Explainable Machine Learning for Autonomous Vehicle Positioning using SHAP

Uche Onyekpe[1,2*], Yang Lu[1], Eleni Apostolopoulou[1], Vasile Palade[2], Eyo E.U.[3], Stratis Kanarachos[4]

[1] *School of Science, Technology and Health, York St John University, York, YO31 7EX, UK*

[2] *Centre for Computational Science and Mathematical Modelling, Coventry University, Priory Road, Coventry, CV1 5FB, UK*

[3] *Faculty of Environment and Technology, Civil Engineering Cluster, University of the West of England, Bristol, UK*

[4] *Faculty of Engineering and Computing, Coventry University, Priory Road, Coventry, CV1 5FB, UK*

Email: *u.onyekpe@yorksj.ac.uk, y.lu@yorksj.ac.uk, eleni.apostolopoulou@yorksj.ac.uk, ab5839@coventry.ac.uk, eyo.eyo@uwe.ac.uk, ab8522@coventry.ac.uk*

Abstract

Despite the recent advancements in Autonomous Vehicle (AV) technology, safety still remains a key challenge for their commercialisation and development. One of the major systems influencing the safety of AVs is its navigation system. Road localisation of autonomous vehicles is reliant on consistent accurate GNSS (Global Navigation Satellite System) positioning information. The GNSS relies on a number of satellites to perform triangulation and may experience signal loss around tall buildings, bridges, tunnels, trees, etc. We previously proposed the Wheel Odometry Neural Network (WhONet) as an approach to provide continuous positioning information in the absence of the GNSS signals. We achieved this by integrating the GNSS output with the wheel encoders measurements from the vehicle whilst also learning the uncertainties present in the position estimation. However, the positioning problem is a safety critical one and thus requires a qualitative assessment of the reasons for the predictions of the WhONet model at any point of use. There is therefore the need to provide explanations for the WhONet's predictions to justify its reliability and thus provide a higher level of transparency and accountability to relevant stakeholders. Explainability in this work is achieved through the use of Shapley Additive exPlanations (SHAP) to examine the decision-making process of the WhONet model on an Inertial and Odometry Vehicle Navigation Benchmark Data subset describing an approximate straight-line trajectory. Our study shows that on an approximate straight-line motion, the two rear wheels are responsible for the most increase in the position uncertainty estimation error compared to the two front wheels.

Keywords – Wheel odometry, Autonomous vehicles, Inertial navigation system, Deep learning, Explainable machine learning, GNSS outage, Positioning, Neural networks

1 Introduction

1.1 Motivation for Autonomous Vehicles

The potential safety benefits of Autonomous Vehicles (AVs) have long been regarded as one of the technology's biggest assets [1]. According to [2], human error accounts for 75% of traffic-related road accidents in the UK, and 94% in the USA. There is the potential to significantly reduce these road accidents by minimising or eliminating the involvement of humans in the operations of vehicles [3]. This has been a strong selling point for self-driving vehicles to a public which, so far, seems unwilling to trust the technology [1]. However, the introduction of AVs could introduce new kinds of accidents. Such safety concerns can affect the customers' intention to use AVs [4], and are majorly responsible for the currently delay in the commercialisation of the vehicles [1].

AVs acquire an understanding of their environment through the use of sensory systems [5]. Ultrasonic systems, LIDARs and cameras are examples of such sensors that can be found on the outside of the vehicle. Cameras and LIDARs are used in the identification of objects, structures, potential collision hazards and pedestrians in the vehicle's path[6]. Cameras are also essential in identifying signs and road markings on structured roads. An indication of the critical role of imaging systems in the operation of autonomous vehicles is clearly noticeable from the numerous sophisticated versions of these systems currently being employed [7]. Despite the importance of imaging systems in the assessment of the vehicle's environments (e.g., determination of markings, vehicle-object relative positions, etc), there is the need to localise a vehicle robustly and continuously with reference to a well-defined coordinate system for real-time positioning and decision making.

1.2 Global Navigation Satellite System (GNSS)

The GNSS receiver uses signals from at least three satellites orbiting Earth to localise the vehicle to a road [8]. In spite of its wide acceptance for positioning as it is unrivalled in terms of cost and coverage, the GNSS is not a perfect positioning system. The GNSS requires a direct line of sight between the GNSS antennae and the satellites to perform localisation, however, in metropolitan areas and similar environments, the line of sight can be blocked by features such as tall buildings, skyscrapers, bridges, dense tree canopies or road tunnels (Yao et al., 2017). Furthermore, the signal from the GNSS could be jammed, leaving the vehicle with no position information [9]. Hence, the GNSS receiver cannot serve as a standalone system of vehicle positioning.

The GNSS enables road localisation of the AV, to localisation the vehicle to a lane, the GNSS is implemented in combination with high accuracy LIDARs, cameras, RADAR and High Definition (HD) maps. There are however instances when the LIDAR and camera could be unavailable for use or uninformative. The usage of low-cost cameras and LIDARs could compromise their functions and accuracy especially during extreme weather events such as blizzards, heavy snow fall, rain,

[*] Corresponding author Uche Onyekpe: u.onyekpe@yorksj.ac.uk

fogs, or sleet [10]. These issues are well-known in the field. Whilst the acquisition of LIDARs of high accuracy, could render it vulnerable to theft and further increase the cost of the AV (Lee Tescher, 2018), camera-based positioning systems may suffer low accuracies depending on the objects in the cameras scene and the external light intensity (Lee Tescher, 2018).

According to tests performed by Cruise LLC and Waymo LLC on level 4 self-driving vehicle applications, the LIDARs scan is matched in real-time unto a High Definition (HD) map (Lee Tescher, 2018). As a result, the system is capable of precisely positioning the vehicle within its surroundings (Lee Tescher, 2018). Nonetheless, the downside of this method is its high computational cost. Moreso, changes in infrastructures within the driving environment could render an HD map temporarily obsolete and thus not effective for navigation.

Tesla which is well known for its no LIDAR and HD map policy. It handles GNSS signal outages by relying on its cameras and road markings until the GNSS signal becomes available. But the question is what happens if a decision is needed to be made on navigating to a new road during the signal loss or what happens when the GNSS signal is lost, and the camera is uninformative? Failure Mode and Effect Analysis (FMEA), which is an analysis performed to identify all the ways a system can fail and identify ways to mitigate them, would need to be performed on all the above failure scenarios to provide a number of fail-safe options to support the safe operation of autonomous vehicles.

1.3 Navigation using Inertial Measurement Sensors

The use of high accuracy Inertial Measuring Unit (IMU) has been proven to be a way to overcome the GNSS reliability issue [11]. The IMU measures the AVs rotational rate and linear acceleration in the x, y and z-axis and computes its orientation, position, and velocity information by continuous dead reckoning. The significant cost of such IMU sensors has however hindered their adoption on autonomous vehicles. Even more, low-cost IMU's have accuracies too low to be used independently on autonomous vehicles as they are plagued by noise and biases which are exponentially cascaded over time for instance during the double integration from acceleration to position [12]. In what is usually regarded as a symbiotic relationship, the GNSS can periodically calibrate the Inertial Navigation System (INS) during signal coverage to improve the position estimation accuracy of the INS during the GNSS outages.

Several researchers [13]–[16] have studied the use of machine learning-based techniques to model the errors and learn the non-linear relationships that exists within the sensor's measurement. Such proposed techniques include Recurrent Neural Networks (RNN) based models in [15]–[20], Multi Feedforward Neural Network (MFNN) based models in [13], [21]–[24], Radial Basis Function Neural Network (RBFNN) based models in [25], [26] and the Input Delay Neural Network (IDNN) in [14]. Despite the numerous research into improving the performance of low-cost INS, the issue remains a challenge in need of cost-effective solutions.

1.4 Inertial Positioning using Wheel Encoder Sensors

Modern vehicles are embedded with a number of sensors that support several advanced driver-assist systems, such as the wheel encoder of the Anti-lock Braking System (ABS). The wheel encoder which operates by measuring the vehicle's wheel or axle speed, has been explored as an alternative to the commonly used low-cost accelerometer of the INS for vehicle positioning [27]. The wheel encoder provides a better position estimation solution compared to the accelerometer as its resolving requires one less integration step in the computation of the vehicle's position, thus minimising the error propagation. Nevertheless, the wheel encoder-based solution is not a perfect one either. The accuracy of the wheel encoder-based position estimation is affected by factors such as changes in the sizes of the tyres and wheel slippages. [28]. A smaller tyre diameter due to a reduction in the tyre pressure or a tyre replacement, leads to an underestimation of the vehicle's displacement vehicles displacement (Onyekpe et al., 2020b), Whereas A larger tyre diameter leads to the vehicle's displacement being overestimated (Onyekpe et al., 2020b).

Reference [28], showed that the errors present within the position estimation obtained from the wheel speed data can be learned by the Long Short-Term Memory (LSTM) neural network even in complex driving environments such as roundabout, successive left and right turns, wet roads, etc. In [19], a Wheel Odometry Neural Network (WhONet) was proposed and shown to provide better estimations in both complex driving scenarios and longer-term GNSS outages of up to 180s with an accuracy averaging 8.62m after 5.6km of travel.

1.5 Motivation for Explainability

Despite the remarkable performance of machine learning on the vehicle positioning problem, there is the requirement of transparency and higher level of accountability from the machine learning based system designs. Explanations for machine learning model's decisions and estimations are thus needed to justify their reliability. This requires greater interpretability, often requiring an understanding of the mechanism underlying the operation of the algorithms. Unfortunately, the blackbox nature of Neural networks is still unresolved, and many estimations are still poorly understood. Commonly, the eXplainable Artificial Intelligence (XAI) procedures consist of ensemble runs, random sampling and Monte-Carlo simulations, which are quite common methods in engineering. XAI comprises of a systematic perturbation of some components of the model, which enables it to observe how it affects the model's estimates mostly using sensitivity analysis. Due to the safety critical nature of the autonomous vehicle navigation systems, interpretability of the vehicular navigation models is necessary and provides sufficient argument for the suitability of a model for use on the road as well as sufficient argument when communicating anomaly behaviours to insurance stakeholders, Original Equipment Manufacturers (OEM) and other relevant stakeholders. We therefore explore in this research the interpretability of the WhONet models in estimating the position of Autonomous vehicles in the absence of GNSS signals.

2 eXplainable Artificial Intelligence (XAI): Background and Current Challenges

2.1 Why XAI (Significance)

Neural Network based models are built on complex non-linear functions and are commonly heavily parameterised [29], [30], [31]. However, the high non-linearity feature as well as complexity of algorithms makes it difficult to understand the internal working mechanisms. More importantly, such opaqueness can create distrust in Artificial Intelligence (AI) based applications. For instance, passengers may feel extremely anxious when sitting in the self-driving cars if the behaviours are not self-explanatory, e.g., a car suddenly turns around at an intersection whereas it normally passes it without explanation. Besides, the AI based models can take wrong actions due to biases in training data. This may cause catastrophic and even life-threatening consequences in medical diagnosis and treatment. As a result, the eXplainable Artificial Intelligence (XAI) becomes highly demanded to interpret models' decision-making and working mechanisms.

Explainability can enable good understanding of a model from different aspects, bringing insights that can be adopted by different stakeholders involved [32]. Figure 1 shows what positive effects can be brought by explainability to stakeholders. For instance, data scientists can easily debug an AI based model, adjust the parameters so as to improve performance, while business owners may care more about whether a model will fit with the business strategy and investment purpose. Risk analysts will need to check the robustness and decide on the deployment of the model, and regulators can evaluate whether a model is reliable as well as what impact can be triggered by its decision on the customers. Finally, consumers can demand for transparency in terms of how decisions were taken. Specifically, explainability in the development of AI approaches can help addressing different critical concerns [33]. In the example about autonomous vehicles, the passengers can trust the automated decision if the car turns around with an explanation such as "*a car accident is detected 200 meters in front of us. We will choose another route from the previous exit which can take 10 minutes more than the usual route*" [34].

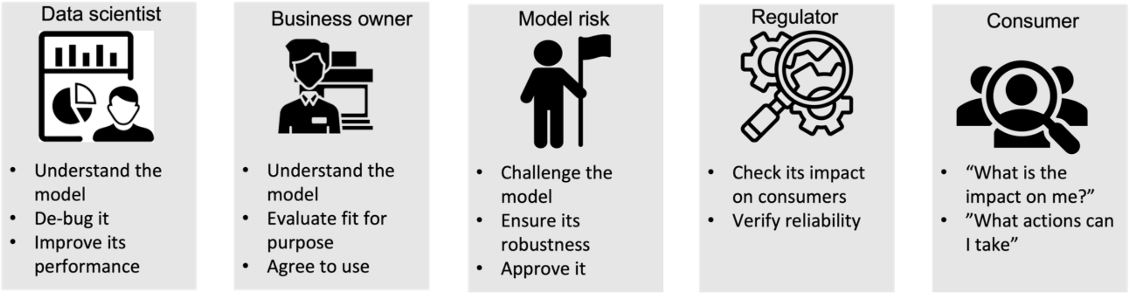

Figure 1. Concerns faced by stakeholders

Table 1. Key stakeholders (Belle & Papantonis, 2021).

Features	**Implications**
Correctness	How confident are we that the variables contributing to the decision making are all and only those of interest? How confident are we that the spurious patterns and correlations have been remove?
Robustness	How confident are we that the model is not vulnerable to minor perturbations, and if so, can it be justified for that outcome? How confident are we that the model does not misbehave in the presence of noisy and missing data?
Bias	are there any biases in the data penalises any group of individuals unfairly? And if yes, can they be identified and corrected?
Improvement	How can the model's prediction be improved concretely? What are the effects of having an enhanced feature space and additional training data?
Transferability	How can the prediction model be generalised from one application domain to another? What properties of the model and data are needed to facilitate the transferability of the model to other domains?
Human comprehensibility	Can the algorithmic machinery of the model be explained to an expert and perhaps a lay person? Is the model's explainability needed for a wider deployment of the model

2.2 What is XAI

2.2.1 Attributes of explainability

Explainability attributes should contain the criteria and characteristics that researchers could use to define the explainability construct. Firstly, it is necessary to make it clear and explicit to the end-users what casual relationships exist between the inputs and the model's predictions [35]. According to [36], the explanation of the logic of an inferential system can help to justify, control, discover and improve the learning algorithm. In [37], Interpretation is referred to as the mapping of an abstract concept (as a predicted class) to a domain that the human can understand. However, an explanation contains all features of a domain that can contribute to making a prediction [37]. Given the definition of interpretability or explainability as "*the degree to which a human observer can understand the reason behind a decision made by the model*" in [38], both notions can be interchangeable.

2.2.2 Theoretical approaches for structuring explanations

Structuring an explanation for ad-hoc applications can involve decision on what information is included or excluded, e.g., causes, contexts, and consequences of the predictions from a model [39]. Some researchers created a classification system for different explanation types, which can be suitable for different learning algorithms in terms of logic interpretation [40]. In addition, de Graaf & Malle in [41] identified that different types of users, problems, and behaviours require different explanations, as illustrated in Figure 2 . With the focus on user types, Glomsrud et al. in [42] summarised four explanation categories, ordered by the levels of completeness required by different user groups, i.e., explanations for developer, assurance explanations, explanations for end-users as well as external explanations. Stevens et al. in [43], proposed three classifications of the types of explanations, the first, called Mechanistic operation which attempts to answer the question “How does it work?”. The second was referred to as ontological explanations which describes the structural properties of the model such as it attributes and components, and how they relate. The third type was referred to as operational explanations which attempts to answer the question “How do I use it?”. Moreso, Sheh and Monteath in [44], provided a more articulated classification of the types of explanations of intelligent systems to include teaching explanations, introspective tracing explanations, introspective informative explanations, post hoc explanations, and executive explanations. Besides, Barzilay et al. in [45] proposed a classification of the knowledge that should be embedded in an explanation [35],[46]. Finally, Sohrabi et al. in [47] introduced a formal framework to generate preferred explanations for a given plan. It is necessary to contextualise the explanation preference to the observational patterns. Certain causes may affect the action, and thus require the explanation to reflect on the past, which means that produced explanations should consider the past events and data.

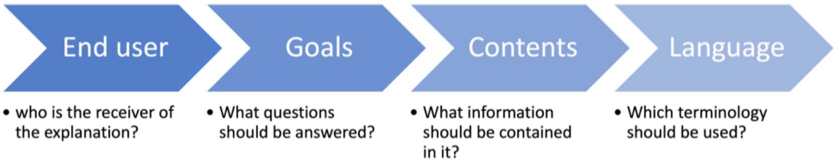
End user
Goals
Contents
Language
• who is the receiver of the explanation?
• What questions should be answered?
• What information should be contained in it?
• Which terminology should be used?

Figure 2. Diagram of the main factors shaping the structure of a machine-generated explanation

2.3 Types of XAI

Six types of post-hoc explanations for opaque models have been listed [48]. In this section we focus on the most prominent types, which include model simplification, feature relevance, and visualisations.

2.3.1 Explanation by Simplification

Explanations by simplification aims to use a simpler model to approximate an opaque one which can be difficult to interpret. A popular technique is Local Interpretable Model-agnostic Explanations (LIME) [49], which can approximate a complex model. For instance, the complex model can be explained using a decision tree model built around the predictions. Ribeiro et al. in [50] designed a similar technique called anchors, which aims to approximate a model locally by using “if-then” rules. In [51], Krishnan and Wu proposed another simplification approach which seeks to partition the training dataset into similar instances while using a decision tree to structure the explanations. Similarly, Bastani et al. in [52] formulated simplification as an extraction process of using a transparent one to approximate a complex model. Particularly, the proposed method suggests building a greedy decision tree based on the predictions from a black-box model to obtain more insights about the original model which inspecting the surrogate one. Tan et al. in [53] considered the simplification as a way of inspecting if the variable set is sufficient to restore the original one with the same accuracy. Wachter et al. in [54] proposed the counterfactual explanations for creating instances closed to those users are interested in explaining. Through comparing the new data point to the original point, users can obtain insights on what minimal changes should be considered to change the decision made based on the original point.

2.3.2 Explanation by Feature Relevance

Feature relevance explanations attempts to measure the influence of each input and provides a ranking of importance scores, showing which corresponding variables are more importance than others for the model. One of the most significant contributions in this area is SHAP (SHapley Additive exPlanations) [55]. The Shapley feature values refer to the average expected marginal contributions of the features to the model’s decisions. Shapley value has proven to be highly influential in the XAI community. Henelius et al. in [56] designed another method based on feature permutation to identify significant variables or variable interactions that are picked up by the model. Additional ways to assess the significance of a feature can be quantifying the feature importance, transforming all features of a dataset, and achieving a new dataset without the influence of a certain feature [57]. Datta et al. in [58] proposed QII (Quantitative Input Influence) to quantify the influence by estimating the performance change with the use of original dataset, and the new dataset with the feature replaced by a random value. In this research, the explainability of the localisation model is investigated based on explanation by feature relevance using SHAP.

2.3.3 Explanation by Visuals

Visual explanation aims to generate visualisations that allow a better understanding of a model. Existing approaches can help in obtaining insights about the decisions as well as how features interact with each other. Consequently, visualisations can be used to appeal to a non-expert audience. For this purpose, Cortez & Embrechts in [59] proposed a series of plots and discussed additional techniques [60], such as the Sensitivity Analysis approaches. Goldstein et al. in [61] introduced the ICE (Individual Conditional Expectation) and PD (Partial Dependence) plots, which can show insights into the relationship between the interested feature and the outcome (whether it is monotonic or linear, for example) [62]. However, the average effects can be misleading and affect identifying variable interactions. Therefore, a more complete approach would be to utilise both ICE and PD plots, given that a relationship exists between these two plots.

3 XAI in Autonomous Vehicle and Localisation

Autonomous vehicles (AVs) have achieved a significant milestone in research and development over the last decade (Atakishiyev et al., 2021). The significance of the need for XAI has been emphasised as the advanced artificial intelligence techniques are applied in self-driving scenarios (Li et al., 2020). Currently, most of the advanced models of AVs are based on machine learning (Bojarski et al., 2016). Therefore, one of the research streams involves constructing a knowledge base into the AV systems, such as making text-based explanations for the vehicle's behaviour (J. Kim & Canny, 2017) (J. Kim et al., 2018). Meanwhile, some other researchers focus on trust computing of explainable AV models (Mittu et al., 2016); (Petersen et al., 2017); (Haspiel et al., 2018); (Cysneiros et al., 2018). For example, the trustworthiness levels of AV systems can be calculated as a reference for insurance companies and customers (Hengstler et al., 2016).

The demand for explainable AVs creates diverse concerns and issues. Specifically, the occurrence of car accidents is considered a fundamental practical concern. According to Riberio et al (2016), users will not adopt a model or a decision if they don't trust the machine (Ribeiro et al., 2016). With an empirical case study, Holliday et al. (2016) also showed that providing explanations can significantly increase users' trust towards a system (Holliday et al., 2016), however regaining the trust can be onerous if it is damaged in an intelligent system (Kim & Song, 2021). Besides, trustworthiness in the decisions made by AVs can support transparency in the system. Such a positive factor can further develop fairness enabling good ethical analysis and causal reasoning of the decisive behaviours (Arrieta et al., 2020), achieving public approval of automated vehicles. In particular, the real-time decisive actions of AVs involve interconnected operational stages of sensing, localisation, planning and control as discussed below.

1. Sensing: As a primary requirement for the self-driving, sensing refers to road surface extraction and object detection (Pendleton et al., 2017). Differ by the information types can be captured as well as the environment, Perception data can be collected by using devices such as the RADAR, LIDAR, ultrasonic sensors and cameras, (Yeong et al., 2021) (Ahangar et al., 2021).

2. Localisation: Localisation enables an AV to locate its position accurately in the physical world (Woo et al., 2018) (Grigorescu et al., 2020) by comparing the location of reflected objects to the high-definition maps. One of effective ways is to use satellite to get a position of self-driving cars, such as determining a global location of a car by using the Global Navigation Satellite System (GNSS) (de Miguel et al., 2020). In the places like underground tunnels and canyons, alternative sensor technologies like Inertial Measurement Units (IMUs) are used combined with GPS, to navigate, control, and direct a car.

3. Planning: Based upon real-time environmental perception and localisation, an AV can plan its trajectory from the starting point to the destination. Particularly, the motion planning needs to consider the interaction with other vehicles, dynamics of the environment such as people met on a trajectory, as well as available navigating resources and infrastructure. Geisberger et al. proposed the contraction hierarchies in fast routing in (Geisberger et al., 2012). Studies on AV's planning use a variety of different terms for relevant components of the planning process. Overall, the planning of a self-driving vehicle can be made in a hierarchical process including three essential constituents (Paden et al., 2016).

4. Control: A feedback controller in an AV can read inputs from an actuator, fulfil the motion and correct errors brought in by actuation variables. With the aim of calculating the optimal solution for the prediction horizon, the feedback controller can make prediction on motions within a short time interval. Model predictive control has been successfully applied in several control applications, including the combined steering and braking, lane-keeping and navigating in adverse conditions dynamically (Liu et al., 2015)(Falcone et al., 2007) (Borrelli et al., 2006).

In this research, we however focus on XAI for AV localisation.

4 Methodology

In this section, we discuss the dataset employed in this study, the mathematical formulation of the target of the machine learning model, details characterising the optimisation and evaluation of the WhONet model, and the SHapley Additive exPlanation method.

4.1 Dataset: IO-VNBD (Inertial and Odometry Vehicle Navigation Benchmark Dataset)

The IO-VNBD, publicly available at [63] is a large scale inertial and odometry dataset created to facilitate the benchmarking, development, and evaluation of positioning algorithms. The dataset is made up of several simple and complex driving scenarios such as residential road drives, sharp cornering hard brakes, dirt roads, roundabout, town drives, dirt roads, etc., and was collected over 5700km and 98 hours of driving. A Ford Fiesta Titanium vehicle as illustrated in Figure 3 was used to collect the data on public roads within the United Kingdom. The dataset contains information describing the dynamics and position of the vehicle such as the speed of the vehicle's wheel (in rad/sec) and GPS coordinates (in degrees) which were extracted from the vehicle's Electronic Control Unit (ECU) at a sampling frequency of 10 Hz. In this research, the V-Vw12 IO-VNB data subset which describes on a motorway within the UK is used. For more information on the IO-VNBD, please see [64].

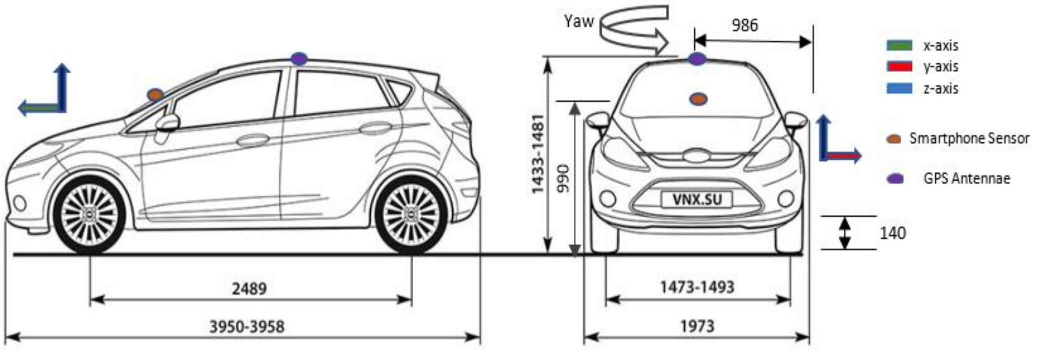

Figure 3. *Data collection vehicle, showing sensor locations* [64].

4.2 Mathematical Formulation of the Learning Problem

The angular velocity of the vehicle's wheels at any time (t) are measured by the wheel encoder. However, there can be uncertainties in the wheel encoders measurements and the state of the tyres due to tyre wearing, changes in tyre pressure and wheel slips. The accuracy of the displacement estimation from the wheel encoder's measurements ω are affected by these uncertainties.

Equations (1) – (4) considers the errors that could affect the calculation of the vehicles wheel speed.

$$\widehat{\omega}^b_{whrl} = \omega^b_{whrl} + \varepsilon^b_{whrl} \tag{1}$$

$$\widehat{\omega}^b_{whrr} = \omega^b_{whrr} + \varepsilon^b_{whrr} \tag{2}$$

$$\widehat{\omega}^b_{whfl} = \omega^b_{whfl} + \varepsilon^b_{whfl} \tag{3}$$

$$\widehat{\omega}^b_{whfr} = \omega^b_{whfr} + \varepsilon^b_{whfr} \tag{4}$$

Where $\widehat{\omega}^b_{whfr}$, $\widehat{\omega}^b_{whfl}$, $\widehat{\omega}^b_{whrl}$, and $\widehat{\omega}^b_{whrr}$ are the noisy wheel speed measurements of the front right, front left, rear left, and rear right wheels; whereas ε^b_{whfr}, ε^b_{whfl}, ε^b_{whrl}, and ε^b_{whrr} are the corresponding errors (uncertainties); and, ω^b_{whfr}, ω^b_{whfl}, ω^b_{whrl}, and, ω^b_{whrr}, are the respective error-free wheel speed measurements.

The calculation of the angular velocity of the rear axle is as shown in Equation (5) and (6).

$$\widehat{\omega}^b_{whr} = \frac{\omega^b_{whrr} + \omega^b_{whrl}}{2} + \frac{\varepsilon^b_{whrr} + \varepsilon^b_{whrl}}{2} \tag{5}$$

Expressing $\frac{\varepsilon^b_{whrr}+\varepsilon^b_{whrl}}{2}$ as ε^b_{whr} and $\frac{\omega^b_{whrr}+\omega^b_{whrl}}{2}$ as ω^b_{whr}

$$\widehat{\omega}^b_{whr} = \omega^b_{whr} + \varepsilon^b_{whr} \tag{6}$$

The vehicle's linear velocity in the body frame can be found from $v = \omega r$, where r is a constant which maps the angular velocity of the rear axle to its linear velocity:

$$v^b_{wh} = \omega^b_{whr} r + \varepsilon^b_{whr} r \tag{7}$$

Take $\varepsilon^b_{whr} r \ as \ \varepsilon^b_{whr,v}$

$$v^b_{whr} = \omega^b_{whr} r + \varepsilon^b_{whr,v} \tag{8}$$

The vehicle's displacement in the body frame can be found through the integration of its velocity from $Equation$ 8 and incrementally updated for continuous tracking. $\varepsilon^b_{whr,x}$ in $Equation$ 9 is the integral of $\varepsilon^b_{whr,v}$ from $Equation$ 8.

$$x^b_{whr} = \int_{t-1}^{t} (\omega^b_{whr} r) + \varepsilon^b_{whr,x} \tag{9}$$

The uncertainty in the position estimation can be found through $Equation$ (10) during the presence of the GNSS signal. The task thus becomes that of estimating $\varepsilon^b_{whr,x}$ during GNSS outages needed to correct the vehicles displacement x^b_{whr}.

$$\varepsilon^b_{whr,x} \approx x^b_{whr} - x^b_{GNSS} \tag{10}$$

where x^b_{GNSS} refers to the true displacement of the vehicle measured according to reference [18] using Vincenty's formula for geodesics on an ellipsoid based on the latitudinal and longitudinal positional information of the vehicle as implemented in [65], [66]. The accuracy of x^b_{GNSS} is however limited to the accuracy of the GNSS which according to [67], is defined as ±3m.

4.3 WhONet's Learning Scheme

We adopt the WhONet model developed and evaluated in [19] based on the simple Recurrent Neural Network proposed by [68]. The WhONet's learning scheme is as presented in Figure 4, where for any time t, the Neural Network's (NN's) input, $X_{t|t-0.9}$, is made up of the wheel speed information of all four wheels of the vehicle: $\widehat{\omega}^b_{whrl}$, $\widehat{\omega}^b_{whrr}$, $\widehat{\omega}^b_{whfl}$ and $\widehat{\omega}^b_{whfr}$ from every tenth of a second within the previous second; $X_t, X_{t-0.1} \dots\dots \ and \ X_{t-0.9}$.

The NN is then tasked with predicting Y_t, which is defined as the error $\varepsilon^b_{whr,x}$ between the GNSS-derived displacement x^b_{GNSS} and the wheel-speed-derived displacement x^b_{whr}.

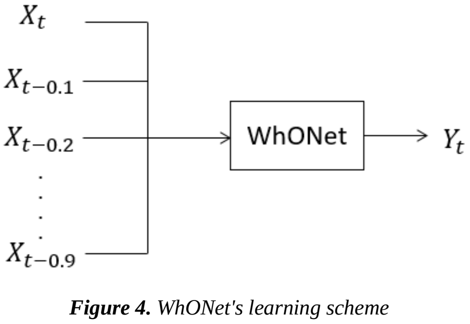

Figure 4. *WhONet's learning scheme*

4.4 Performance Evaluation Metric

The Cumulative Root Square Error (CRSE) metric as adopted in [17] is used to evaluate the performance of the WhONet model in this work. The CRSE describes the cumulative root mean squared of the prediction error for every one second of the total duration of the GNSS outage. The mathematical definition of the CRSE is as presented in Equation (12).

$$\text{CRSE} = \sum_{t=1}^{N_t} \sqrt{{e_{pred}}^2} \tag{11}$$

Where N_t is the length of the GNSS defined as 18s, e_{pred} refers to the prediction error, and t is the sampling period.

We also adopt the Average Error Per Second (AEPS) metric from [16] in evaluating the performance of the WhONet model. The AEPS measures the average error of the prediction every second of the GNSS outage and is defined mathematically in Equation (12) below.

$$\text{A}EPS = \frac{1}{N_t} \cdot \sum_{t=1}^{N_t} e_{pred}. \tag{12}$$

4.5 Training of the WhONet Models

The training of the WhONet model is done according to [19]. The model is trained using the Keras–Tensorflow version 1.15 platform [69], in order to ensure compatibility with the SHAP library [70]. Unlike the training dataset is made up of the first 80% of the V-Vw12 data subset of the IO-VNB dataset as presented in Table 1. The V-Vw12 training set used to train the WhONet Model is characterised by motion on an approximate straight-line trajectory on the motorway over a distance of 265m. The model was optimised with the adamax optimiser using an initial learning rate of 0.0007 and a mean absolute error loss function. Table 2 highlights the parameters characterising the training of the WhONet model.

Table 2. *WhONet's training parameters.*

Parameters	Displacement Estimation
Learning rate	0.0007
Dropout rate	0.05
Time step	1
Hidden layers	1
Hidden neurons	72
Batch size	128

4.6 WhONet's Evaluation

The WhONet model is evaluated on the last 20% of the V-Vw12 IO-VNBD data subset characterised by approximately 18 seconds.

The performance of the WhONet is examined on a relatively easy scenario, i.e., an approximate straight-line travel on the motorway to measure the WhONet's performance on a relatively easy driving situation. Nonetheless, the motorway scenario could be considered challenging due to the large distance covered per second. GPS outages are assumed on the test scenarios, for the purpose of the investigation with a prediction frequency of 1s.

4.7 SHapley Additive exPlanations (SHAP) Method

SHAP which was proposed in 2017 [55], is a unified framework for the interpretation of the predictions of machine learning models. It is regarded as the only locally accurate and consistent method for feature attribution based on expectations. As well as being able to provide interpretable predictions, SHAP also interprets feature importance scores from complex models. SHAP values presents a unified measure of feature importance by assigning an importance value φ_i to each feature, as such describing the effects of having that feature included in the model's prediction. SHAP values in cooperative game theory could be represented mathematically as follows:

$$\varphi_i = \sum\nolimits_{S\subseteq F,\{i\}} \frac{|S|!\,(|F|-|S|-1)!}{|F|!} [f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \tag{11}$$

Here F refers to a set of all the features, S is a subset of all features from F after the i^{th} value has been removed. Consequently, two models f_S and $f_{S\cup\{i\}}$ are retrained and then a comparison is made between the predictions from these models and the current input $f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)$, where x_S describes the input feature's values in the set S. The estimation of φ_i from $2^{|F|}$ differences is done by the approximation of the Shapley value by either Shapley quantitative influence or performing Shapley sampling.

We employ the SHAP approach in this study to interpret the predictions of the WhONet model. The SHAP analysis on the WhONet model is presented and discussed in Section 5.

5 Results and Discussions

The results from the evaluation 0f the WhONet model on the test data subset are presented on Table 3. The results reported, shows that the adapted WhONet model achieves a CRSE of 0.72 m and AEPS of 0.04m/s after about 265m and 18 seconds of travel.

Table 3. *Performance measures of the WhONet model on the test dataset*

Performance Evaluation Metrics	Results
CRSE	0.72m

AEPS	0.04m/s

To further understand the predictions of the WhONet model trained on the motorway dataset, the SHAP values were calculated for the test observations. Figure 5 shows each feature ordered according to its average absolute SHAP value, with the feature that contributes the most to the model's output identified as the top placed bar in the illustration. Similar to Figure 5, Figure 6 presents a SHAP summary plot which also ranks the features according to their influence on the model's output. However, for each feature captured on the summary plot, there are multiple coloured dots which each represents the SHAP value for that feature, and for each observation in the test dataset. The colour of the dots indicate the magnitude of the value of the feature, with the red dots indicating high wheel speed values and blue dots representing low wheel speed values. A greater distance from zero shows a greater influence on the model's prediction whilst a smaller distance shows less impact. The SHAP summary plot, essentially, reaveals the effect an increase or a decrease on a specific feature affects the WhONet's position error estimation and to what degree. Figures 5 and 6 show that the top 5 features with the greatest impact on the WhONet's output are the wheel speed of the front right wheel at time $t - 0.1s$, the front left wheel speed at $t - 0.2s$, the front right wheel speed at $t - 0.3s$, the front right wheel speed at $t - 0.9s$, and the rear left wheel speed at $t - 0.1s$. Of these 5 features, an increase in the value of the wheel speed of the front right wheel at $t - 0.1s$ and $t - 0.9s$ leads to a decrease in the position error prediction. Conversely, an increase in the value of the wheel speed of the front left wheel speed at $t - 0.2s$, the front right wheel speed at $t - 0.3s$ and the rear left wheel speed at $t - 0.1s$ leads to an increase in the position error estimation. We also notice that the higher the speed of both front wheels at $t - 0.1s$ the higher the accuracy of the position error estimation. Similarly, these behaviors are repeated for the front right wheel speed at $t - 0.9s$, and $t - 0.8s$ and for the front left wheel speed at $t - 0.4s$, $t - 0.7s$, and $t - 08s$. These observations hint at a greater connection between the two front wheels and the accuracy of the predicted position error compared to the two rear wheels. We investigate these observations further by looking at the SHAP waterfall plot and the SHAP decision plot.

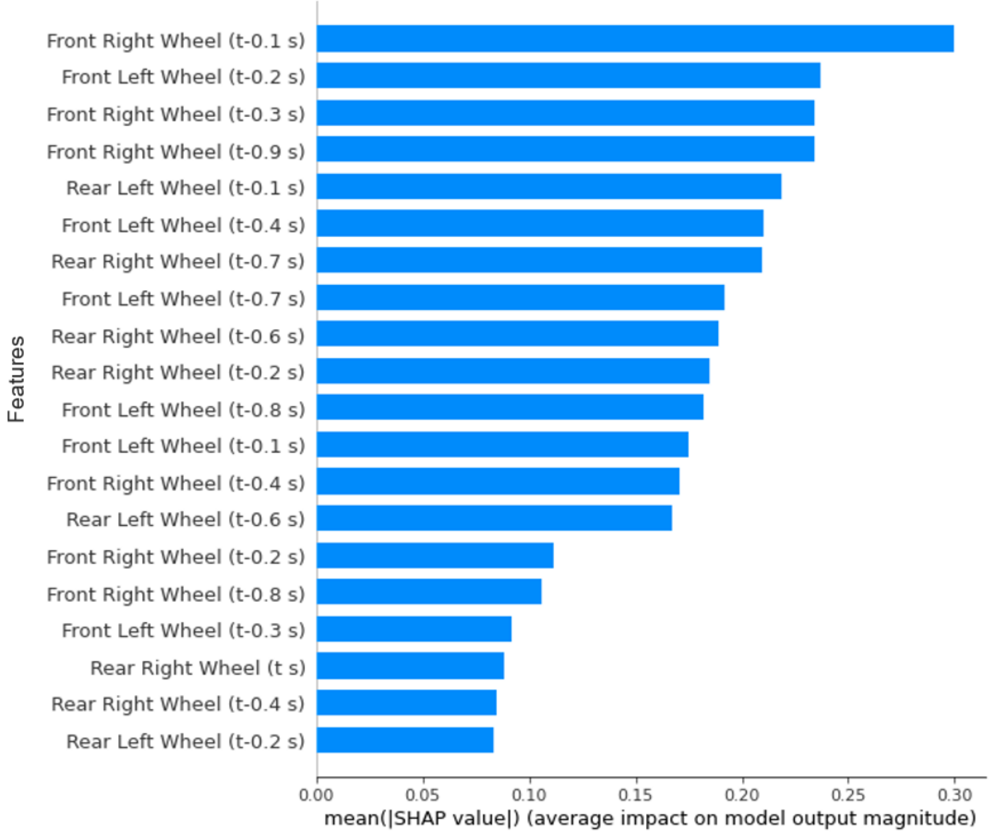

Figure 5. *Average absolute SHAP value per feature*

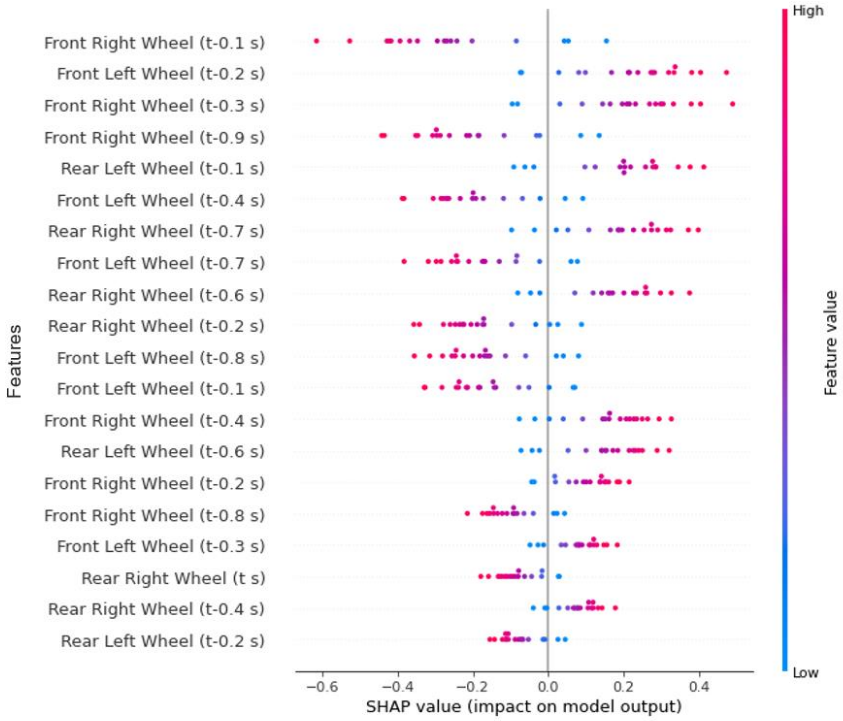

Figure 6. *SHAP summary plot showing individual feature contributions*

The SHAP waterfall plot, shown in Figure 7, offers more insight into how each feature affects the WhONet's position error estimation. Starting from the expected value, when no features are taken into account (see the bottom of Figure 7), the waterfall plot shows how applying one feature at a time affects the predicted position error, increasing or decreasing the error in the model's estimations, until it reaches the final model output, after all features have been taken into account. The features are sorted according to their influence on the predicted position error with those that have the least impact on the model's output shown at the bottom of the plot and those with the most impact positioned at the top. The colour of the arrows indicates the direction of the change with red arrows showing an increase in the model's predicted position error and blue arrows showing a decrease. In this specific test observation , the wheel speed of the front left wheel at $t - 0.2s$ increases the predicted position error the most. However, considering the SHAP values for each wheel speed across the timesteps within the sequence as presented on Table 4, it is observed that overall, the front left wheel contributes significantly to the reduction of the error in the predicted positional uncertainty compared to the other three wheels. It is further observed that the rear left wheel increases the error in the position uncertainty estimation the most. Table 5 reports the aforementioned behaviours across 4 additional test observations.

Table 4. *Total SHAP values for wheel each across the timesteps*

	Front Left	**Front Right**	**Rear Left**	**Rear Right**
t	-0.01	0.02	0.03	-0.11
t-1	-0.18	-0.28	0.22	0.01
t-2	0.27	0.14	-0.07	-0.17
t-3	0.09	0.23	0.06	0.09
t-4	-0.20	0.16	0.00	0.08
t-5	-0.04	0.08	-0.08	0.04
t-6	-0.03	-0.04	0.17	0.20
t-7	-0.21	0.01	-0.03	0.20
t-8	-0.18	-0.11	0.05	-0.05
t-9	-0.01	-0.22	0.06	-0.06
Grand total	**-0.50**	**0.00**	**0.41**	**0.23**

Table 5. *Average SHAP value for 5 test observations*

Test Observations	**Front Left**	**Front Right**	**Rear Left**	**Rear Right**
Observation 1	-0.50	0.00	0.41	0.23
Observation 2	0.13	0.18	-0.05	-0.04
Observation 3	-0.72	-0.01	0.61	0.35
Observation 4	-0.70	-0.09	0.47	0.35
Observation 5	0.22	0.10	-0.18	-0.10
Average	**-0.31**	**0.04**	**0.25**	**0.16**

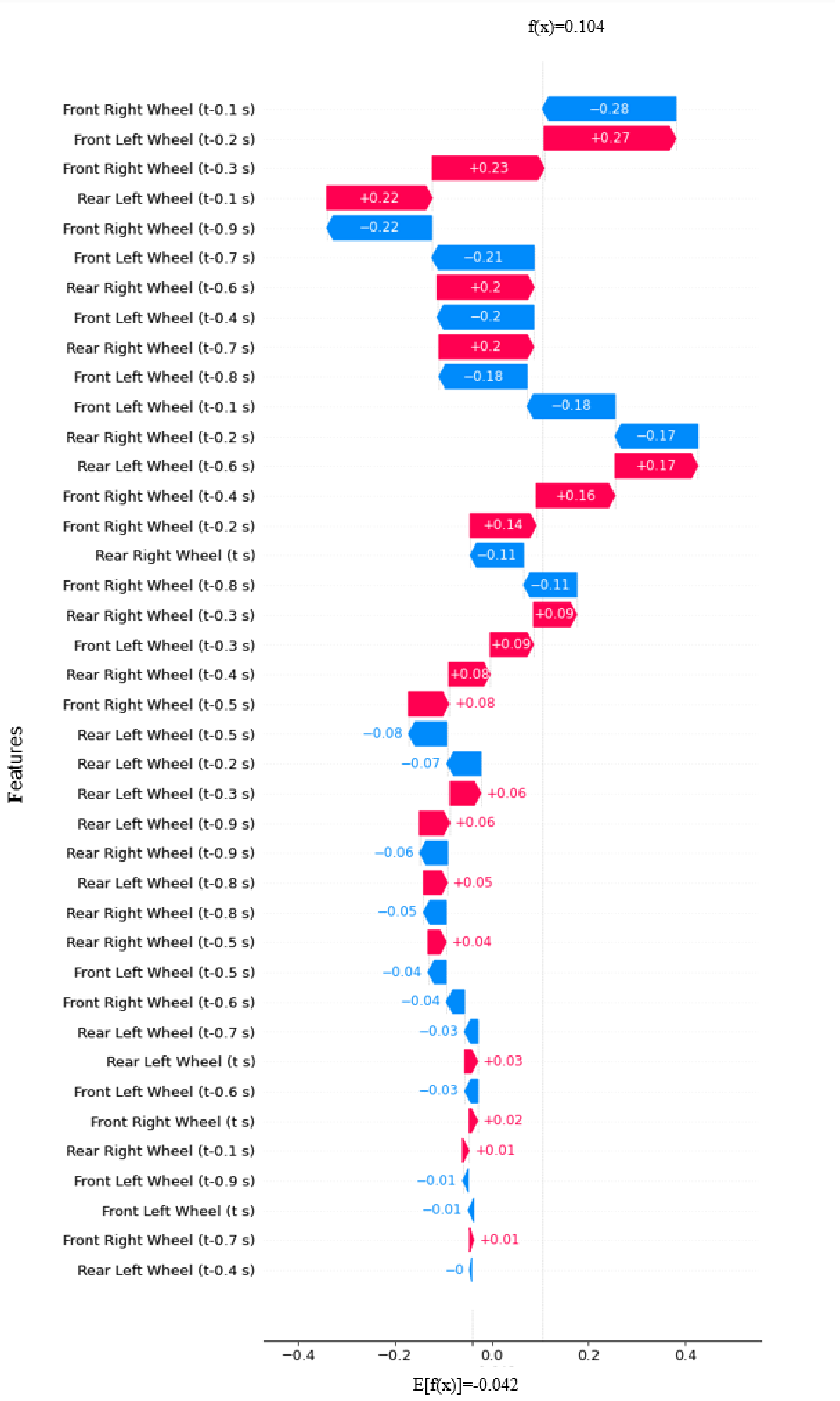

Figure 7. Sample SHAP *waterfall plot for a single observation*

Figure 8 provides a collective demonstration on how the model arrives at it's estimations across all the test observations. The decision plot uses a line graph to illustrate how the WhONet model navigates through the decision for each test observation, consequently demonstrating the effect of each feature on the decisions made by the model. Starting from the bottom, much like the waterfall plot, the decision plot shows how the SHAP values of the 20 most important features accumulate to move the model's predicted position error from the expected value to the final prediction . The y-axis, again, shows each feature ordered from lowest at the bottom of the plot to highest at the top, according to their influence of the model's predicted position error. The movement of the line on the x-axis is the result of the SHAP value for that feature. The results in the decision plot confirms what was show in the waterfall plot. In the majority of the cases, the rear wheels seem to increase the value of the predicted position error while the front wheels decreases it.

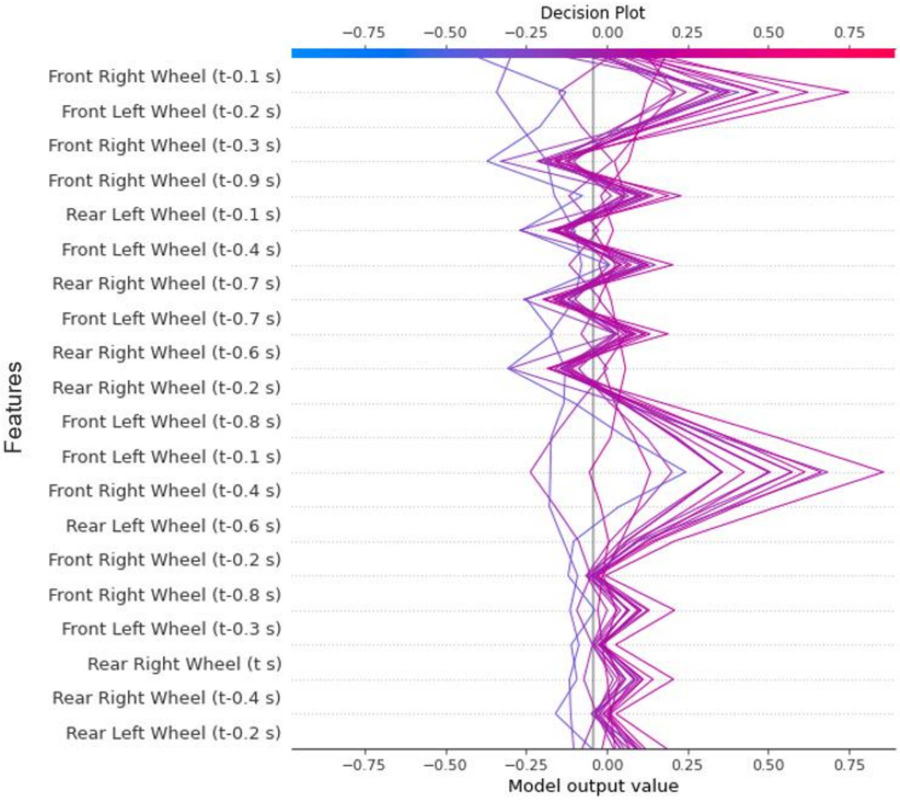

Figure 8. *Decision plot across all test observations*

6 Conclusions

In this work, we have examined the interpretability of the WhONet model on a relatively simple scenario: an approximate straight line trajectory on the motorway. Our study shows that overall, the two rear wheels are responsible for the most increase in the position error estimation, with the rear left being the most prevalant of the two. Although the reason for this isn't immediately clear, the contributions of the measurements from the front wheels compared to the wheels could be attributed to the vehicle being a front wheel drive. Nevertheless, these behaviours have been observed on a motion on an approximate straight line. Future research would involve exploring the generalisation of these behaviors to more complex scenarios, especially those chracterised by a differences in the wheel speeds of the left and right front wheels, such as on a round about, successive left right turns, etc. The output of this study could provide insights on how to improve the performance of the WhONet model for safer autonomous vehicle navigation.

Furthermore, gaining a deeper insight into how features influence the model's predicted position error offers transparency into the decision making of the model. This can be valuable for the different stakeholders. For insurance companies, for example, explainability can offer a deeper understanding of the underlying causes in case of an accident. This information can also help manufacturers improve autonomous vehicles so that they take into account features that increase the predicted position error during the manufacturing process, consequently reducing the chance of an accident happening in the first place. By understanding the model, car retailers can have better knowledge of the vehicles they have available and can highlight their strengths and weaknesses to potential customers. Finally, for consumers knowing how features affect the predicted position error reduces the element of the unknown and provides some transparency into how the autonomous vehicle makes certain decisions.

7 References

[1] J. Wang, L. Zhang, Y. Huang, and J. Zhao, "Safety of Autonomous Vehicles," *Journal of Advanced Transportation*, vol. 2020, 2020, doi: 10.1155/2020/8867757.

[2] P. Liu, R. Yang, and Z. Xu, "How Safe Is Safe Enough for Self-Driving Vehicles?," *Risk Analysis*, vol. 39, no. 2, pp. 315–325, Feb. 2019, doi: 10.1111/risa.13116.

[3] A. Papadoulis, M. Quddus, and M. Imprialou, "Evaluating the safety impact of connected and autonomous vehicles on motorways," *Accident Analysis and Prevention*, vol. 124, pp. 12–22, Mar. 2019, doi: 10.1016/j.aap.2018.12.019.

[4] J. Lee, D. Lee, Y. Park, S. Lee, and T. Ha, "Autonomous vehicles can be shared, but a feeling of ownership is important: Examination of the influential factors for intention to use autonomous vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 411–422, Oct. 2019, doi: 10.1016/J.TRC.2019.08.020.

[5] S.-J. Babak, S. A. Hussain, B. Karakas, and S. Cetin, "Control of autonomous ground vehicles: a brief technical review - IOPscience," 2017. https://iopscience.iop.org/article/10.1088/1757-899X/224/1/012029 (accessed Mar. 22, 2020).

[6] K. Onda, T. Oishi, and Y. Kuroda, "Dynamic Environment Recognition for Autonomous Navigation with Wide FOV 3D-LiDAR," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 530–535, Jan. 2018, doi: 10.1016/j.ifacol.2018.11.579.

[7] S. Ahmed, M. N. Huda, S. Rajbhandari, C. Saha, M. Elshaw, and S. Kanarachos, "Pedestrian and Cyclist Detection and Intent Estimation for Autonomous Vehicles: A Survey," *Applied Sciences*, vol. 9, no. 11, p. 2335, Jun. 2019, doi: 10.3390/app9112335.

[8] W. Yao *et al.*, "GPS signal loss in the wide area monitoring system: Prevalence, impact, and solution," *Electric Power Systems Research*, vol. 147, no. C, pp. 254–262, Jun. 2017, doi: 10.1016/j.epsr.2017.03.004.

[9] Gerard O'Dwyer, "Finland, Norway press Russia on suspected GPS jamming during NATO drill," 2018. https://www.defensenews.com/global/europe/2018/11/16/finland-norway-press-russia-on-suspected-gps-jamming-during-nato-drill/ (accessed Jun. 04, 2019).

[10] B. Templeton, "Cameras or Lasers?," 2017. http://www.templetons.com/brad/robocars/cameras-lasers.html (accessed Jun. 04, 2019).

[11] M. G. Petovello, M. E. Cannon, and G. Lachapelle, "Benefits of using a tactical-grade IMU for high-accuracy positioning," *Navigation, Journal of the Institute of Navigation*, vol. 51, no. 1, pp. 1–12, 2004, doi: 10.1002/J.2161-4296.2004.TB00337.X.

[12] C. Chen, X. Lu, A. Markham, and N. Trigoni, "IONet: Learning to Cure the Curse of Drift in Inertial Odometry," pp. 6468–6476, 2018.

[13] K. W. Chiang, A. Noureldin, and N. El-Sheimy, "Constructive neural-networks-based MEMS/GPS integration scheme," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 582–594, Apr. 2008, doi: 10.1109/TAES.2008.4560208.

[14] A. Noureldin, A. El-Shafie, and M. Bayoumi, "GPS/INS integration utilizing dynamic neural networks for vehicular navigation," *Information Fusion*, vol. 12, no. 1, pp. 48–57, 2011, doi: 10.1016/j.inffus.2010.01.003.

[15] W. Fang *et al.*, "A LSTM Algorithm Estimating Pseudo Measurements for Aiding INS during GNSS Signal Outages," *Remote Sensing*, vol. 12, no. 2, p. 256, Jan. 2020, doi: 10.3390/rs12020256.

[16] H. fa Dai, H. wei Bian, R. ying Wang, and H. Ma, "An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network," *Defence Technology*, 2019, doi: 10.1016/j.dt.2019.08.011.

[17] U. Onyekpe, V. Palade, and S. Kanarachos, "Learning to Localise Automated Vehicles in Challenging Environments Using Inertial Navigation Systems (INS)," *Applied Sciences 2021, Vol. 11, Page 1270*, vol. 11, no. 3, p. 1270, Jan. 2021, doi: 10.3390/app11031270.

[18] U. Onyekpe, S. Kanarachos, V. Palade, and S.-R. G. Christopoulos, "Vehicular Localisation at High and Low Estimation Rates during GNSS Outages : A Deep Learning Approach," in *In: Wani M.A., Khoshgoftaar T.M., Palade V. (eds) Deep Learning Applications, Volume 2. Advances in Intelligent Systems and Computing, vol 1232.*, V. P. M. Arif Wani, Taghi Khoshgoftaar, Ed. Springer Singapore, 2020, pp. 229–248. doi: 10.1007/978-981-15-6759-9_10.

[19] U. Onyekpe, V. Palade, A. Herath, S. Kanarachos, and M. E. Fitzpatrick, "WhONet: Wheel Odometry neural Network for vehicular localisation in GNSS-deprived environments," *Engineering Applications of Artificial Intelligence*, vol. 105, p. 104421, 2021, doi: 10.1016/J.ENGAPPAI.2021.104421.

[20] U. Onyekpe, V. Palade, S. Kanarachos, and S.-R. G. Christopoulos, "A Quaternion Gated Recurrent Unit Neural Network for Sensor Fusion," *Information*, vol. 12, no. 3, p. 117, Mar. 2021, doi: 10.3390/info12030117.

[21] K.-W. Chiang, "The Utilization of Single Point Positioning and Multi-Layers Feed-Forward Network for INS/GPS Integration." pp. 258–266, Sep. 12, 2003.

[22] R. Sharaf, A. Noureldin, A. Osman, and N. El-Sheimy, "Online INS/GPS integration with a radial basis function neural network," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 3, pp. 8–14, Mar. 2005, doi: 10.1109/MAES.2005.1412121.

[23] N. El-Sheimy, K. W. Chiang, and A. Noureldin, "The utilization of artificial neural networks for multisensor system integration in navigation and positioning instruments," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1606–1615, Oct. 2006, doi: 10.1109/TIM.2006.881033.

[24] M. Malleswaran, V. Vaidehi, and S. A. Deborah, "CNN based GPS/INS data integration using new dynamic learning algorithm," *International Conference on Recent Trends in Information Technology, ICRTIT 2011*, no. June, pp. 211–216, 2011, doi: 10.1109/ICRTIT.2011.5972270.

[25] M. Malleswaran, V. Vaidehi, A. Saravanaselvan, and M. Mohankumar, "Performance analysis of various artificial intelligent neural networks for GPS/INS Integration," *Applied Artificial Intelligence*, vol. 27, no. 5, pp. 367–407, 2013, doi: 10.1080/08839514.2013.785793.

[26] L. Semeniuk and A. Noureldin, "Bridging GPS outages using neural network estimates of INS position and velocity errors," in *Measurement Science and Technology*, Oct. 2006, vol. 17, no. 10, pp. 2783–2798. doi: 10.1088/0957-0233/17/10/033.

[27] P. Merriaux, Y. Dupuis, P. Vasseur, and X. Savatier, "Wheel Odometry-based Car Localization and Tracking on Vectorial Map (Extended Abstract)," 2014.

[28] U. Onyekpe, S. Kanarachos, V. Palade, and S.-R. G. Christopoulos, "Learning Uncertainties in Wheel Odometry for Vehicular Localisation in GNSS Deprived Environments," in *International Conference on Machine Learning Applications (ICMLA)*, Dec. 2020, pp. 741–746. doi: 10.1109/ICMLA51294.2020.00121.

[29] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2016, vol. 2016-December, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Adv Neural Inf Process Syst*, vol. 28, 2015.

[32] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," in *2007 IEEE 23rd international conference on data engineering workshop*, 2007, pp. 801–810.

[33] V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers in big Data*, p. 39, 2021.

[34] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, "Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions," *arXiv preprint arXiv:2112.11561*, 2021.

[35] M. Fox, D. Long, and D. Magazzeni, "Explainable planning," *arXiv preprint arXiv:1709.10256*, 2017.

[36] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)," *IEEE access*, vol. 6, pp. 52138–52160, 2018.

[37] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[38] H. K. Dam, T. Tran, and A. Ghose, "Explainable software analytics," in *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results*, 2018, pp. 53–56.

[39] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," *Information Fusion*, vol. 76, pp. 89–106, 2021.

[40] M. Ribera and A. Lapedriza, "Can we do better explanations? A proposal of user-centered explainable AI.," in *IUI Workshops*, 2019, vol. 2327, p. 38.

[41] M. M. A. de Graaf and B. F. Malle, "How people explain action (and autonomous intelligent systems should too)," 2017.

[42] J. A. Glomsrud, A. Ødegårdstuen, A. L. S. Clair, and Ø. Smogeli, "Trustworthy versus explainable AI in autonomous vessels," in *Proceedings of the International Seminar on Safety and Security of Autonomous Vessels (ISSAV) and European STAMP Workshop and Conference (ESWC)*, 2019, pp. 37–47.

[43] S. R. Haynes, M. A. Cohen, and F. E. Ritter, "Designs for explaining intelligent agents," *International Journal of Human-Computer Studies*, vol. 67, no. 1, pp. 90–110, 2009.

[44] R. Sheh and I. Monteath, "Introspectively assessing failures through explainable artificial intelligence," in *IROS Workshop on Introspective Methods for Reliable Autonomy*, 2017, pp. 40–47.

[45] R. Barzilay, D. McCullough, O. Rambow, J. DeCristofaro, T. Korelsky, and B. Lavoie, "A new approach to expert system explanations," 1998.

[46] P. Langley, B. Meadows, M. Sridharan, and D. Choi, "Explainable agency for intelligent autonomous systems," 2017.

[47] S. Sohrabi, J. Baier, and S. McIlraith, "Preferred explanations: Theory and generation via planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011, vol. 25, no. 1, pp. 261–267.

[48] A. B. Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion*, vol. 58, pp. 82–115, 2020.

[49] M. T. Ribeiro, S. Singh, and C. Guestrin, "' Why should i trust you?' Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[50] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32, no. 1.

[51] S. Krishnan and E. Wu, "Palm: Machine learning explanations for iterative debugging," in *Proceedings of the 2Nd workshop on human-in-the-loop data analytics*, 2017, pp. 1–6.

[52] O. Bastani, C. Kim, and H. Bastani, "Interpretability via model extraction," *arXiv preprint arXiv:1706.09773*, 2017.

[53] S. Tan, R. Caruana, G. Hooker, and Y. Lou, “Distill-and-compare: Auditing black-box models using transparent model distillation,” in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 303–310.

[54] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the GDPR,” *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[55] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” 2017. Accessed: May 02, 2022. [Online]. Available: https://github.com/slundberg/shap

[56] A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou, “A peek into the black box: exploring classifiers by randomization,” *Data Min Knowl Discov*, vol. 28, no. 5, pp. 1503–1529, 2014.

[57] J. Adebayo and L. Kagal, “Iterative orthogonal feature projection for diagnosing bias in black-box models,” *arXiv preprint arXiv:1611.04967*, 2016.

[58] A. Datta, S. Sen, and Y. Zick, “Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems,” in *2016 IEEE symposium on security and privacy (SP)*, 2016, pp. 598–617.

[59] P. Cortez and M. J. Embrechts, “Opening black box data mining models using sensitivity analysis,” in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2011, pp. 341–348.

[60] P. Cortez and M. J. Embrechts, “Using sensitivity analysis and visualization techniques to open black box data mining models,” *Information Sciences*, vol. 225, pp. 1–17, 2013.

[61] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, “Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation,” *journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015.

[62] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[63] U. Onyekpe, V. Palade, S. Kanarachos, and A. Szkolnik, “IO-VNBD: Inertial and Odometry benchmark dataset for ground vehicle positioning,” *Data in Brief*, vol. 35, 2021, doi: 10.1016/j.dib.2021.106885.

[64] U. Onyekpe, V. Palade, S. Kanarachos, and A. Szkolnik, “IO-VNBD: Inertial and odometry benchmark dataset for ground vehicle positioning,” *Data in Brief*, vol. 35, p. 106885, May 2021, doi: 10.1016/j.dib.2021.106885.

[65] T. Vincenty, “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations,” *Survey Review*, vol. 23, no. 176, pp. 88–93, 1975, doi: 10.1179/sre.1975.23.176.88.

[66] M. Pietrzak, “vincenty · PyPI,” 2016. https://pypi.org/project/vincenty/ (accessed Apr. 12, 2019).

[67] “VBOX Video HD2,” 2019. https://www.vboxmotorsport.co.uk/index.php/en/products/video-loggers/vbox-video (accessed Feb. 26, 2020).

[68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Internal Representations By Error Propagation,” 1985.

[69] Google Brain, “tensorflow 1.15.” 2017.

[70] S. Lundberg, “shap 0.40.0.” Oct. 2021.