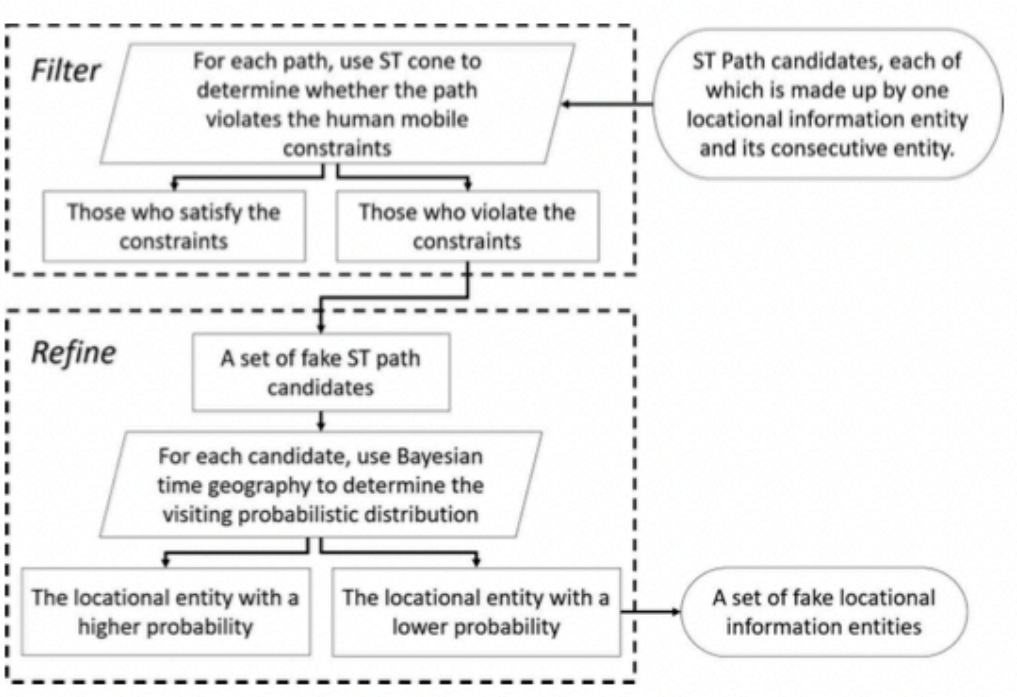

Figure 14. Flow Chart for detecting fake geolocation (Wood, 2008)

Jiang et al. (2021) proposed a deep learning model, called DeepPose to realise GPS spoofing attempts on mobile environments using motion sensor data. Neural network was used to eliminate the noise, recovering a vehicle's path from several sensor inputs, then suggested a new strategy to predict the real destination from sensor readings on the Google Map. Then, the reconstructed path is used for GPS spoofing detection. Figure 15 demonstrates the GPS spoofing on vehicle movement. The advantage of using motion sensor data establishing the real trajectory is that spoofers can hijack GPS signals of nearby devices using Opensource Software-Defined GPS Signal Simulator (Ebinuma, 2022), but cannot alter motion sensor data. As a result, those data can be used to recognise vehicle speed and direction and calculate the vehicle position then compare with the reported GPS location to point out spoofing attacks. The model was trained and evaluated using two datasets from real driving situations, BDD100K, a Large-scale Diverse Driving Video Database (Yu et al., 2020) and from Jiang et al. (2021) in Norfolk, Virginia. From their observation, after reaching 30% of the road, the vehicle was detected correctly with 80% of accuracy and false alarm was less than 8%. The accuracy was improved as the vehicle moves further on the street. Their model could detect spoofing attacks quickly when it happened.

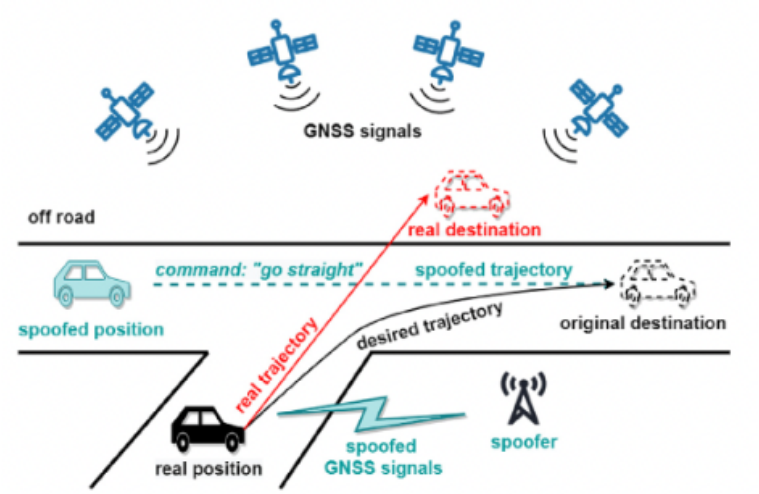

Figure 15. GPS spoofing on vehicle (Peng et al., 2021)

To determine whether an order was done with fake GPS apps or not based on the drivers' ping behaviors in a GOJEK public dataset, Mar'i & Pangestu (2021) proposed using a machine learning algorithm called Logistic Regression. To reach the best result, data was pre-processed first by calculating distances and time between pings. Their model reached a precision value at 74.7%, recall at 99.4%, an accuracy score of 74.7% and F1 scored 85.5%. Although recall had a high value, it would also be better to have a higher specificity and accuracy score because in the GOJEK case, the aim is to have the rate of actual fake GPS orders (negative cases) correctly predicted as highest as possible.

Table 4. Machine learning solutions to detect spoofing applications

Citation	Technique	Pre-processing	Modelling	Evaluation Metrics
(Jiang et al., 2021)	Deep learning model, called DeepPOSE to realise GPS spoofing attempts on mobile environments	Neural network was used to eliminate the noise, recovering a vehicle's path from several sensor inputs	A new strategy to predict the real destination from sensor readings on the Google Map. Then, the reconstructed path is used for GPS spoofing detection.	after reaching 30% of the road, the vehicle was detected correctly with 80% of accuracy and false alarm was less than 8%. The accuracy was improved as the vehicle moves further on the street.
(Mar'i and Pangestu , 2021)	Logistic Regression	Calculate distances and time between pings	A machine learning algorithm called Logistic Regression	Precision value at 74.7%, recall at 99.4%, an accuracy score of 74.7% and F1 scored 85.5%
(Zhao and Sui, 2017)	Bayesian time geographic method using millions of geotagged tweets	Collected three million geolocation tweets from Twitter's API within two days and stored information like who sent the tweet, where and when it was and other attributes of the user.	- Adopted the filter and refine strategy. First, an ST cone was used to detect the fake paths which exceeded the human moving speed. Then, they calculated the probability the user tended to visit places by a Bayesian time geography model. It is more likely that the fake location had a lower visiting probability. - Tested using million geotagged tweets	Applied the nearest neighbour analysis to geotag samples by randomly selected 900 geotags. Their z-scores were from –42 to –48, meaning that all the listed geotag groups were clustered. Among all Nearest Neighbour Index values, the fake geotag sample had the greatest value. This meant that the fake geotags, though spreading in wider area, were less clustered than the spatial pattern of normal geotags.

2.7 Discussion

After reviewing several papers, it can be seen that there are basically two methods to spoof GPS location. Firstly, GPS signals can be interfered to transmit GPS coordinates inaccurately to the target. Secondly, GPS location of a device can be faked using fake GPS applications. There are a number of research proposing solutions for the first one, but not many papers suggesting solutions for the later, especially location spoofing in a short distance like the GOJEK case. Those spoofing applications can be detected by using methods in GOJEK application on mobile devices. However, those methods are still not perfect and not be able to detect all the spoofing cases. Furthermore, as the technology always develops, spoofers will try their best to invent new spoofing methods. Because of this, this paper examines several possible methods based on the previous research and suggest the best one to detect fake location on mobile devices.

Previous case studies showed that geolocation data could be analysed to find patterns and helped a lot in a variety of areas. Furthermore, there are differences between geolocation data of real GPS location and fake GPS location (App Ninjas, 2017). Those studies have shed a light on GPS location spoofing detection. To detect fake GPS location, patterns of those geolocation can be defined and then be used to mark the suspicious ones. This can be done by looking at the data and selecting the conditions manually or automatically using machine learning.

When it comes to big data, manual work would take a lot of time. Willem Pienaar, Data Science Platform Lead at GOJEK revealed that operating in 167 cities in Indonesia, GOJEK has more than 1 million drivers using the application and each day, they have hundred thousand drivers online and serving customers at the same time. Ajey Gore, Group Chief Technology Officer at GOJEK added that they ping each drivers every 10 seconds. In total, there are approximately eight billion pings per day and six million pings per minute (Google Cloud, 2022). Due to the huge number of data GOJEK have, around 5TB data generated daily, manual classification cannot be done. Here, machine learning can play an important role in detecting fake locations because it would reduce the manual work and can be updated frequently to deal with the ongoing spoofing development.

In conclusion, geolocation spoofing is mentioned a lot in recent papers, but not many of them proposing solutions solving fake GPS on mobile devices like the GOJEK case. Fake GPS

techniques will become more and more complicated and advanced. Manually categorising them would not be a good solution in a big data world. As a result, this paper is going to recommend a machine learning model to deal with this situation.

2.8 Summary

This chapter looks at the past research papers to learn more about location spoofing and to identify which method should be the best to investigate more. In general, location can be spoofed by interfering the signals transmitted to the GPS devices and by using fake location services on mobile devices. The first method has been tackled in a number of papers, but the second was not. By investigating further, it is found that geolocation data of the fake location has common patterns compared to the normal ones. There were previous papers using this as a sign to find suspicious location. Some of them used machine learning algorithms. Due to the high traffic data of GOJEK and the fast innovation of technology, machine learning would be a good solution for GOJEK case. Therefore, this paper will examine several machine learning models to find the best one and optimise it for GOJEK to detect spoofing.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This chapter will talk about the research methodology to see how machine learning is going to be applied in the GOJEK case. Methodology consists of selecting data, pre-processing the dataset, transforming data into a structured and comprehensible, balancing the dataset, applying machine learning techniques to build models and evaluating the models using evaluation measures. These steps are described in the figure 16.

Figure 16. Flow chart of the implementation

3.2 Proposed Method

3.2.1 Data Selection

Data used in this research is an anonymised dataset on Kaggle (2019) of PING signals from GOJEK partner drivers within 47 days.

3.2.2 Data Preparation (Preprocessing and Transformation)

To begin, the data should be cleaned up from the dataset as the flow figure 17.

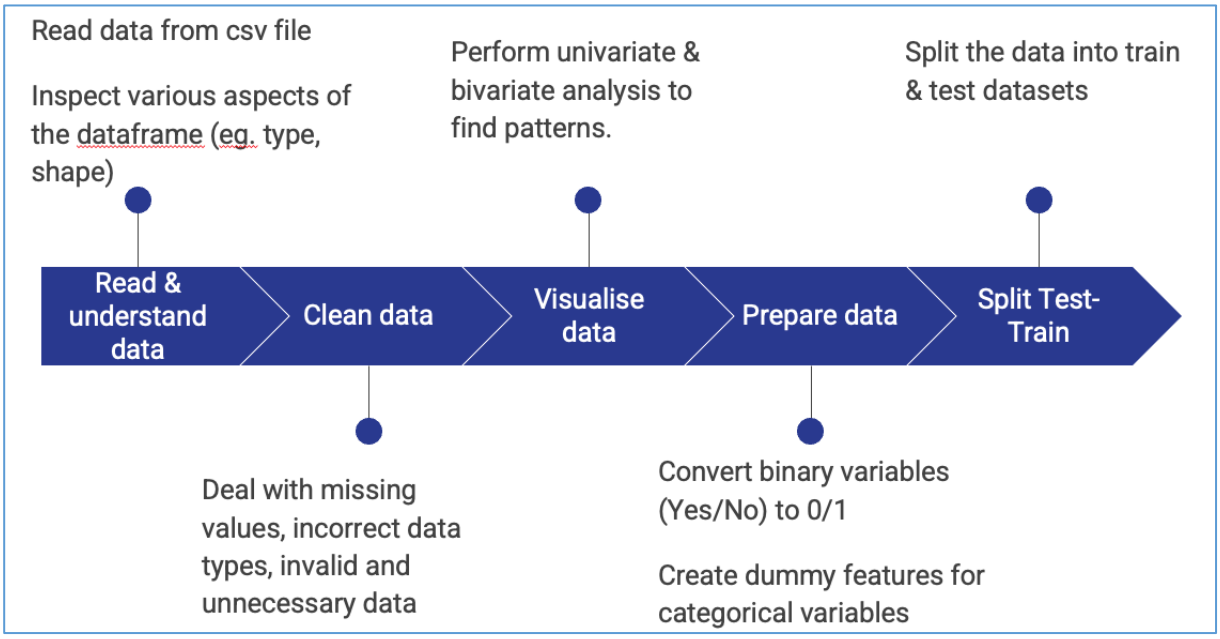

Figure 17. Flow chart of data preparation

After reading the data from the comma-separated values (CSV) file, missing values, incorrect data types, invalid and unnecessary data will be dealt with. Those types of data will be examined to see if they are associated with the target column. If so, they can be replaced using simple imputation approach. In a survey paper published in 2021, Emmanuel et al. (2021) mentioned that most of the papers used simple imputation to handle missing data by mean, median or mode of the variables. Otherwise, when the columns with missing data do not have any impact on the target, they can be eliminated with pairwise deletion which is known to produce low bias results for both missing completely at random and missing at random data.

Then, data will be visualised using Exploratory Data Analysis techniques to find patterns by comparing differences between fake and real orders. As suggested by (Salgado et al., 2016), line plot and histograms will be used to explored univariate continuous and categorical variables respectively. On the other hand, scatter plot is suitable to visualise relationship between variables.

Next, new features that may fit better on data modelling can be created from existing variables.

- 'Date' attribute will need to be changed. Here, if the numbers from one to seven are used for the days of the week from Sunday to Saturday, day 1 (Sunday) and 7 (Saturday) are very far to each other, but they should not be because Sunday is near Saturday.

Dossman (2017) revealed that such cyclical feature should be encoded using sin/cos transformation so that the model can use it effectively as presented in the equation (1) and (2) below.

$$day\_cos = \cos\left(2 \times 3.14 \times \frac{day(numeric)}{7}\right) \quad (1)$$

$$day\_sin = \sin\left(2 \times 3.14 \times \frac{day(numeric)}{7}\right) \quad (2)$$

Besides that, a new variable called 'is_weekend' will be created and use as an indicator of weekdays and weekend.

- Similarly, 'hours' attribute will be encoded using sin/cos transformation as the equation (3) and (4).

$$hour\_cos = \cos\left(2 \times 3.14 \times \frac{hour}{24}\right) \quad (3)$$

$$hour\_sin = \sin\left(2 \times 3.14 \times \frac{hour}{24}\right) \quad (4)$$

Furthermore, 'hours' will be transformed to is_working_hour, which will have '1' in the value when the 'hours' is between 9 am and 6 pm. Otherwise, it will be '0'.

- 'Seconds' is in Linux format; thus, it will be converted to datetime format and will be compared with the 'Date' column to see if they are the same and if 'Seconds' has any meaning or is useful for the model.

- 'Latitude' and 'longitude' will be used to calculate the distance between two consecutive pings with the "Haversine" method, according to Hijmans (2021), working best for small distances.

At the final step of this stage, training data are split into train and test data before building models. Train data will be used to train the models, while test data will be for models testing and evaluations.

3.2.3 Modelling Techniques

After cleaning data, models will be built as the following chart. Before applying any algorithm, feature scaling is important in machine learning to bring all numeric variables in the same standard, so that one significant value would not have any impact on the algorithm because of their magnitude (Baijayanta, 2020).

Then, the model will be formed using Logistic Regression. Peng et al. (2002) pointed out that Logistic Regression can figure out how dependent variable depends on independent variables. This algorithm collects data into different groups and each group represents a class of each data type. The data will then be divided into a variety of categories by a logit function like the equation (5).

$$logit(x) = \log\left(\frac{x}{1-x}\right) \qquad (5)$$

Another model that can be built is Decision Tree. This method divides a dataset into branch-like segments that construct a tree with nodes and branches (Song & Lu, 2015). It starts with a root decision node, askes a Yes/No question and expands into many subtrees. This method is easy to understand as it is similar to what human usually do when making decisions. In fact, it compares the attributes of a record in the dataset with the values of decision node and based on the comparison, jumps to the next node by following the branch. At the next node, the algorithm compares the value of the attribute with the other nodes again and move further. The process is continued until the leaf node of the tree is reached. The diagram in figure 18 is a graphical illustration of a Decision Tree Algorithm.

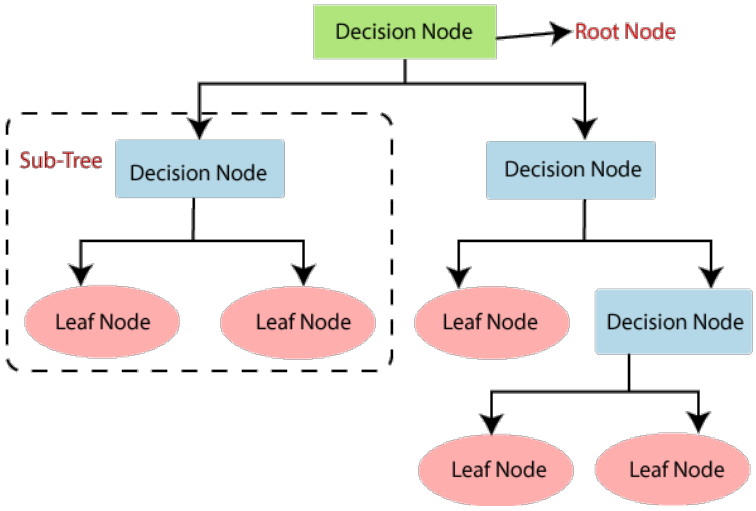

Figure 18. Decision Tree (Arain, 2021)

A decision tree with a number of layers can lead to overfitting issues, which can be solved using Random Forest Classifiers, an ensemble classification algorithm consisting of many decisions trees (Fawagreh et al., 2014). Each tree acts as a classifier to define the label, then the label which has the highest number of votes is used to decide the label of the instance as illustrated in figure 19.

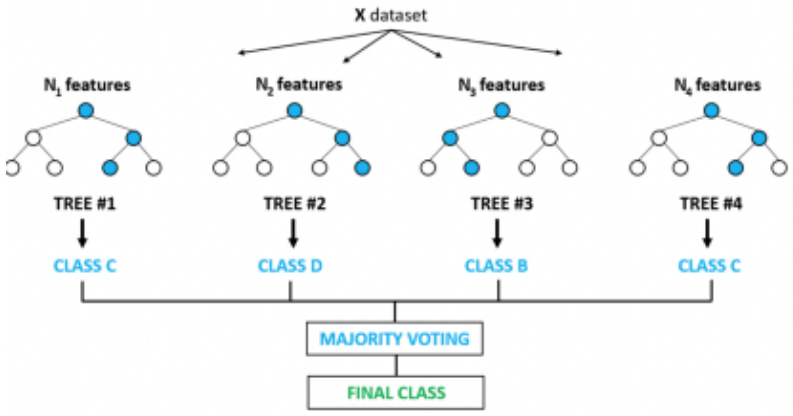

Figure 19. Example of ensemble of decision trees (random forest) (Kirasich et al., 2018)

3.2.4 Evaluation

To compare the models, first, a confusion matrix is needed, to see how the system mislabels the true and fault value of the predicted variable (Powers, 2020). The matrix, as in figure 20, shows the number of cases where real GPS data predicted correctly (true positive), the number of cases

where the GPS data is real, but the system predicted as fake (false negative), the number of cases where the fake GPS data is predicted correctly (true negative) and the number of cases where the GPS data is fake but the system predicted as real (false positive). Then, these four variables will be used to calculate the accuracy, specificity and recall to evaluate the performance of the models.

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Figure 20. Confusion matrix (Wikipedia, 2022)

Accuracy is used to calculate the correct records compared to the total records as equation 6. As a result, a high accuracy value proves that the classification prediction is good.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \qquad (6)$$

Recall (true positive rate) is the proportion of positive cases predicted correctly as positive, calculated by the equation 7. A high recall value means that there is a high ratio of real gps cases predicted correctly.

$$Recall = \frac{TP}{TP+FN} \times 100\% \qquad (7)$$

In contrast, specificity (true negative rate) is the proportion of real negative cases predicted as negative correctly, calculated by equation 8. The higher the specificity value, the more fake gps cases are predicted exactly.

$$Specificity = \frac{TN}{N} = \frac{TN}{TN+FP} = 1 - FPR \qquad (8)$$

Cross-validation (CV) will also be applied to evaluate models. By partitioning the dataset into train and test sets, the number of samples using for model learning has been reduced, and the results depend on a particular random choice for the pair of the sets. A procedure called cross-validation is a solution to solve this problem. In this approach, using k-fold CV, there are k splits. In each split, the training set is split into k smaller sets. The model is trained using k – 1 of the folds as training data. The remaining part of the data is used to validate the resulting model. The performance score is the average value of all the scores computed in each split. Although this can be expensive, with this way, all the data can be used to both train and validate models. Figure 21 adopted from Scikit-learn page - Cross-validation: evaluating estimator performance (2022), illustrates the cross validation procedure.

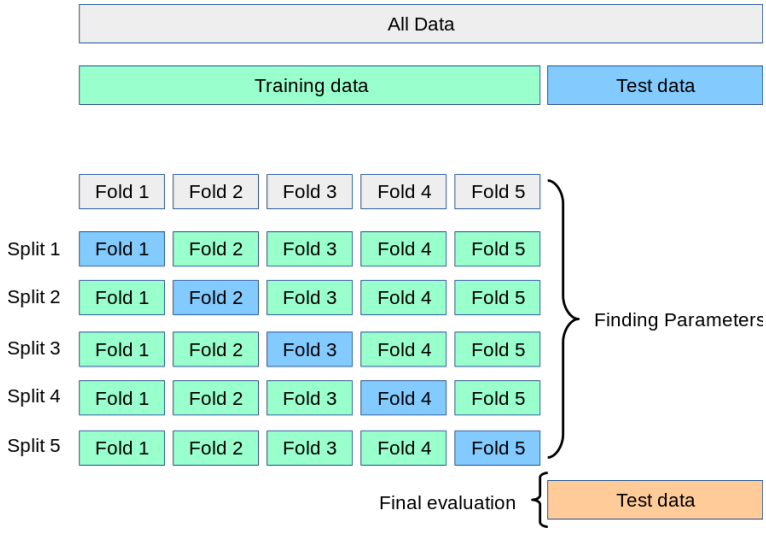

Figure 21. Cross validation procedure (Scikit-learn, 2022)

Finally, the model with high score, recall, specificity and accuracy will be selected as the purpose of the study is to have the rate of actual fake GPS pings correctly predicted as highest as possible. The evaluation flow is shown in the figure 22 below.

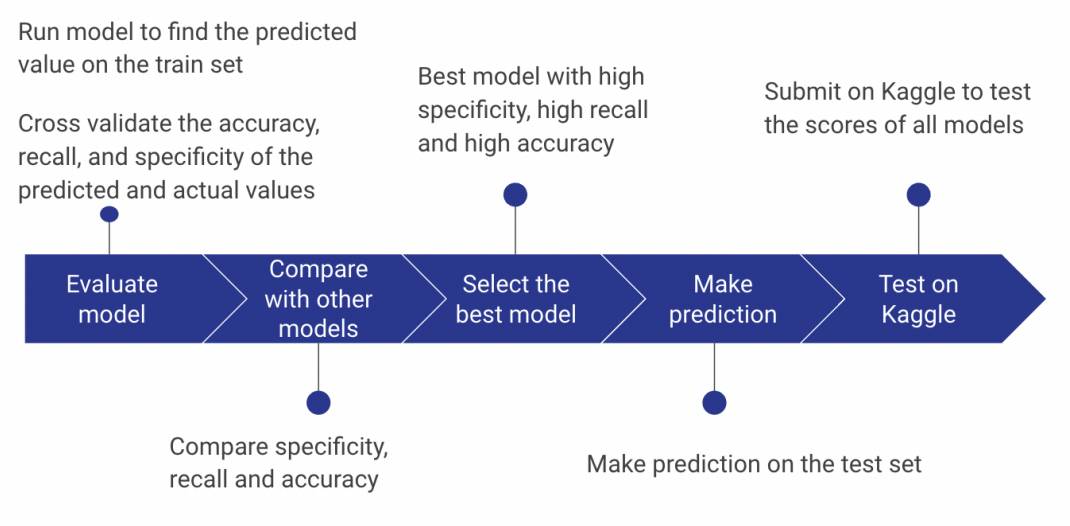

Figure 22. Evaluation flow chart

3.3 Summary

This third chapter described the methodology used in this research. Data used is an anonymised dataset on Kaggle (2019) of PING signals from GOJEK partner drivers. After cleaning data, univariate and bivariate analysis will be performed to see the relationship between variables. Then, new features, such as day_cos, day_sin, is_weekend, is_working_hour, distance will be created to support the model. After that, data will be split, and different models will be built, like Logistic Regression, Decision Tree and Random Forest. Finally, those models will be evaluated with accuracy, recall and specificity. The one with high recall, specificity and accuracy will best fit the purpose as the GOJEK case needs true GPS pings and fake GPS pings correctly predicted as highest as possible.

CHAPTER 4: ANALYSIS

4.1 Introduction

This chapter describes the analysis steps which have been done to support the research. First of all, the dataset was read and examined. Then, the data was prepared and visualised to perform Exploratory Data Analysis. After variables being transformed and done with feature engineering, significant features were selected using RFE and different models were built. Finally, the models were evaluated using confusion metrics and cross validation methods to select the best model.

4.2 Dataset Description

The dataset contains two files. The train dataset has 567545 rows (called either observations or pings) and 11 features, containing numeric feature variables, a string order_id column, and a binary target column. The test dataset has 81334 rows and 10 features, which are similar to the train dataset, but the label is excluded. Table 5 and table 6 show the first five rows in the train and test datasets respectively. Table 7 displays the data type of each variable.

Table 5. First five rows in the train dataset

	order_id	service_type	driver_status	date	hour	seconds	latitude	longitude	altitude_in_meters	accuracy_in_meters	label
0	RB193	GO_RIDE	UNAVAILABLE	2018-02-05	6	1548890667	-6.922910	107.631301	NaN	23.027	0
1	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890680	-6.923039	107.631250	712.0	9.577	0
2	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890690	-6.923039	107.631250	712.0	9.577	0
3	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890700	-6.923048	107.631230	713.0	8.139	0
4	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890710	-6.922968	107.631253	713.0	7.029	0

Table 6. First five rows in the test dataset

	order_id	service_type	driver_status	date	hour	seconds	latitude	longitude	altitude_in_meters	accuracy_in_meters
0	F1770	GO_FOOD	AVAILABLE	2018-02-05	18	1548932409	-6.923841	107.700414	NaN	5.0
1	F1770	GO_FOOD	AVAILABLE	2018-02-05	18	1548932419	-6.923841	107.700414	NaN	5.0
2	F1770	GO_FOOD	AVAILABLE	2018-02-05	18	1548932429	-6.923841	107.700414	NaN	5.0
3	F1770	GO_FOOD	AVAILABLE	2018-02-05	18	1548932439	-6.923841	107.700414	NaN	5.0
4	F1770	GO_FOOD	AVAILABLE	2018-02-05	18	1548932449	-6.923841	107.700414	NaN	5.0

Table 7. Types of all variables

```
RangeIndex: 567545 entries, 0 to 567544
Data columns (total 11 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   order_id             567545 non-null  object
 1   service_type         567545 non-null  object
 2   driver_status        567545 non-null  object
 3   date                 567545 non-null  object
 4   hour                 567545 non-null  int64
 5   seconds              567545 non-null  int64
 6   latitude             567545 non-null  float64
 7   longitude            567545 non-null  float64
 8   altitude_in_meters   413142 non-null  float64
 9   accuracy_in_meters   567545 non-null  float64
 10  label                567545 non-null  int64
dtypes: float64(4), int64(3), object(4)
memory usage: 47.6+ MB
```

Features description:

- order_id - an anonymous id unique to a given order number. Each order_id (ride) includes several observations ("pings"). Order_id starting with an "R" corresponds to a "GORIDE" service and those starting with a "F" are a "GOFOOD" service.

- service_type - service type, can be GORIDE or GOFOOD. The data in both train and test data are equally distributed between the service types.

- driver_status - status of the driver PING, can be AVAILABLE, UNAVAILABLE, OTW_DROPOFF, OTW_PICKUP.

- date – The data were collected for 47 consecutive days, from 2018-02-05 to 2018-03-23 for train data and from 2018-02-05 to 2018-03-22 for test data.

- hour – time in hours

- seconds – time in seconds (in linux format)
- latitude - GPS latitude
- longitude - GPS longitude
- altitude_in_meters - GPS altitude in meters
- accuracy_in_meters - GPS accuracy in meters, the smaller the more accurate

Target:

- label - label describing whether GPS is true (1) or fake (0)

Table 8 shows a statistical summary of the dataset, including mean, standard deviation, minimum, 25-50-75th percentiles and maximum value of altitude and accuracy in meters. The average accuracy value was 31 meters, most of pings had accuracy value around 10 meters while there was a place which was very less accurate, with more than 9,000 meters of accuracy. The mean value of altitude was 745 meters and 75% of pings had altitude 790 meters, which is appropriate to the average altitude in Bandung (Indonesia) 768 meters above the sea (Bandung, 2022).

Table 8. Numeric columns description

	altitude_in_meters	accuracy_in_meters
count	413142.000000	567545.000000
mean	744.885056	31.031366
std	105.357303	199.671686
min	-1075.000000	0.010000
25%	708.300000	0.500000
50%	748.400000	4.133000
75%	790.481659	10.380000
max	4674.877000	9628.102000

4.3 Data Preparation

4.3.1 Handling missing data and columns with value 0

There is only one column having null value and one column having value zero, which are 'Altitude_in_meters' and 'hour'. 'Altitude_in_meters' variable is missing in 154,403 rows which is 27.2% of the dataset. Since GoJek provides the service for a short distance transportation only (within a city), altitude is not very important to distinguish between different places. Therefore, 'altitude_in_meters' variable was removed from the dataset. Except the 'label' variable, 'hour' is the only column having value 0 with 7233 rows. This means 7233 orders were requested at mid-night.

4.3.2 Handling date time columns

'Seconds' variable is in the Linux format (1548890667). The in-built function *utcfromtimestamp* of datetime was used to convert it into datetime format. However, after converting, the Linux date time did not match the 'date' and 'hour' variables as the table 9. Therefore, the Linux date time was not useful and was removed.

Table 9. Linux date in the dataset after converting from 'Seconds'

	order_id	service_type	driver_status	date	hour	seconds	latitude	longitude	accuracy_in_meters	label	first_letter_order_id	linux_date
0	RB193	GO_RIDE	UNAVAILABLE	2018-02-05	6	1548890667	-6.922910	107.631301	23.027	0	R	2019-01-30 23:24:27
1	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890680	-6.923039	107.631250	9.577	0	R	2019-01-30 23:24:40
2	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890690	-6.923039	107.631250	9.577	0	R	2019-01-30 23:24:50
3	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890700	-6.923048	107.631230	8.139	0	R	2019-01-30 23:25:00
4	RB193	GO_RIDE	AVAILABLE	2018-02-05	6	1548890710	-6.922968	107.631253	7.029	0	R	2019-01-30 23:25:10

'Date' variable was converted to the day of the week. For example, February 05, 2022, is Monday, which is the start of the week, so the day of the week is 0.

4.3.3 Creating new variables related to location, time and velocity of each status of one ride

One ride had four statuses, including available, pick up, drop off and unavailable. In each status, speed, location, and time differences were examined between the first and the last pings to detect abnormal behaviours of the drivers.

First of all, the distance and time difference between two consecutive pings were calculated. The distance was defined by applying the haversine function to the latitudes and longitudes of the two consecutive pings. Using the time when one ping occurred and the coordinates where it was located at, the driving velocity was calculated by dividing the distance between the two pings by the time difference between them. This is to determine if there was any location jumping in the drivers' movement or not. If the speed was high and was not as appropriate as the normal riding speed, it could be that the driver changed their locations using a fake location application.

Next, the time drivers spent on each status would also be considered. The data was grouped by order and driver status to calculate the start date, start time, end date, end time and their sin/cos transformations of each status. With the start time and start date, new variables called 'is_weekend' and 'is_business_hour' were created. A ping was categorised as "is weekend" if the start day was greater than four and 'is business hour' when it happened between 8am and 6pm. Then, the difference between the start time and end time was calculated to define how many days lasted between the first and last pings of one ride. Figure 23 shows some rides lasted more than one day, which was quite suspicious because normally, GoJek ride around the city should not take too long.

```
# Define how many days last between 1st and last pings in 1 ride
data["days_diff"] = (data["max_date"] - data["min_date"]).astype(str)
```

```
data["days_diff"].value_counts()
```

```
0 days    565096
1 days      2449
Name: days_diff, dtype: int64
```

=> Some rides overlap 2 days

Figure 23. Rides last more than one day

Finally, based on the distance and time difference between two consecutive pings, aggregation of all pings in one status, was utilised to create new variables comparing between the first and the last ping of one ride's status. Those new variables were minimum accuracy, maximum accuracy, mean accuracy, median accuracy, maximum distance, total distance, maximum time difference, total time difference. Total distance and total time difference helped to come up with average velocity of one ride's status. Figure 24 represents a sample of new variables created in this section.

	order_id	driver_status	min_date	max_date	min_hour	max_hour	min_seconds	max_seconds	min_day_of_week	days_diff	max_distance	total_distance
0	RB193	UNAVAILABLE	2018-02-05	2018-02-05	6	6	1548890667	1548890667	0	0 days	NaN	0.000000
1	RB193	AVAILABLE	2018-02-05	2018-02-05	6	6	1548890680	1548892030	0	0 days	0.615576	2.232941
2	RB193	AVAILABLE	2018-02-05	2018-02-05	6	6	1548890680	1548892030	0	0 days	0.615576	2.232941
3	RB193	AVAILABLE	2018-02-05	2018-02-05	6	6	1548890680	1548892030	0	0 days	0.615576	2.232941
4	RB193	AVAILABLE	2018-02-05	2018-02-05	6	6	1548890680	1548892030	0	0 days	0.615576	2.232941
5	RB193	AVAILABLE	2018-02-05	2018-02-05	6	6	1548890680	1548892030	0	0 days	0.615576	2.232941

Figure 24. Sample of new variables

4.4 Exploratory Data Analysis

4.4.1 Visualising Numeric Variables

Numeric variables were visualised to find correlation between them. Figure 25 displays the top 10 correlation pairs. The time (seconds), minimum time (min_seconds) and maximum time (max_seconds) were highly correlated. Similarly, distance and velocity were correlated.

```
Top 10 correlation pairs:

seconds          min_seconds        1.000000
                 max_seconds        1.000000
min_seconds      max_seconds        1.000000
distance         velocity           0.999900
total_distance   avg_velocity       0.999603
longitude_diff   latitude_diff      0.999484
day_of_week      min_day_of_week    0.999018
start_hour_sin   end_hour_sin       0.986019
start_hour_cos   end_hour_cos       0.984331
hour             max_hour           0.978484
dtype: float64
```

Figure 25. Top 10 correlation pairs

Next, fraud and non-fraud data were separated to compare between their numerical features as figure 26 and 27. It can be seen from the two charts that velocity of fraud pings was higher than that of normal pings.

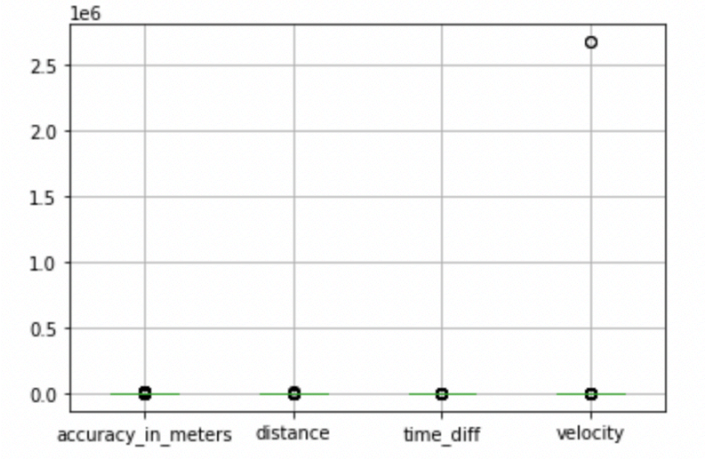

Figure 26. Numerical features of fraud pings

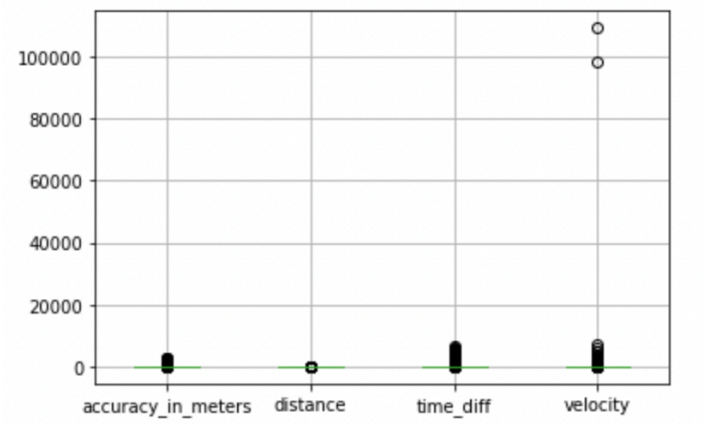

Figure 27. Numerical features of non-fraud pings

4.4.2 Visualising Categorical Variables

Categorical variables were visualised to find the patterns of fraud pings.

Accuracy of fake location pings were higher than that of real ones as shown in figure 28. For real pings, the accuracy was around 10 meters to 30 meters. The maximum accuracy was 45 meters. On the other hand, accuracy of spoofing location pings was between 20 and up to more than 80 meters.

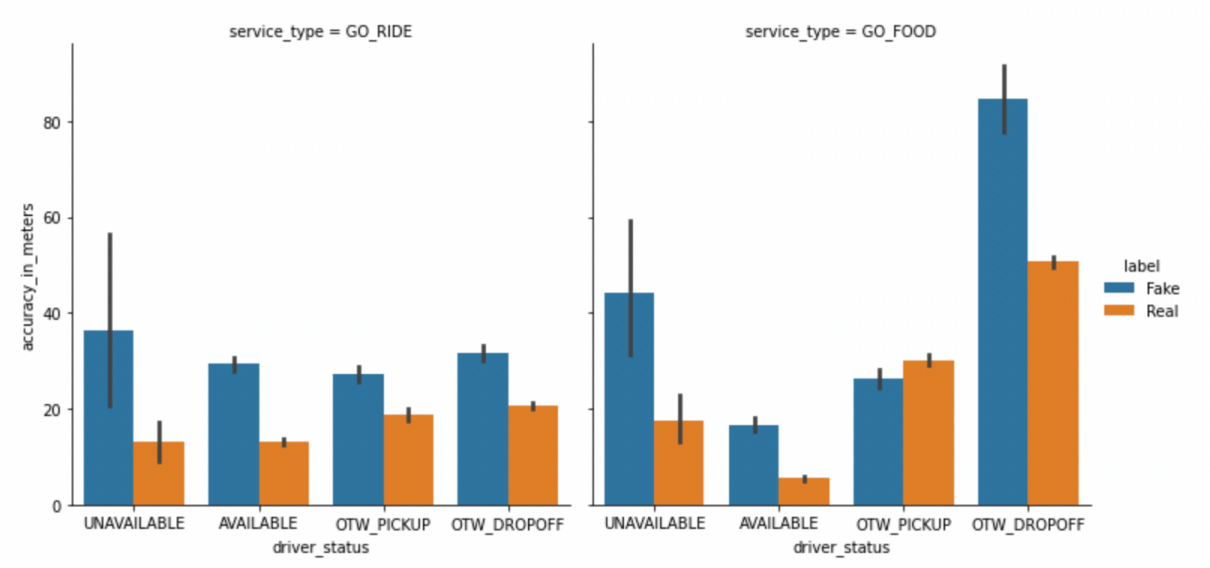

Figure 28. Accuracy between fraud and non-fraud pings

GoJek provided two service type, which are "Go Food" and "Go Ride". The number of pings in "Go Food" was 335,863, which was higher than that of "Go Ride" with 231,682 pings. However, "Go Ride" had more fraud pings than "Go Food". 42.8% of "Go Ride" pings were classified as fraud, while only 18% of "Go Food" pings were fake location. In fact, most of the fraud pings belonged to the "Go Ride" service as the chart in figure 29, which was almost double those "Go Food" pings.

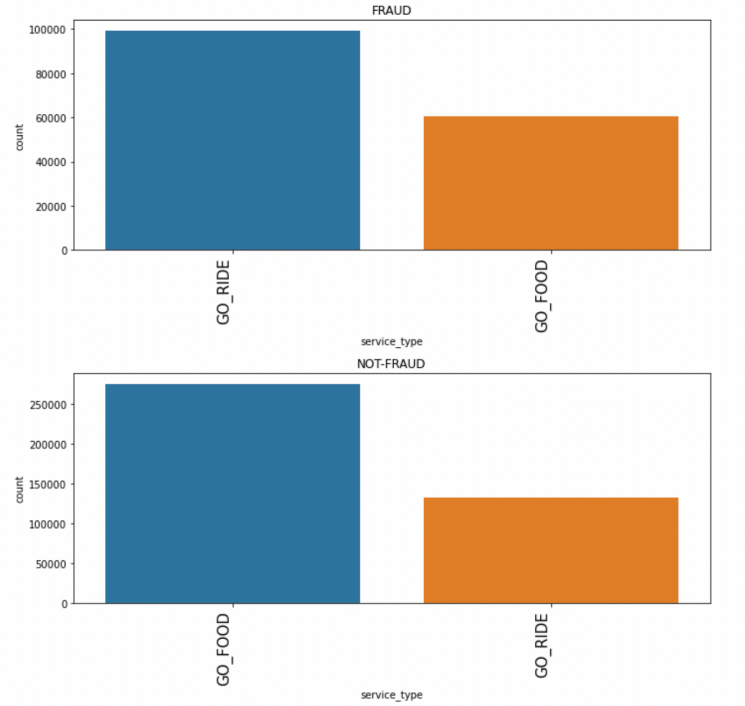

Figure 29. Fraud in different services

In terms of driving statuses as in figure 30, the percentages of fraud and not fraud pings were quite similar. There was also no clear pattern to define fraud in each status of each service although the “GORIDE” service had more time “available” than the “GOFOOD” service.

Figure 30. Fraud in different statuses

4.4.3 Visualising on map

Rides are plotted on map with Folium, a Python library to map geospatial data, to see the movement patterns between fraud and real orders. The circles with colour red, green, black, and blue represent the pings with unavailable, available, pickup and drop off statuses respectively. From the four maps in figure 31, 32, 33 and 34, it can be seen that real orders had the pattern of movement: Available -> Pick up -> Drop off -> Available, but fake orders did not follow that pattern.

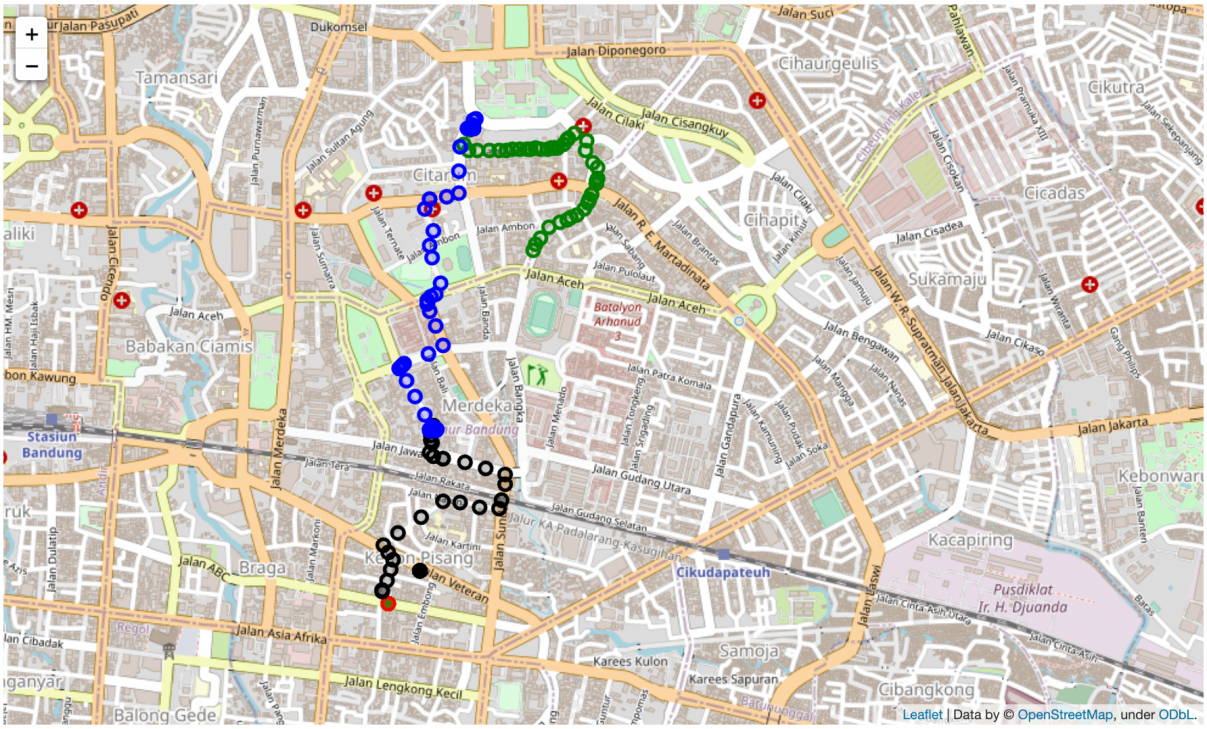

Figure 31. Movement pattern of Go Food real order

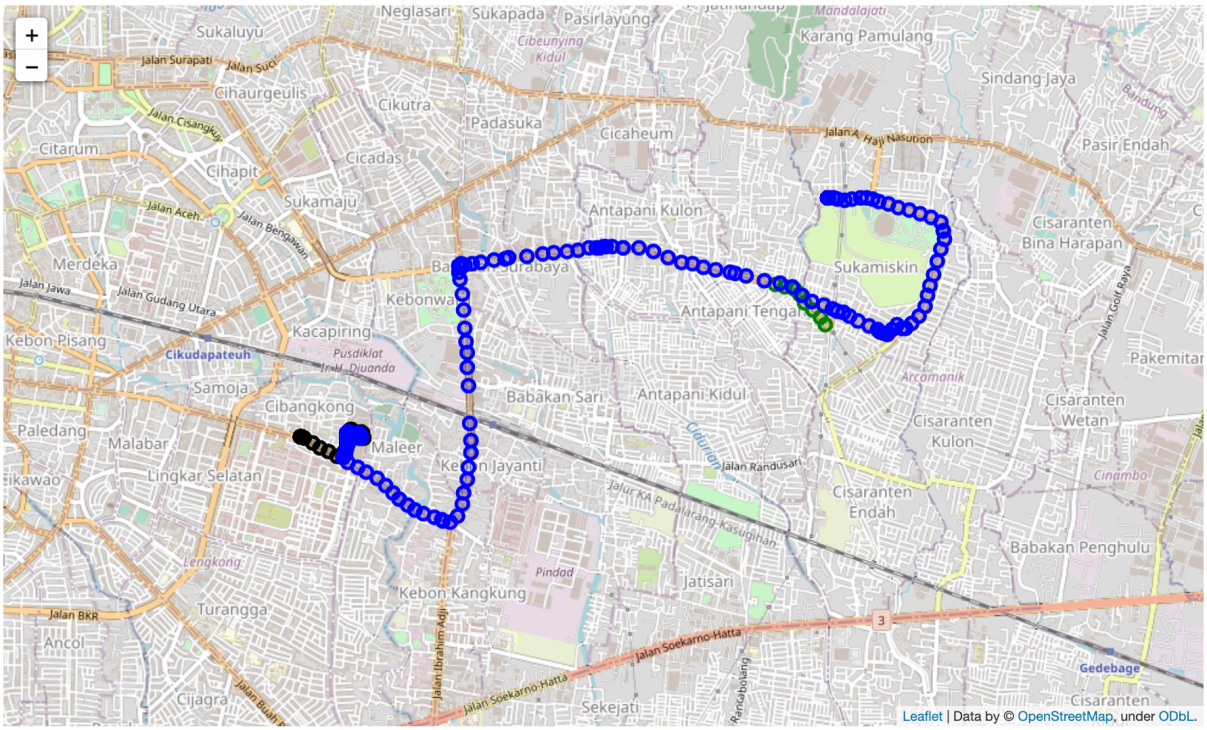

Figure 32. Movement pattern of Go Food fake order

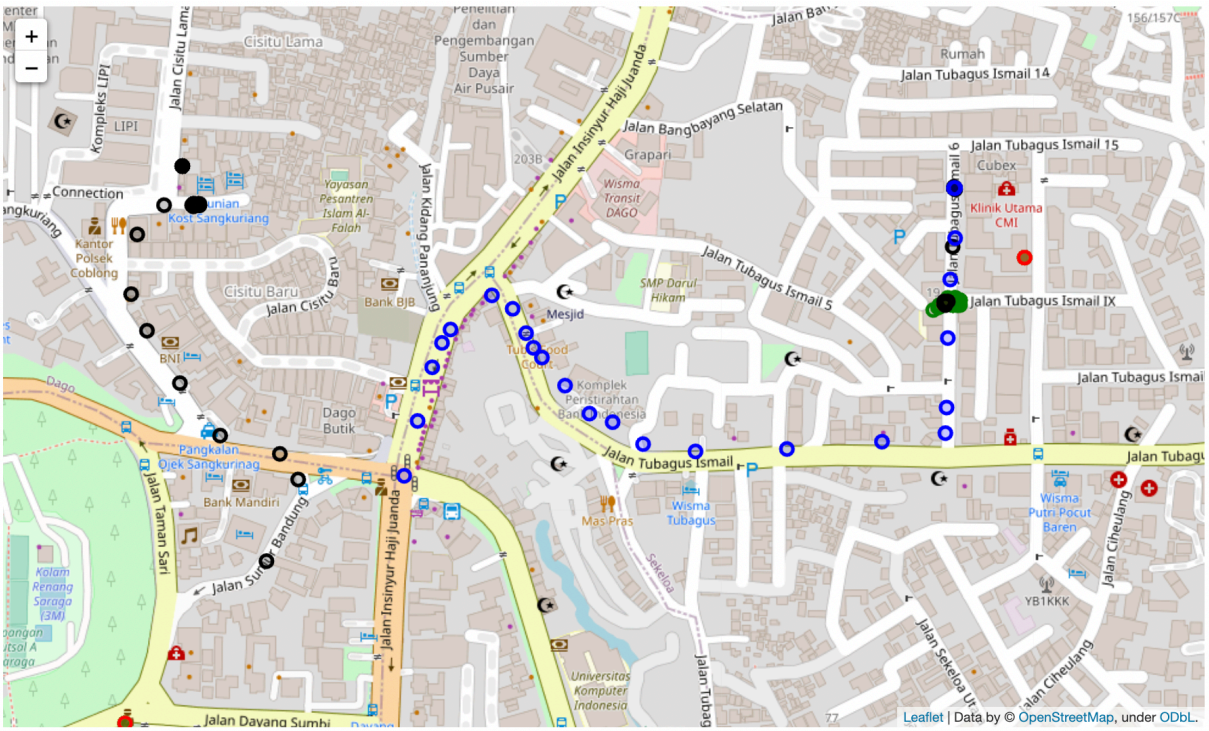

Figure 33. Movement pattern of Go Ride real order

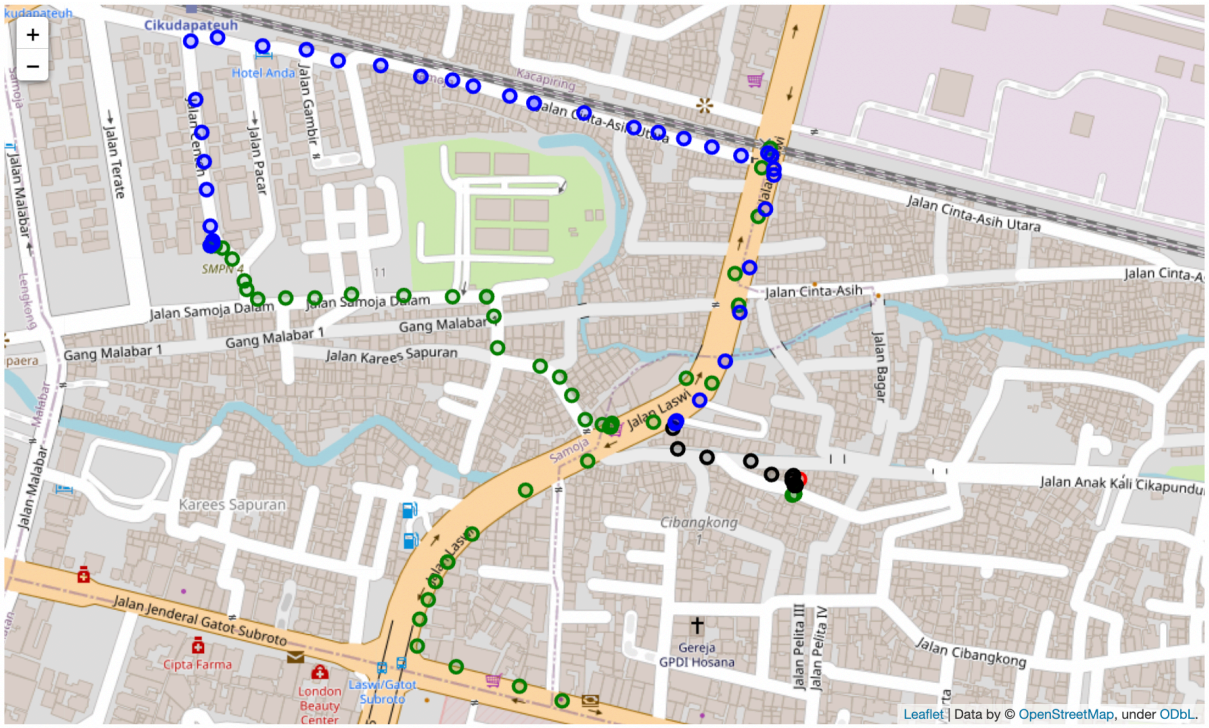

Figure 34. Movement pattern of Go Ride fake order

4.4.4 Checking imbalance in the dataset

The data should be checked whether it is imbalanced or not. If it is imbalanced, a particular sampling technique should be applied as it is crucial to detect the minority class correctly. Machine learning models should not be biased to identify only the majority class but should give equal weight or importance towards the minority class as well. The number of pings labelled as unreal was less than that of real ones. About 28% of the data was fraud. According to Google Machine Learning Course (Imbalanced Data, 2022), the proportion of minority class between 20% to 40% of the data set is mild as shown in table 10. Therefore, in the GoJek case, the data was not imbalanced. As a result, there was no need to do the oversample approach.

Table 10. Degree of imbalance

Degree of imbalance	Proportion of Minority Class
Mild	20-40% of the data set
Moderate	1-20% of the data set
Extreme	<1% of the data set

4.5 Feature Engineering and Variable Transformation

4.5.1 Converting binary variables to 0/1

To build the models, binary variables were converted to numerical format because the libraries does not support the categorical variable. Specifically, is_weekend and is_business_hour were transformed to 0 and 1 values.

4.5.1 Transform the dataset to have one row per ride

Up to this step, in the current dataset, one order had many rows because it had several driving statuses and each status had a number of pings, making the data very long. There was a need to transform the dataset so that one ride was on one row only to make it shorter and wider. With a wide dataset, there would be more variables for machine learning models to select and predict the unseen data.

Firstly, several pings of one status should be shorter to one row only. To do this, for each variable, only the maximum and minimum values among all the pings in each status should be kept and the others should be removed. Then, rows with similar data were removed and as a result, one status would only have one row. Secondly, a number of new columns were created to represent values of all statuses of one ride, such as total_distance_AVAILABLE, total_distance_OTW_DROPOFF, total_distance_OTW_PICKUP, total_distance_UNAVAILABLE. Thanks to that, for one ride, only one row was needed to exist in the dataset. The others could be removed.

The result of this transformation was that one ride only had one row in the dataset. Each ride had the maximum and minimum values of each status. There were 3,500 rows in the data representing 3,500 orders. Each order had 79 features as shown in the image 35.

```
# Get all columns
data.columns
```

```
Index(['order_id', 'service_type', 'label', 'min_day_of_week_AVAILABLE',
       'min_day_of_week_OTW_DROPOFF', 'min_day_of_week_OTW_PICKUP',
       'min_day_of_week_UNAVAILABLE', 'days_diff_AVAILABLE',
       'days_diff_OTW_DROPOFF', 'days_diff_OTW_PICKUP',
       'days_diff_UNAVAILABLE', 'max_distance_AVAILABLE',
       'max_distance_OTW_DROPOFF', 'max_distance_OTW_PICKUP',
       'max_distance_UNAVAILABLE', 'total_distance_AVAILABLE',
       'total_distance_OTW_DROPOFF', 'total_distance_OTW_PICKUP',
       'total_distance_UNAVAILABLE', 'max_time_diff_AVAILABLE',
       'max_time_diff_OTW_DROPOFF', 'max_time_diff_OTW_PICKUP',
       'max_time_diff_UNAVAILABLE', 'total_time_diff_AVAILABLE',
       'total_time_diff_OTW_DROPOFF', 'total_time_diff_OTW_PICKUP',
       'total_time_diff_UNAVAILABLE', 'avg_velocity_AVAILABLE',
       'avg_velocity_OTW_DROPOFF', 'avg_velocity_OTW_PICKUP',
       'avg_velocity_UNAVAILABLE', 'is_weekend_AVAILABLE',
       'is_weekend_OTW_DROPOFF', 'is_weekend_OTW_PICKUP',
       'is_weekend_UNAVAILABLE', 'is_business_hour_AVAILABLE',
       'is_business_hour_OTW_DROPOFF', 'is_business_hour_OTW_PICKUP',
       'is_business_hour_UNAVAILABLE', 'min_date_cos_AVAILABLE',
       'min_date_cos_OTW_DROPOFF', 'min_date_cos_OTW_PICKUP',
       'min_date_cos_UNAVAILABLE', 'min_date_sin_AVAILABLE',
       'min_date_sin_OTW_DROPOFF', 'min_date_sin_OTW_PICKUP',
       'min_date_sin_UNAVAILABLE', 'start_hour_cos_AVAILABLE',
       'start_hour_cos_OTW_DROPOFF', 'start_hour_cos_OTW_PICKUP',
       'start_hour_cos_UNAVAILABLE', 'start_hour_sin_AVAILABLE',
       'start_hour_sin_OTW_DROPOFF', 'start_hour_sin_OTW_PICKUP',
       'start_hour_sin_UNAVAILABLE', 'end_hour_cos_AVAILABLE',
       'end_hour_cos_OTW_DROPOFF', 'end_hour_cos_OTW_PICKUP',
       'end_hour_cos_UNAVAILABLE', 'end_hour_sin_AVAILABLE',
       'end_hour_sin_OTW_DROPOFF', 'end_hour_sin_OTW_PICKUP',
       'end_hour_sin_UNAVAILABLE', 'min_acc_AVAILABLE', 'min_acc_OTW_DROPOFF',
       'min_acc_OTW_PICKUP', 'min_acc_UNAVAILABLE', 'max_acc_AVAILABLE',
       'max_acc_OTW_DROPOFF', 'max_acc_OTW_PICKUP', 'max_acc_UNAVAILABLE',
       'mean_acc_AVAILABLE', 'mean_acc_OTW_DROPOFF', 'mean_acc_OTW_PICKUP',
       'mean_acc_UNAVAILABLE', 'median_acc_AVAILABLE',
       'median_acc_OTW_DROPOFF', 'median_acc_OTW_PICKUP',
       'median_acc_UNAVAILABLE'],
      dtype='object')
```

Figure 35. Features after the transformation

4.5.2 Transform null data

The approaches to deal with null data for categorical and numerical variables were slightly different. For categorical data, null data was replaced with is_na, so later it became a feature when creating dummy features in the next step. For numerical data, new columns with suffix '_is_na' were created to indicate if there are missing values for each variable. Then, the NaN values in numerical columns were replaced with zero.

4.5.3 Create dummy features

In the dataset, categorical variables have various forms. To allow machine learning models to read such predictors, first, they must be converted to numerical representations (Kuhn and Johnson, 2019), which are called dummy features. Each dummy takes either the value zero or one, indicating the absence or presence of each category in each categorical column in the original dataset. For example, 'service_type' column contains two categories, Go Food and Go Ride. After the one-hot encoding, there was a new column called "service_type_GO_RIDE" showing whether the ride is Go Ride or not. As a result, the service_type was no longer needed and was removed. Figure 36 presenting the result of one-hot encoding of the service_type feature.

	service_type
3	GO_RIDE
6	GO_FOOD
10	GO_FOOD
14	GO_FOOD
18	GO_FOOD

->

	service_type_GO_RIDE
3	1
6	0
10	0
14	0
18	0

Figure 36. Result of one-hot encoding of the service_type feature

4.5.4 Remove unnecessary columns

A number of unnecessary columns are removed as they are not needed anymore. For instance, latitude, longitude, date, hour were dropped as those are columns which were used to create new useful columns. Columns with high correlation with others were also be deleted. To illustrate, there were pairs having absolute correlation equal to one as figure 37 and one variable in each pair should be dropped as they are redundant.

```
Top 10 correlation pairs:

min_date_cos_AVAILABLE_is_na         min_acc_AVAILABLE_is_na             1.0
min_day_of_week_AVAILABLE_is_na      end_hour_cos_AVAILABLE_is_na        1.0
is_business_hour_UNAVAILABLE_is_na   mean_acc_UNAVAILABLE_is_na          1.0
min_day_of_week_AVAILABLE_is_na      total_time_diff_AVAILABLE_is_na     1.0
is_business_hour_UNAVAILABLE_is_na   max_acc_UNAVAILABLE_is_na           1.0
min_day_of_week_AVAILABLE_is_na      is_weekend_AVAILABLE_is_na          1.0
is_business_hour_UNAVAILABLE_is_na   min_acc_UNAVAILABLE_is_na           1.0
min_day_of_week_AVAILABLE_is_na      is_business_hour_AVAILABLE_is_na    1.0
                                     min_date_cos_AVAILABLE_is_na        1.0
is_business_hour_UNAVAILABLE_is_na   end_hour_sin_UNAVAILABLE_is_na      1.0
dtype: float64
```

Figure 37. Top 10 correlation pairs

4.6 Feature Selection using RFE

According to (Recursive Feature Elimination, 2019), RFE (Recursive feature elimination) is a feature selection approach fitting dependent and independent variables in the dataset to remove weak features until it reaches the specified number of features. RFE ranks features, and recursively eliminated a number of features per loop to prevent collinearity and dependencies existing in the dataset. Figure 38 shows the top 30 features selected by RFE.

```
# All of the columns selected by RFE
rfe_col = X.columns[rfe.support_]
rfe_col
```

```
Index(['min_day_of_week_AVAILABLE', 'min_day_of_week_OTW_PICKUP',
       'min_day_of_week_UNAVAILABLE', 'max_distance_AVAILABLE',
       'max_distance_UNAVAILABLE', 'total_distance_AVAILABLE',
       'total_distance_OTW_PICKUP', 'avg_velocity_OTW_PICKUP',
       'avg_velocity_UNAVAILABLE', 'is_weekend_AVAILABLE',
       'is_weekend_OTW_DROPOFF', 'is_weekend_OTW_PICKUP',
       'is_business_hour_AVAILABLE', 'is_business_hour_OTW_DROPOFF',
       'is_business_hour_OTW_PICKUP', 'is_business_hour_UNAVAILABLE',
       'start_hour_cos_UNAVAILABLE', 'start_hour_sin_AVAILABLE',
       'start_hour_sin_OTW_DROPOFF', 'start_hour_sin_OTW_PICKUP',
       'start_hour_sin_UNAVAILABLE', 'end_hour_cos_UNAVAILABLE',
       'end_hour_sin_AVAILABLE', 'end_hour_sin_OTW_DROPOFF',
       'min_acc_OTW_DROPOFF', 'min_day_of_week_OTW_DROPOFF_is_na',
       'min_day_of_week_UNAVAILABLE_is_na', 'max_distance_OTW_DROPOFF_is_na',
       'max_distance_UNAVAILABLE_is_na', 'service_type_GO_RIDE'],
      dtype='object')
```

Figure 38. top 30 features selected by RFE

4.7 Test-train split

Before building models, the dataset was split into two sets which are a train set and a test set. According to (Training and Test Sets: Splitting Data, 2022), a train set is a subset to train a model and a test set is a subset to test the trained model. If the model fit the training set, but

does not fit the test set well, the model has the overfitting issue. The model should work well for both training and testing datasets. Using the 'sklearn.model_selection.train_test_split', 70% of the dataset was split into the train set and the rest was the test set.

4.8 Model implementation

After the preprocessing process to obtain the necessary attributes, a variety of machine learning models, like Logistic Regression, Decision Tree and Random Forest were formed, to classify fake gps pings and normal pings.

Decision Tree models were built with tunable hyperparameters, such as maximum depth, minimum samples before split and minimum samples in leaf node (Decision tree classification, 2021). During the implementation of a decision tree, maximum tree depth is a limit to stop further nodes splitting when the depth has been reached with maximum depth. The depth was tuned and specified from three (in the figure 39) to seven to find the most effective threshold. min_samples_split specifies the minimum number of samples required to split an internal node, while min_samples_leaf specifies the minimum number of samples required to be at a leaf node. They were tried in this study with the threshold as 20.

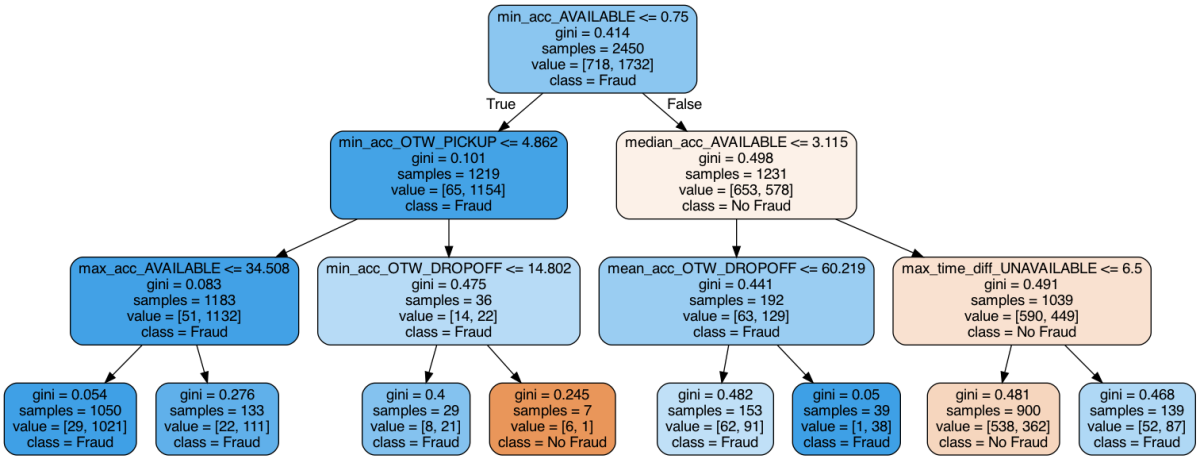

Figure 39. Decision Tree with max depth 3

To get a more accurate prediction, random forest models were implemented without hyperparameter tuning first, then with hyperparameter tuning. As a collection of decision trees, random forest randomly selects observations to build many decision trees and then averages the

results. Random forest model first was built with default parameters, and by measuring relative importance of each variable, it found important features as figure 40.

	Varname	Imp
64	mean_acc_AVAILABLE	0.069357
58	min_acc_OTW_PICKUP	0.062857
56	min_acc_AVAILABLE	0.057969
68	median_acc_AVAILABLE	0.055761
70	median_acc_OTW_PICKUP	0.037013
...	...	...
74	min_day_of_week_OTW_PICKUP_is_na	0.000219
83	days_diff_AVAILABLE_1 days	0.000046
85	days_diff_OTW_PICKUP_1 days	0.000041
84	days_diff_OTW_DROPOFF_1 days	0.000031
86	days_diff_UNAVAILABLE_1 days	0.000018

Figure 40. Important features found by Random Forest

Then, it was built with tuned parameters, specifically random_state=42, n_estimators=100, and max_depth=5. Finally, the Random Forest was tuned with Random Search as the figure 41. Figure 42 shows the best parameters of the random forest.

```
RandomizedSearchCV(cv=5, estimator=RandomForestClassifier(), n_iter=30,
                   n_jobs=4,
                   param_distributions={'bootstrap': [True, False],
                                        'criterion': ['gini', 'entropy'],
                                        'max_depth': [1, 5, 10, 20, 50, 75, 100,
                                                      150, 200],
                                        'min_samples_leaf': [1, 2, 3, 4],
                                        'min_samples_split': [1, 2, 5, 10, 15,
                                                              20, 30],
                                        'n_estimators': array([ 100,  307,  514,  721,  928, 1135, 1342, 1550,
1964, 2171,
       2378, 2585, 2792, 3000])},
                   random_state=42, verbose=2)
```

Figure 41. Random Forest was tuned with Random Search

```
1 rf_random.best_params_
```

```
{'n_estimators': 1757,
 'min_samples_split': 5,
 'min_samples_leaf': 2,
 'max_depth': 150,
 'criterion': 'entropy',
 'bootstrap': False}
```

Figure 42. Best parameters of the random forest

For Logistic Regression model, to avoid overfitting, Standard Scaler was used to scale numerical variables and the model was assessed with StatsModels. Variables with high VIF (VIF greater than 5) and high p-value (p-value greater than 0.05) were removed. ROC curve, a graph showing the model's performance as figure 43, had the area below the ROC was 0.74, which was good.

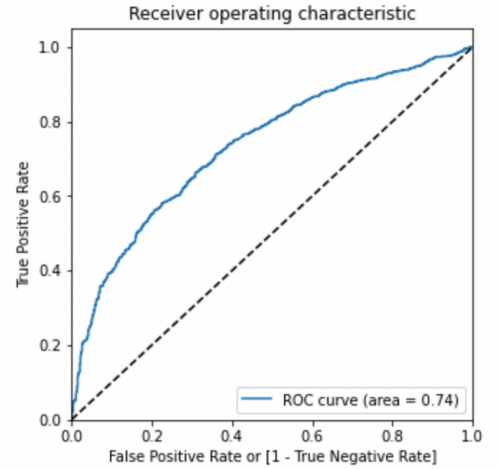

Figure 43. ROC curve - model's performance

The model predicted the probability of a ping to be a real one. With the cutoff point of 0.5, the accuracy was only 0.716. Therefore, there was a need to find an optimal cutoff point. Accuracy, sensitivity and specificity were plotted for various probabilities. They met at 0.65, thus 0.65 was the optimum point to take as a cutoff probability. This means pings with probability from 0.65 and above will be a normal ping. Otherwise, it is a fraud location ping.

4.9 Summary

In summary, the chapter presented the approach from dataset visualisation and preparation to feature engineering and models building. While preparing and doing models implementation, several interesting insights were found. Fraud data tended to have abnormal signs compared to the normal data, such as high velocity, high accuracy, and random movements. Furthermore, variables mentioning day of week, distance, velocity and business hour were defined as important features by the RFE.

CHAPTER 5: RESULTS AND DISCUSSIONS

5.1 Introduction

After the models are built, they will be evaluated to find the best one. Chapter five contains the results of the models' evaluation and the prediction of the unseen data. At the end of this chapter, the best model fitting the GoJek dataset will be revealed.

5.2 Model Evaluation

To measure the performance of algorithms in this study, evaluation methods were used, including cross validation, accuracy, precision and recall.

Logistic Regression Model was tried with several different cut-off points and selected the best cut-off point at 0.65. However, the accuracy, specificity and recall values were not good as shown in table 11. Hence, it was not fit for this study.

Table 11. Logistic Regression model evaluation

	Accuracy		**Specificity**		**Recall**	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
Logistic Regression	0.681	0.602	0.647	0.897	0.695	0.472

The table 12 shows the metrics' result of Decision Tree models. Although the model with tuned minimum samples split and minimum samples leaf had high accuracy, specificity and recall with train set, those values of test set were not high enough, meaning that those models were overfit. The Decision Tree model with tuned maximum depth had those values similar on both test and train sets. Furthermore, its' cross validation average score was higher than the other two. Therefore, among all decision tree models tried, the one controlling the depth of the tree best fit the purpose of this study.

Table 12. Metrics' result of Decision Tree models

Decision Tree Models	Cross validation average score	Accuracy		Specificity		Recall	
		Train	Test	Train	Test	Train	Test
max depth = 3	0.765	0.781	0.781	0.758	0.81	0.79	0.768
min_samples_split=20	0.746	0.922	0.77	0.886	0.673	0.938	0.813
min_samples_leaf=20	0.751	0.853	0.76	0.734	0.667	0.903	0.801

Diving more into the decision tree models with tuned maximum depth, the results returned as below. The models with maximum 5, 6, and 7 had the best values of cross validation average score, accuracy, specificity and recall as presented in table 13.

Table 13. Decision tree models with tuned parameters

Decision Tree Models	Cross validation average score	Accuracy		Specificity		Recall	
		Train	Test	Train	Test	Train	Test
max depth = 3	0.765	0.781	0.781	0.758	0.81	0.79	0.768
max depth = 4	0.759	0.803	0.771	0.816	0.81	0.798	0.754
max depth = 5	0.767	0.818	0.761	0.875	0.832	0.795	0.73
max depth = 6	0.76	0.829	0.772	0.898	0.841	0.801	0.742
max depth = 7	0.764	0.86	0.788	0.879	0.81	0.852	0.778

Random Forest models witnessed a clear overfit issue as shown in table 14. With the default Random Forest Model, the accuracy, specificity and recall all had value as 1 on the train data, while with test data, those values were less than 0.9. Similar issue happened on the model tuned with Randomised Search. As a result, among all random forest models tried, the one tuned manually best fit the purpose of this study.

Table 14. Random Forest models evaluation

Random Forest Models	Cross validation average score	Accuracy		Specificity		Recall	
		Train	Test	Train	Test	Train	Test
Default	0.807	1	0.804	1	0.701	1	0.849
Tuned Manually	0.786	0.825	0.787	0.776	0.76	0.846	0.798
Tuned with Randomised Search	0.81	1	0.808	1	0.72	1	0.85

5.3 Predict unseen data

All models were tested on test set (unseen data) by submitting the predicted results to Kaggle's platform. On Kaggle, the test set was split into two parts. 50% of the predictions from the test set were used to calculated public score. The other 50% were used for private score. Table 15 highlights the result.

Table 15. Kaggle scores

Models	Private Score	Public Score
Random Forest tuned manually	0.69449	0.75777
Random Forest tuned with RandomSearch	0.67763	0.78113
Random Forest default	0.70078	0.77168
Decision Tree with min samples before split = 20	0.6598	0.63681
Decision Tree with min samples in leaf node = 20	0.60785	0.59943
Decision Tree with max depth = 7	0.71383	0.65268
Decision Tree with max depth = 6	0.75967	0.74331
Decision Tree with max depth = 5	0.76348	0.76668
Decision Tree with max depth = 4	0.76243	0.74103
Decision Tree with max depth = 3	0.74852	0.7319
Logistic Regression	0.70477	0.68767

Overfitting issues happened in some models where the private score and public score were quite different. To illustrate, Random Forest models had public scores greater than 0.75, while their private scores were less than 0.7. This means those models worked better on one set of data compared to the other.

Decision Tree models controlling the maximum depth showed good results where all of them had private score greater than 0.7. The one with max depth 7 was overfitted as the private score was good, but the public score was not good. The max depth 5 was the best model since the private score was quite similar to the public score and they were all around 0.76, which is in the top four of submissions on Kaggle in terms of both private and public scores.

5.4 Summary

To sum up, the models were evaluated by cross validation, accuracy, precision and recall. Decision Tree models, especially the one with max depth as five appeared to be the best fit for the dataset. The same result was shown when testing on unseen data as Kaggle returned good score for that Decision Tree model.

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

Real location can be changed by interfering the GPS signals and by fake location applications. Most of the previous papers proposed solutions to deal with the first method. There was one paper predicting the fake location on mobile device using Machine Learning Logistic Regression model (Mar'i & Pangestu, 2021). No other models were investigated in that paper while other machine learning models can be used for classification, such as Decision Tree, Naive Bayes and K-Nearest Neighbors. Therefore, this paper examined several machine learning models, Logistic Regression, Decision Tree and Random Forest with parameters tunning to find the best one for GOJEK to detect spoofing.

6.2 Conclusion

In conclusion, with the results in chapter 5, this study successfully predicted and classified fake GPS pings in GOJEK's dataset. By reviewing previous papers, it has been found that there were patterns in location spoofing cases and machine learning could detect that, especially in the GOJEK case. In this study, three algorithms were used, namely Logistic Regression, Decision Tree and Random Forest. Their performances were measured by a 5-fold Cross Validation, Accuracy, Specificity and Recall. Decision Tree model had the highest scores, so it was implemented further by tuning hyper parameters. The result was that Decision Tree model with maximum depth as five was the best one. It had the cross validation average score 76.7%, maximum accuracy value of 81.8%, specificity 87.5% and recall 79.5%. The model also received the high result on Kaggle submission with the maximum accuracy of 76.3% on test (unseen) data.

6.3 Contribution to knowledge

Three findings that can be learned from this study are approaches to work with spatial data, characteristics of unreal GPS location and models that work well to detect location spoofing.

The first outcome is one of the best practices that can be used when dealing with geolocation dataset. With spatial data available such as coordinates, altitude, accuracy and time variables, a lot of information can be revealed. From coordinates, locations can be extracted using the Nominatim function. Altitude in the data then can be compared with the real altitude at the location to see if it matches the real one. Time difference and distance between two consecutive pings should be used to calculated velocity. Velocity and accuracy can help to classify between real and fake locations.

The second outcome of this research is the abnormal signs of unreal locations observed from the GoJek dataset. "Go Ride" had more fraud pings than "Go Food". Accuracy and velocity of fake location pings were higher than those of real ones. Real orders had the pattern of movement: Available -> Pick up -> Drop off -> Available, but fake orders did not follow that pattern.

The final outcome is that Decision Tree and Random Forest returned better results compared to Logistic Regression. However, Random Forest can give overfitting results. Decision Tree with tuning maximum tree depth demonstrated a good fit for location spoofing detection.

6.4 Future Recommendations

Further analysis can be done by implementing different algorithms or doing hyper parameter-tuning with different approaches which due to the time limitation, this research cannot cover. Naive Bayes and K-Nearest Neighbors models can be tried. Bayesian Optimization is also a good method to tune hyper parameters (Rendyk, 2021). In other words, the dataset of GoJek can be utilised to evaluate classification performance of other machine learning algorithms.

REFERENCES

3UTools, (2022) *3UTools*. [online] 2022. Available at: http://www.3u.com/ [Accessed 18 Sep. 2022].

ABIOLA, (2017) *Drivers on these platforms are using a fake GPS app called 'Lockito' to scam riders*. [online] Available at: https://www.pulse.ng/news/tech/uber-taxify-drivers-on-these-platforms-are-using-a-fake-gps-app-called-lockito-to/z2jsc1f [Accessed 1 Sep. 2022].

Ahmad, M., Farid, M.A. and Ahmed, S., (2019) Countermeasures against Spoofing. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, [online] February, pp.1–8. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8673518.

Anaconda, (2022) *End User License Agreement - Anaconda® Distribution*. [online] Anaconda. Available at: https://www.anaconda.com/eula-anaconda-distribution [Accessed 23 Jul. 2022].

Android Developers, (2022) *Location*. [online] Available at: https://developer.android.com/reference/android/location/Location#isFromMockProvider%28%29 [Accessed 15 Sep. 2022].

Anon (2019) *Recursive Feature Elimination*. [online] Scikit-yb. Available at: https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html [Accessed 5 Dec. 2022].

Anon (2021) *Decision tree classification*. [online] IBM. Available at: https://www.ibm.com/docs/en/db2/9.7?topic=classification-decision-tree [Accessed 30 Nov. 2022].

Anon (2022) *Bandung*. [online] Google Arts And Culture. Available at: https://artsandculture.google.com/entity/bandung/m01zll8?hl=en [Accessed 1 Dec. 2022].

Anon (2022) *Cross-validation: evaluating estimator performance*. [online] Scikit-learn. Available at: https://scikit-learn.org/stable/modules/cross_validation.html [Accessed 1 Dec. 2022].

Anon (2022) *Imbalanced Data*. [online] developers.google.com. Available at: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data [Accessed 30 Nov. 2022].

Anon (2022) *Training and Test Sets: Splitting Data*. [online] Google Developers. Available at: https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data [Accessed 4 Dec. 2022].

App Ninjas, (2017) *What Are GPS Spoofing Apps Actually Doing?* [online] Medium.com. Available at: https://medium.com/@theappninjas/what-are-gps-spoofing-apps-actually-doing-5c9f373540c4 [Accessed 10 Sep. 2022].

Arias-Pérez, J., Coronado-Medina, A. and Perdomo-Charry, G., (2022) *Big data analytics capability as a mediator in the impact of open innovation on firm performance*. *Journal of Strategy and Management*, Available at: https://www.researchgate.net/publication/352411670_Big_data_analytics_capability_as_a_mediator_in_the_impact_of_open_innovation_on_firm_performance.

Bahel, V., Pillai, S. and Malhotra, M., (2020) A Comparative Study on Various Binary Classification Algorithms and their Improved Variant for Optimal Performance. *2020 IEEE Region 10 Symposium, TENSYMP 2020*, May, pp.495–498.

Baijayanta, R., (2020) All about Feature Scaling. *Towards Data Science*. [online] Available at: https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35.

Bhatti, J. and Humphreys, T.E., (2017) Hostile Control of Ships via False GPS Signals: Demonstration and Detection. *Navigation, Journal of the Institute of Navigation*, [online] 641, pp.51–66. Available at: https://onlinelibrary.wiley.com/doi/epdf/10.1002/navi.183?saml_referrer.

Bluestacks, (2022) *Bluestacks*. [online] Available at: https://www.bluestacks.com/ [Accessed 20 Sep. 2022].

Chao, J., (2020) *Who are the new investors of GoJek?* [online] Available at: https://thelowdown.momentum.asia/who-are-the-new-investors-of-gojek/ [Accessed 25 Jul. 2022].

Chapman, A., (2017) GPS Spoofing. *2017 Tech Notes*. [online] Available at: https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/Red_Chapman.pdf.

Chong, R., (2021) *Android Emulator Detection*. [online] Medium.com. Available at: https://ray-chong.medium.com/android-emulator-detection-4d0f994aab5e [Accessed 23 Sep. 2022].

Damaini, A.A., Nugroho, G.S. and Suyoto, (2018) Fraud crime mitigation of mobile application users for online transportation. *International Journal of Interactive Mobile Technologies*, 123, pp.153–167.

Dossman, C., (2017) Top 6 Errors Novice Machine Learning Engineers Make. *AI³ | Theory, Practice, Business*.

Ebinuma, T., (2022) *GPS-SDR-SIM*. Available at: https://github.com/osqzss/gps-sdr-sim.

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B. and Tabona, O., (2021) *A survey on missing data in machine learning*. [online] *Journal of Big Data*, Springer International Publishing. Available at: https://doi.org/10.1186/s40537-021-00516-9.

Fan, Y., Zhang, Z., Trinkle, M., Dimitrovski, A.D., Song, J. bin and Li, H., (2015) A Cross-Layer Defense Mechanism Against GPS Spoofing Attacks on PMUs in Smart Grids. *IEEE Transactions on Smart Grid*, 66, pp.2659–2668.

Fawagreh, K., Gaber, M.M. and Elyan, E., (2014) Random forests: From early developments to recent advancements. *Systems Science and Control Engineering*, [online] 21, pp.602–609. Available at: https://doi.org/10.1080/21642583.2014.956265.

Florene, U., (2020) *Gojek contributed USD 7.1 billions to Indonesia's economy in 2019, study says*. [online] Kr Asia. Available at: https://kr-asia.com/gojek-contributed-usd-7-1-billions-to-indonesias-economy-in-2019-study-says [Accessed 29 Jul. 2022].

George, (2022) *How to Fake GPS Location on iPhone*. [online] iKream. Available at: https://www.ikream.com/fake-gps-location-on-iphone-33149 [Accessed 22 Sep. 2022].

GOJEK, (2019) *Fake GPS Detection*. [online] GOJEK Bootcamp 2019 Kaggle Challenge. Available at: https://www.kaggle.com/c/godsbootcamp04/data [Accessed 20 Jul. 2022].

Google Cloud, (2022) *GO-JEK: Using Machine Learning for forecasting and dynamic pricing*. [online] Google Cloud. Available at: https://cloud.google.com/customers/go-jek [Accessed 5 Sep. 2022].

Haddad, R., Kelly, T., Leinonen, T. and Saarinen, V., (2014) Using locational data from mobile phones to enhance the science of delivery. June, pp.1–55.

Haider, Z. and Khalid, S., (2017) Survey on effective GPS spoofing countermeasures. *2016 6th International Conference on Innovative Computing Technology, INTECH 2016*, [online] pp.573–577. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7845038.

Hidajat, T., Kusuma, A.H. and Sulchan, A., (2021) Gamification in Ride-Hailing: What Drives a Driver to Drive. In: *Proceedings of the 3rd International Conference on Banking, Accounting, Management and Economics (ICOBAME 2020)*. Atlantis Press.

Hijmans, R.J., (2021) Introduction to the geosphere package (version 1 . 9-92). *Cran*, [online] pp.1–19. Available at: https://cran.r-project.org/web/packages/geosphere/vignettes/geosphere.pdf.

Istiardi, R.F. and Setiawati, dan E., (2022) The Effect of Internal Control and Fraud Triangle on Fraud Behavior in Online Transportation. *Proceedings of the International Conference on Economics and Business Studies (ICOEBS 2022)*, 655Icoebs, pp.106–111.

iSunshare, (2022) *2 Ways to Convert User App to System App on Android*. [online] Available at: https://www.isunshare.com/android/2-ways-to-convert-user-app-to-system-app.html [Accessed 10 Sep. 2022].

iTools, (2022) *iTools*. [online] Available at: https://www.itools4.com/ [Accessed 19 Sep. 2022].

Jakarta, (2019) *Go-Jek prosecutes partners creating fake orders*. [online] The Jakarta Post. Available at: https://www.thejakartapost.com/news/2019/07/23/go-jek-prosecutes-partners-creating-fake-orders.html [Accessed 1 Aug. 2022].

Jansen, K., Schafer, M., Moser, D., Lenders, V., Popper, C. and Schmitt, J., (2018) Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks. *Proceedings - IEEE Symposium on Security and Privacy*, [online] 2018-May, pp.1018–1031. Available at: https://ieeexplore.ieee.org/document/8418651.

Jiang, P., Wu, H. and Xin, C., (2021) DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network. *Digital Communications and Networks*, [online] September 2021, pp.1–13. Available at: https://doi.org/10.1016/j.dcan.2021.09.006.

Jovanovic, A., Botteron, C. and Fariné, P., (n.d.) Multi-test Detection and Protection Algorithm Against Spoofing Attacks on GNSS Receivers. pp.1258–1271.

Jupyter, (2015) *Installing the classic Jupyter Notebook interface*. [online] Jupyter Team. Available at: https://docs.jupyter.org/en/latest/install/notebook-classic.html [Accessed 29 Jul. 2022].

Kerns, A.J., Shepard, D.P., Bhatti, J.A. and Humphreys, T.E., (2014) Unmanned Aircraft Capture and Control via GPS Spoofing. *Journal of Field Robotics*, [online] 314, pp.617–636. Available at: https://rnl.ae.utexas.edu/images/stories/files/papers/unmannedCapture.pdf.

Khong, N., (2019) *Drivers use GPS spoofing, fake apps to defraud Grab, says ride-sharing firm*. [online] Todau Online. Available at: https://www.todayonline.com/singapore/grab-outlines-ways-its-drivers-try-defraud-ride-sharing-firm [Accessed 25 Jul. 2022].

Koh, J.Y., Nevat, I., Leong, D. and Wong, W.C., (2016) Geo-Spatial Location Spoofing Detection for Internet of Things. *IEEE Internet of Things Journal*, 36, pp.971–978.

Kotawadekar, R., (2021) *Satellite data: Big data extraction and analysis*. [online] *Artificial Intelligence in Data Mining: Theories and Applications*. INC. Available at: http://dx.doi.org/10.1016/B978-0-12-820601-0.00008-2.

Kuhn, M. and Johnson, K., (2019) *Feature engineering and selection: A practical approach for predictive models*. [online] *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Available at: https://bookdown.org/max/FES/.

Kuhn, M.G., (2004) An Asymmetric Security Mechanism for Navigation Signals. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3200, pp.239–252.

Lee, J.H. and Buehrer, R.M., (2012) Characterization and Detection of Location Spoofing Attacks. *Journal of Communications and Networks*, 144, pp.396–409.

Lim, Y.-X., (2019) *An Introduction to Gojek's Machine Learning Platform*. [online] Gojek Tech. Available at: https://www.gojek.io/blog/an-introduction-to-gojeks-machine-learning-platform.

Liu, G., Zhang, R., Wang, C. and Liu, L., (2019) Synchronization-free GPS spoofing detection with crowdsourced air traffic control data. *Proceedings - IEEE International Conference on Mobile Data Management*, [online] 2019-JuneMdm, pp.260–268. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8788766.

Lotan, G., Ananny, M., Gaffney, D. and Boyd, D., (2011) The Revolutions Were Tweeted : Information Flows During the 2011 Tunisian and Egyptian Revolutions Web Ecology Project Web Ecology Project. 5, pp.1375–1406.

Maddison, R. and Ni Mhurchu, C., (2009) *Global positioning system: A new opportunity in physical activity measurement*. *International Journal of Behavioral Nutrition and Physical Activity*, .

Magiera, J. and Katulski, R., (2015) Detection and mitigation of GPS spoofing based on antenna array processing. *Journal of Applied Research and Technology*, [online] 131, pp.45–57. Available at: http://dx.doi.org/10.1016/S1665-6423(15)30004-3.

Magisk Manager, (2022) *Download Magisk Manager Latest Version 25.2 For Android 2022*. [online] Available at: https://magiskmanager.com/ [Accessed 14 Sep. 2022].

Maheshwari, S., (2016) How Fake News Goes Viral: A Case Study. *The New York Times*. [online] Available at: https://www.nytimes.com/2016/11/20/business/media/how-fake-news-spreads.html?_r=0.

Mar'i, F. and Pangestu, G., (2021) Classification of Fake GPS in GOJEK Application using Logistic Regression. *ACM International Conference Proceeding Series*, pp.94–99.

Memuplay, (2022) *Memuplay*. [online] Available at: https://www.memuplay.com/ [Accessed 25 Sep. 2022].

van der Merwe, J.R., Zubizarreta, X., Lukčin, I., Rügamer, A. and Felber, W., (2018) Classification of Spoofing Attack Types. *2018 European Navigation Conference, ENC 2018*, July, pp.91–99.

Moslehpour, M., Ismail, T., Purba, B. and Wong, W.K., (2022) What makes go-jek go in indonesia? The influences of social media marketing activities on purchase intention. *Journal of Theoretical and Applied Electronic Commerce Research*, 171, pp.89–103.

Nandi, (2019) The Influence of Online Transportation Application to the Mobility and Economic of the Society (Case Study on Using Grab and Go-Jek in Bandung, Indonesia). *IOP Conference Series: Earth and Environmental Science*, 2861.

Ossmann, M., (2022) *Hackrf*. Available at: https://github.com/greatscottgadgets/hackrf [Accessed 20 Sep. 2022].

Pandey, N., Tandon, S. and Jain, P., (2021) EasyChair Preprint Geolocation Based Recommender System GEOLOCATION BASED RECOMMENDER SYSTEM.

Pandita, A., (2022) *Is Your Android Rooted? Detect With These 6 Proven Ways*. [online] Available at: https://www.appknox.com/blog/root-detection-techniques [Accessed 12 Sep. 2022].

Peng, C.Y.J., Lee, K.L. and Ingersoll, G.M., (2002) An introduction to logistic regression analysis and reporting. *Journal of Educational Research*, 961, pp.3–14.

Powers, D.M.W., (2020) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. [online] May. Available at: http://arxiv.org/abs/2010.16061.

Psiaki, M.L. and Humphreys, T.E., (2015) GNSS Spoofing and Detection The Problem : Surreptitious Receiver. [online] June, pp.1–11. Available at: https://radionavlab.ae.utexas.edu/images/stories/files/papers/gnss_spoofing_detection.pdf.

Qadri, R., (2021) *Delivery Drivers Are Using Grey Market Apps to Make Their Jobs Suck Less*. [online] Tech By Vice. Available at: https://www.vice.com/en/article/7kvpng/delivery-drivers-are-using-grey-market-apps-to-make-their-jobs-suck-less [Accessed 1 Sep. 2022].

Rendyk, (2021) *Bayesian Optimization: bayes_opt or hyperopt*. [online] Data Science Blogathon. Available at: https://www.analyticsvidhya.com/blog/2021/05/bayesian-optimization-bayes_opt-or-hyperopt/ [Accessed 12 Dec. 2022].

Robinson, P. and Scassa, T. eds., (2022) *The Future of Open Data*. *News.Ge*. University of Ottawa Press.

Rustamov, R., (2018) *Multifunctional Operation and Application of GPS*. [online] Available at: https://www.intechopen.com/books/6540.

Salgado, C.M., Azevedo, C., Proença, H. and Vieira, S.M., (2016) *Setting the Stage: Rationale Behind and Challenges to Health Data Analysis*. *Secondary Analysis of Electronic Health Records*.

Shinde, P.P. and Shah, S., (2018) A Review of Machine Learning and Deep Learning Applications. *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*.

Siamak, S., Dehghani, M. and Mohammadi, M., (2019) Counteracting GPS Spoofing Attack on PMUs by Dynamic State Estimation. *2019 Smart Gird Conference, SGC 2019*. [online] Available at: https://ieeexplore.ieee.org/document/9056583.

Simon, Annina, Deo, Mahima, Selvam, Venkatesan, Babu and Ramesh, (2016) An Overview of Machine Learning and its Applications. International Journal of Electrical Sciences & Engineering. Volume. 22-24. [online] January. Available at: https://www.researchgate.net/publication/289980169_An_Overview_of_Machine_Learning_and_its_Applications.

Song, Y. and Lu, Y., (2015) Decision tree methods : applications for classification and prediction. 272, pp.130–135.

Stanley, M. and Gunn, G., (2018) Using Technology to Solve Today's Water Challenges. *Winnipeg: International Institute for ...*, [online] August. Available at: https://www.greengrowthknowledge.org/sites/default/files/downloads/best-practices/IISD_using-technology-solve-water-challenges.pdf.

Subramanian, S.G. and Crowley, M., (2018) Using spatial reinforcement learning to build forest wildfire dynamics models from satellite images. *Frontiers in ICT*, 5APR, p.3389.

Sudheer, A., Baranwal, I. and Gadhoke, B., (2022) KNN-Based Hybrid Models for Geolocation based Problems. [online] pp.1–6. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9848200.

Tenkanen, H. and Toivonen, T., (2020) Longitudinal spatial dataset on travel times and distances by different travel modes in Helsinki Region. *Scientific Data*, [online] 71, pp.1–15. Available at: http://dx.doi.org/10.1038/s41597-020-0413-y.

The University of Texas at Austin, (2013) *Spoofing a Superyacht at Sea*. [online] The University of Texas at Austin. Available at: https://news.utexas.edu/2013/07/30/spoofing-a-superyacht-at-sea/ [Accessed 30 Jul. 2022].

Warner, J.S., Johnston, R.G. and Cpp Los Alamos, (2003) GPS Spoofing Countermeasures. [online] pp.1296–1296. Available at: http://mirror.thelifeofkenneth.com/lib/electronics_archive/GPS-Spoofing-Countermeasures.pdf.

Wong, S.K. and Yiu, S.M., (2020) Location spoofing attack detection with pre-installed sensors in mobile devices. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 114, pp.16–30.

Wood, J., (2008) *Filter and Refine Strategy*. [online] Springer, Boston, MA. Available at: https://link.springer.com/referenceworkentry/10.1007/978-0-387-35973-1_415.

Young, J.R.M.D., (2022) *Xcode Spoof Location: The Complete Guide in 2022*. [online] Foneazy. Available at: https://www.foneazy.com/tips/xcode-spoof-location/ [Accessed 15 Sep. 2022].

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V. and Darrell, T., (2020) BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, [online] pp.2633–2642. Available at: https://openaccess.thecvf.com/content_CVPR_2020/papers/Yu_BDD100K_A_Diverse_Driving_Dataset_for_Heterogeneous_Multitask_Learning_CVPR_2020_paper.pdf.

Zhao, B. and Sui, D.Z., (2017) True lies in geospatial big data: detecting location spoofing in social media. *Annals of GIS*, [online] 231, pp.1–14. Available at: http://dx.doi.org/10.1080/19475683.2017.1280536.

Zhao, B. and Zhang, S., (2019) Rethinking Spatial Data Quality: Pokémon Go as a Case Study of Location Spoofing. *Professional Geographer*, [online] 711, pp.96–108. Available at: https://doi.org/10.1080/00330124.2018.1479973.

Zola, A. and Fontecchio, M., (2021) *Spatial data*. [online] TechTarget. Available at: https://www.techtarget.com/searchdatamanagement/definition/spatial-data#:~:text=Spatial data is any type,in a geographic coordinate system [Accessed 20 Sep. 2022].

APPENDIX A: RESEARCH PLAN

This research is expected to be done within four months, including three phases. The first phase, lasting one month, is to research and review relevant papers and datasets, then come up with a final decision on the objectives, scope, and plan of the study. The second phase, taking two months, starts in the second month of the project to apply machine learning onto the selected dataset. Data was prepared by cleaning, exploring, visualising to understand it. Then, models will be built and compared to select the best one. In the third phase, everything will be documented into a report, which take up to one month. Detailed plan is described in the Gantt charts in figure 8, 9 and 10.

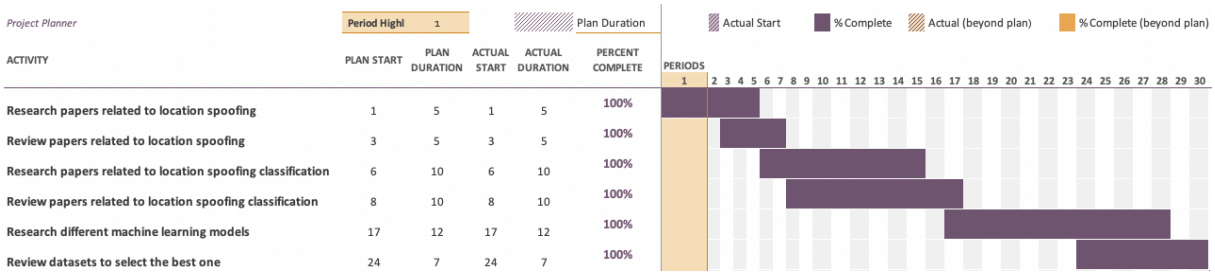

Figure 44. Gantt chart of first phase

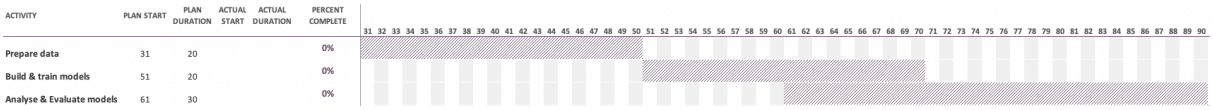

Figure 45. Gantt chart of second phase

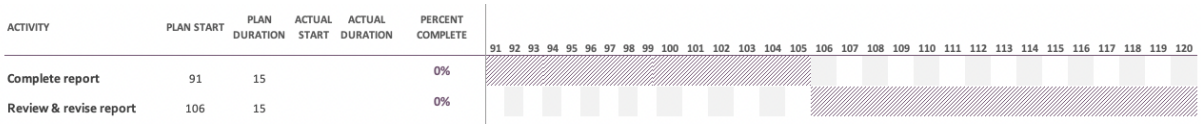

Figure 46. Gantt chart of third phase

APPENDIX B: RESEARCH PROPOSAL

FAKE GPS DETECTION USING MACHINE LEARNING MODELS

LINH VUU

MSc Data Science

Research Proposal

JULY 2022

Abstract

GOJEK is a location-based mobile application providing transportation services using the Global Positioning System (GPS) to detect the drivers and the customers' position. Due to the high competition among the drivers and attractive bonuses paid from the company to the most active drivers, many drivers try to use Fake GPS location application to spoof their locations to receive more orders. As a result, the honest drivers would be impacted detrimentally as they lose their chances to approach customers. To deal with this problem, this proposal suggests using machine learning algorithms to detect spoofing cases using GOJEK public dataset. Several models, including Logistic Regression, Decision Tree and Random Forest will be applied to the dataset and evaluated based on the specificity and accuracy to select the best one. This research would have a significant impact on GOJEK company as they can protect innocent drivers and reduce the waiting time for the customers by eliminated cases where drivers are too far away but fake location to be near them. Further research in other industry can also use this implementation as an example of how to detect location spoofing. This project would need Jupyter Notebook, Anaconda and will take around four months to go from the research phase to the modelling and the final stage of reporting.

Table of Contents

1. Background

More than 70% of people around the world have smart phones with Global Positioning Systems (GPS) and Wifi capabilities as a result of technology evolution (Haddad et al., 2014). Using cellular network stations, GPS and wifi positioning, a lot of mobile applications can acquire locational information to provide location-based services. One of cell phone applications applying location navigation features to transportation successfully is GOJEK (Mar'i and Pangestu, 2021).

Moslehpour et al. (2022) informed in their paper that GOJEK Indonesia is one of the largest technology companies in the mobile commerce. Established in 2010, GOJEK has evolved from a ride-hailing phone service into an advanced technology firm providing location-based services on mobile application, both transportation and delivery. 7.1 billion USD was contributed by GOJEK to economy in Indonesia in 2019 (Florene, 2020) and received investment from large companies such as Google and Mitsubishi (Chao, 2020). In 2021, Mar'i & Pangestu (2021) noted that the GOJEK application had almost 2 million driver partners and 200 million downloads, showing that this application has been known commonly.

According to Nandi (2019), in their online survey conducted between September and November of 2017, 96% of responses admitted that they used motorcycle application in their daily activities. 53% of them chose GOJEK over Grab and Uber, which demonstrated a huge demand for GOJEK in recent years. As a result, the number of GOJEK drivers also increased significantly. Unemployed people became GOJEK drivers because the job does not require having a university degree. Also, a lot of employees resigned from their jobs to be drivers due to the attractive income and the flexible working hours. In fact, according to a survey conducted by Hidajat et al. (2021) sixty per cent of drivers chose to work full time because they had no other job and forty per cent of them previously had jobs but then quit and became drivers.

Apart from the advantages, there are several issues in this industry, including frauds committed by drivers (Damaini et al., 2018). To take benefits from the applications without moving, drivers use a fake GPS application to make their phones locating them in different places from their current positions. This would help them to reach the customers faster or even create fake trips, which is unfair for other drivers working honestly. Interestingly, in 2018, the Institute for Development of Economics and Finance surveyed 516 online drivers and found that 61% of drivers knew that their fellows were cheating (Hidajat et al., 2021).

To solve this issue, this paper applies machine learning, a computer science field that studies a given data, defines the patterns and makes predictions on the new dataset (Simon et al., 2016). According to Lim (2019), machine learning was leveraged to deal with problems in logistics, economics and transport, including dispatching the right drivers, estimating the correct prices, recommending the favourite food and building trust with their customers. Thanks to that, they had more than 20 products in 4 countries with hundreds of million orders per month (Lim, 2019). To categorise Fake GPS cases in GOJEK dataset, Decision tree, Random Forest Classifiers and Logistic Regression will be applied on GOJEK dataset in this research to see which one is suitable for fraud detection in the GOJEK case since they are common algorithms using for binary classification (Bahel et al., 2020).

2. Related Research

Global Positioning System (GPS), the only completely functional Global Navigation Satellite System (GNSS), was found and implemented by the United States Department of Defense. Twenty-four satellites orbit the earth, transmit signals to devices with GPS function, determining their location, direction, and speed (Maddison and Ni Mhurchu, 2009). The positions those devices are calculated by the distance between themselves and three or more GPS satellites, using trilateration, a mathematical technique. Nowadays, GPS plays a significant role in life with a wide range of applications, such as security, navigation system, commercial and scientific research (Rustamov, 2018). The more popular positioning technology is, the harder people try to manipulate it with different location spoofing techniques. To illustrate, the location was spoofed to spread fake news in society (Maheshwari, 2016), to prevent protestors from being attacked by Egyptian police (Lotan et al., 2011), and to mislocate ships in the Mediterranean Sea (Anon, 2013). Location spoofing was used increasingly in the Internet of Things (Koh et al., 2016) and in cyber warfare (Lee and Buehrer, 2012). Players spoofed location in Pokemon Go, a location-based game, to catch more pokemon and receive more in-game items (Zhao & Zhang, 2019).

There are several papers investigated how machine learning can be used to detect location spoofing. To detect spoofing location in social media, Zhao & Sui (2017) developed a Bayesian time geographic method using millions of geo-tagged tweets. Wong & Yiu (2020) recommended a behavioral detection approach using the accelerometer and gyroscope commonly attached in most of the mobile devices. They verified whether the GPS location was

real or not by comparing with the direction extracted from gyroscope data and examined the number of steps collected from accelerometer in order to confirm if the movement' step length aligns with the approximate length of adult's steps. Jiang et al. (2021) proposed a deep learning model, called DeepPOSE to realise GPS spoofing attempts on mobile environments. Neural network was used to eliminate the noise, recovering a vehicle's path from several sensor inputs, then suggested a new strategy to predict the real destination from sensor readings on the Google Map. Then, the reconstructed path is used for GPS spoofing detection. Figure 1 demonstrates the GPS spoofing on vehicle movement.

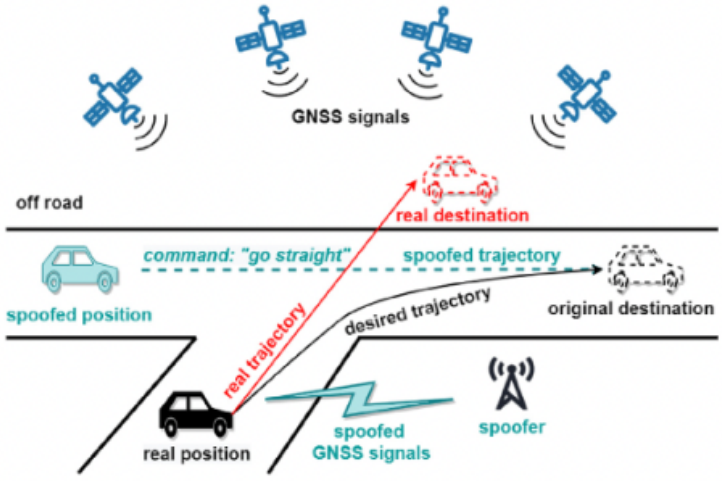

Figure 47. GPS spoofing on vehicle (Peng et al., 2021)

To determine whether an order was done with fake GPS apps or not based on the drivers' ping behaviors in a GOJEK public dataset, Mar'i & Pangestu (2021) proposed using a machine learning algorithm called Logistic Regression. Their model reached a precision value at 74.7%, recall at 99.4%, an accuracy score of 74.7% and F1 scored 85.5%. Although recall had a high value, it would be better to have a higher specificity and accuracy score because in the GOJEK case, the aim is to have the rate of actual fake gps orders (negative cases) correctly predicted as highest as possible.

3. Aim and Objectives

The principal purpose of this paper is to use the same GOJEK public dataset with different analysis approaches attempting to improve the algorithm, evaluate performance of other machine learning models and produce better results, increasing the rate of correctly defining

fake gps trips. The successful identification of spoofing location using trips' details allows technology transportation companies to treat their driver partners fairly and to reduce losses in ghost trips where their partners had the capabilities of teleportation.
This study aims for the three purposes:

- To find patterns by analysing relationships between the details of drivers ping signals.
- To build different models to automatically decide whether an order is real or not.
- To compare and evaluate the performance of models, then select the most accurate model.

5. Significance of the Study

This study helps to define fake gps rides, thus, it brings benefits not only to online transportation service providers, their customers, and all legal drivers but also to other location-based applications.
By recognising fraud driving activities, the companies can reduce their losses when not giving rewards to spoofers and all honest drivers can have opportunities to earn bonuses through performance incentives. For example, GOJEK drivers can earn Rp. 200,000.00 when collecting 30 points on GoRide service and earn Rp. 400,000.00 with 21 points on GoCar service (Jakarta, 2019). This leads to drivers using multiple phones, accounts and spoofing location applications to artificially create and deliver driving services (Khong, 2019). Recently, Grab suffered a case of payment fraud on Grabhitch, losing $41,000 (RM165,000) and arrested ten suspects while according to Istiardi & Setiawati (2022), GOJEK lost Rp. 500,000,000.00 paying to perpetrators committing fraud orders.
When stopping drivers to fake their locations to approach customers, the companies can ensure a true waiting time for the customers and a fair working environment for other drivers. Foo Wui Ngiap, head of trust, identity, safety and info security at Grab indicated that drivers used spoofing apps to be located near customers when they were actually not (Khong, 2019). Because of that, nearby drivers could not take that ride and the fraudsters took more time to come to the passengers compared to what Grab app estimated. As a result, other drivers would lose their income and customers would not be happy and think that Grab allocate rides inaccurately.

This research is also helpful for other location-based applications, like Pokémon Go, StepN, as a reference to detect location spoofing based on users' behaviours. Although other applications can collect different data and users can have different behaviours, some approaches to detect location spoofing and algorithm used in building and comparing models can be utilised to prevent spoofing in general.

6. Scope of the Study

The scope of this study is to use GOJEK published dataset to predict fake location cases based on drivers' behaviours. The limitation of the scope is that the dataset only contains a few features describing the orders. Other data can also be collected from the application to define whether the location is spoofed or not, such as the devices' information, moving velocity and vertical accuracy. However, those details can be either very hard to collect by the software or so confidential that cannot be published to the community for researching purposes. Nevertheless, it is not in the scope of this dataset and this paper.

7. Research Methodology

7.1 Introduction

Methodology consists of selecting data, pre-processing the dataset, transforming data into a structured and comprehensible, balancing the dataset, applying machine learning techniques to build models and evaluating the models using evaluation measures. These steps are described in the figure 2.

Figure 48. Flow chart of the implementation

7.2 Dataset description

Data used in this research is an anonymised dataset on Kaggle (2019) of PING signals from GOJEK partner drivers within 47 days with 567,545 rows, containing numeric feature variables, a string order_id column, and a binary target column.

Features description:

- order_id - an anonymous id unique to a given order number
- service_type - service type, can be GORIDE or GOFOOD
- driver_status - status of the driver PING, can be AVAILABLE, UNAVAILABLE, OTW_DROPOFF, OTW_PICKUP
- date – from 05 February, 2018 to 23 March, 2018
- hour – time in hours
- seconds – time in seconds (in linux format)
- latitude - GPS latitude
- longitude - GPS longitude
- altitude_in_meters - GPS altitude in meters
- accuracy_in_meters - GPS accuracy in meters, the smaller the more accurate

Target:

- label - label describing whether GPS is true (1) or fake (0)

7.3 Data Preparation (Preprocessing and Transformation)

To begin, the data should be cleaned up from the dataset as the flow figure 3.

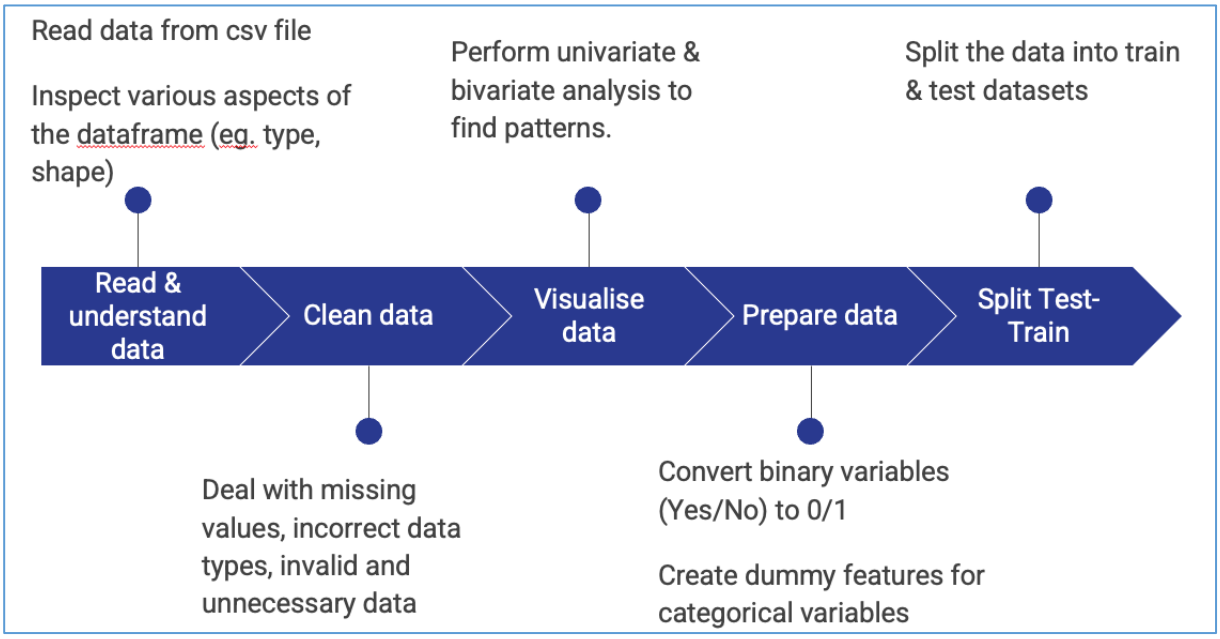

Figure 49. Flow chart of data preparation

After reading the data from the CSV file, missing values, incorrect data types, invalid and unnecessary data will be dealt with. Those types of data will be examined to see if they are associated with the target column. If so, they can be replaced using simple imputation approach. In a survey paper published in 2021, (Emmanuel et al., 2021) mentioned that most of the papers used simple imputation to handle missing data by mean, median or mode of the variables. Otherwise, when the columns with missing data do not have any impact on the target, they can be eliminated with pairwise deletion which is known to produce low bias results for both missing completely at random and missing at random data.

Then, data will be visualised using Exploratory Data Analysis techniques to find patterns by comparing differences between fake and real orders. As suggested by (Salgado et al., 2016), line plot and histograms will be used to explored univariate continuous and categorical variables respectively. On the other hand, scatter plot is suitable to visualise relationship between variables.

Next, new features that may fit better on data modelling can be created from existing variables.

- 'Date' attribute will need to be changed. Here, If the numbers from one to seven are used for the days of the week from Sunday to Saturday. It can be seen that day 1 (Sunday) and 7 (Saturday) are very far to each other, but they should not be because Sunday is near Saturday. Dossman (2017) revealed that such cyclical feature should be

encoded using sin/cos transformation so that the model can use it effectively as presented in the equation (1) and (2) below.

$$day\_cos = \cos\left(2 * 3.14 * \frac{day(numeric)}{7}\right)$$

Equation 1. Cos transformation

$$day\_sin = \sin\left(2 * 3.14 * \frac{day(numeric)}{7}\right)$$

Equation 2. Sin transformation

Besides that, a new variable called 'is_weekend' will be created and use as an indicator of weekdays and weekend.

- 'Hours' attribute will be transformed to is_working_hour, which will have '1' in the value when the 'hours' is between 9 am and 6 pm. Otherwise, it will be '0'.
- 'Seconds' is in Linux format; thus, it will be converted to datetime format and will be compared with the 'Date' column to see if they are the same and if 'Seconds' has any meaning or is useful for the model.
- 'Latitude' and 'longitude' will be used to calculate the distance between two consecutive pings with the "Haversine" method, according to Hijmans (2021), working best for small distances.

At the final step of this stage, data are split into train and test data before building models. Train data will be used to train the models, while test data will be for models testing and evaluations.

7.4 Modelling Techniques

After cleaning data, models will be built as the following chart. Before applying any algorithm, feature scaling is important in machine learning to bring all numeric variables in the same standard, so that one significant value would not have any impact on the algorithm because of their magnitude (Baijayanta, 2020).

Then, the model will be formed using Logistic Regression. Peng et al. (2002) pointed out that Logistic Regression can figure out how dependent variable depends on independent variables. This algorithm collects data into different groups and each group represents a class of each data type. The data will then be divided into a variety of categories by a logit function like the equation (3).

$$logit(x) = \log\left(\frac{x}{1-x}\right)$$

Equation 3. Logit function

Another model that can be built is Decision tree. This method divides a dataset into branch-like segments that construct a tree with nodes and branches (Song & Lu, 2015). It starts with a root decision node, askes a Yes/No question and expands into many subtrees. This method is easy to understand as it is similar to what human usually do when making decisions. In fact, it compares the attributes of a record in the dataset with the values of decision node and based on the comparison, jumps to the next node by following the branch. At the next node, the algorithm compares the value of the attribute with the other nodes again and move further. The process is continued until the leaf node of the tree is reached. The diagram in figure 4 is a graphical illustration of a Decision Tree Algorithm.

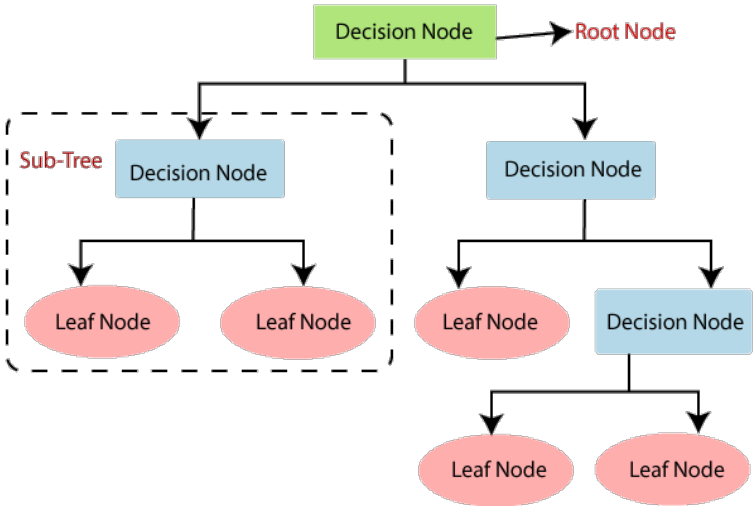

Figure 50. Decision Tree (Arain, 2021)

A decision tree with a number of layers can lead to overfitting issues, which can be solved using Random Forest Classifiers, an ensemble classification algorithm consisting of many decisions trees (Fawagreh et al., 2014). Each tree acts as a classifier to define the label, then the label

which has the highest number of votes is used to decide the label of the instance as illustrated in figure 5.

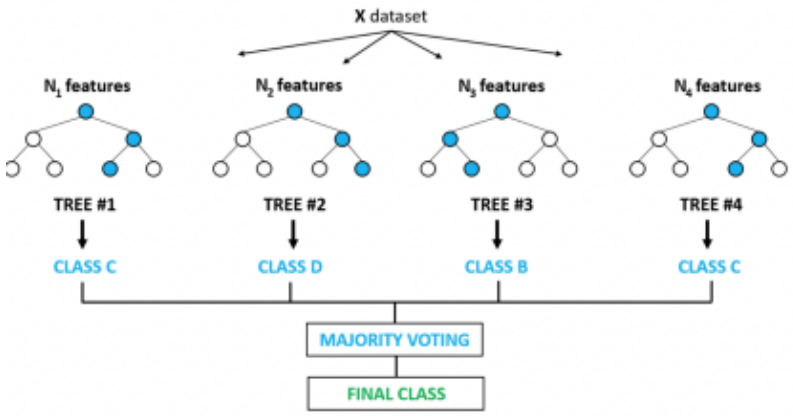

Figure 51. Example of ensemble of decision trees (random forest) (Kirasich et al., 2018)

7.5 Evaluation Metrics

To compare the models, first, a confusion matrix is needed, to see how the system mislabels the true and fault value of the predicted variable (Powers, 2020). The matrix, as in figure 6, shows the number of cases where real GPS data predicted correctly (true positive), the number of cases where the GPS data is real, but the system predicted as fake (false negative), the number of cases where the fake GPS data is predicted correctly (true negative) and the number of cases where the GPS data is fake but the system predicted as real (false positive). Then, these four variables will be used to calculate the accuracy, specificity and recall to evaluate the performance of the models.

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Figure 52. Confusion matrix (Wikipedia, 2022)

Accuracy is used to calculate the correct records compared to the total records as equation 4. As a result, a high accuracy value proves that the classification prediction is good.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

Equation 4. Accuracy equation

Recall (true positive rate) is the proportion of positive cases predicted correctly as positive, calculated by the equation 5. A high recall value means that there is a high ratio of real gps cases predicted correctly.

$$Recall = \frac{TP}{TP + FN} * 100\%$$

Equation 5. Recall equation

In contrast, specificity (true negative rate) is the proportion of real negative cases predicted as negative correctly, calculated by equation 6. The higher the specificity value, the more fake gps cases are predicted exactly.

$$Specificity = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

Equation 6. Specificity equation

Therefore, finally the model with high specificity and accuracy will be selected as the purpose of the study is to have the rate of actual fake gps pings correctly predicted as highest as possible. The evaluation flow is shown in the figure 7 below.

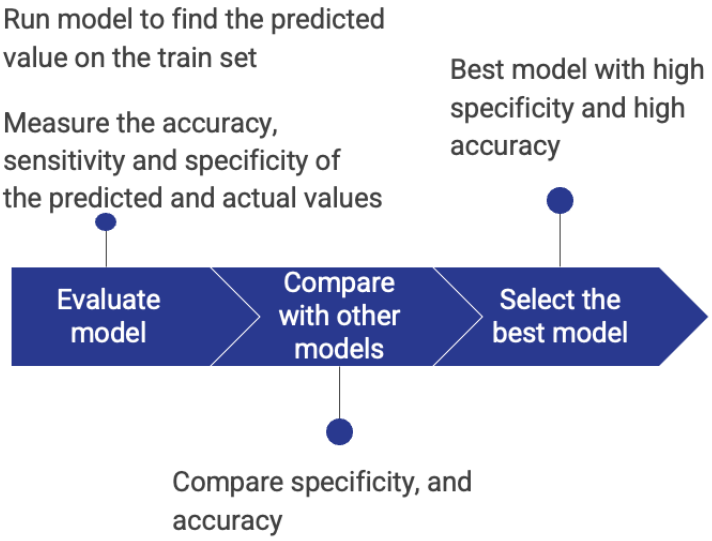

Figure 53. Evaluation flow chart

8. Requirements Resources

In order to analyse data and build models, Jupyter Notebook needs to be setup. **Python** is a requirement (Python 3.3 or greater) to install Jupyter Notebook. Anaconda is recommended to install Python and Jupyter. As mentioned on 2015), with Anaconda, Python, Jupyter Notebook, and other common packages for data scientific can be installed conveniently.

Anaconda is freely distributed as the terms of the EULA for Anaconda Distribution (Anaconda, 2022). It can work well on 64-bit macOS 10.13+, Windows 8 or newer, or Linux, including Ubuntu, RedHat, CentOS 7+, and others. Disk space with minimum 5 GB is required to install Anaconda.

9. Research Plan

This research is expected to be done within four months, including three phases. The first phase, lasting one month, is to research and review relevant papers and datasets, then come up with a final decision on the objectives, scope, and plan of the study. The second phase, taking two months, starts in the second month of the project to apply machine learning onto the selected dataset. Data was prepared by cleaning, exploring, visualising to understand it. Then, models will be built and compared to select the best one. In the third phase, everything will be documented into a report, which take up to one month. Detailed plan is described in the Gantt charts in figure 8, 9 and 10.

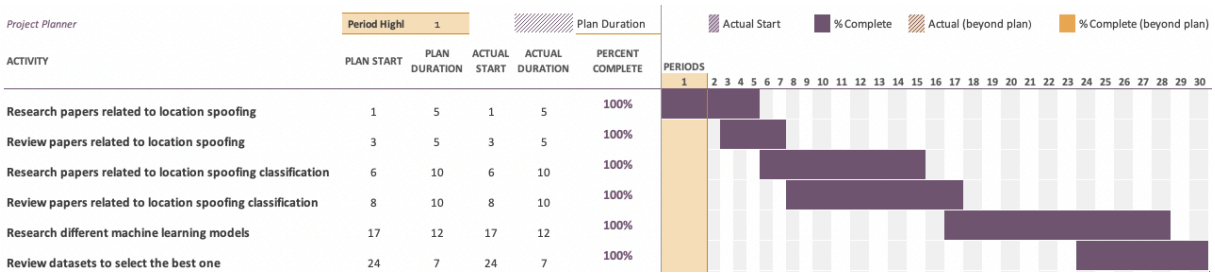

Figure 54. Gantt chart of first phase

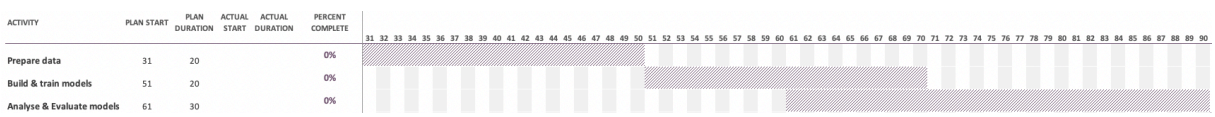

Figure 55. Gantt chart of second phase

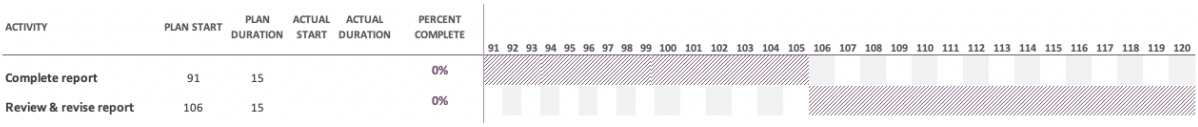

Figure 56. Gantt chart of third phase