



**COMILLAS**

UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

## SHADING DETECTION IN GNSS SYSTEMS FOR JOHN DEERE

Autor: Antonio Pardo de Santayana Navarro

Director: Dr. Leonard Franklin Register

Madrid



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Shading Detection in GNSS Systems for John Deere

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2022/2023 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Antonio Pardo de Santayana Navarro      Fecha: 07/ 06/ 2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Dr. Leonard Franklin Register      Fecha: ...../ ...../ .....





**COMILLAS**

UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

## SHADING DETECTION IN GNSS SYSTEMS FOR JOHN DEERE

Autor: Antonio Pardo de Santayana Navarro

Director: Dr. Leonard Franklin Register

Madrid

# Acknowledgements

I would like to take a moment to give credit and highlight certain institutions and individuals who have played a significant role in the completion of my bachelor's thesis and in my studies at ICAI.

Thanks, firstly, to all the professors I have had, both in ICAI and in the University of Texas at Austin, for helping me create the knowledge base upon which this bachelor's thesis is constructed, and fostering my personal curiosity.

I would also like to express gratitude to the University of Texas at Austin and John Deere, for the opportunity to conduct such a relevant thesis, apart from offering the necessary tools and general support in order to conduct it.

Furthermore, I would like to thank my fellow group members during our work on this project in Austin: Anmol Anand, Chiadika Obinwa, Drew Conyers and Sathya Balakumar. Although this thesis focuses on my personal contributions it would have not been possible without their involvement.

And finally, I must also thank my family and friends for supporting me throughout these four years, and all those before. Their unwavering love and inspiration have been crucial for the realization of this project.

Sincerely,

Antonio Pardo de Santayana Navarro

# SHADING DETECTION IN GNSS SYSTEMS FOR JOHN DEERE

**Author: Pardo de Santayana Navarro, Antonio.**

Supervisor: Register, Dr. Leonard Franklin.

Collaborating Entity: John Deere

## ABSTRACT

This project aims to address communication disruption between John Deere tractors and satellites caused by shading when a tractor travels beneath an object. The solution involved developing, training, and testing different models using signal processing techniques from data collected on a GNSS board to predict shading conditions of raw GNSS data. Based on testing and experimentation, the report does not recommend using solely GNSS data.

**Keywords:** GNSS, GPS, Shading, Machine Learning, John Deere

## 1. Introduction

This report aims to provide a thorough summary and assessment for a shading detection device, developed in the University of Texas at Austin in collaboration with John Deere. This project intended to identify the satellite and tractor communication disruption that occurs when a tractor travels beneath an object.

The Global Navigation Satellite System (GNSS), a collective designation for a network of satellites that uses trilateration to determine position (Inside GNSS, 2013), is how the tractor communicates with the satellite. In order to determine when a tractor travels underneath an object and connection between the two is broken Machine Learning techniques are introduced.

The presented solution was to develop, train, and test different models integrating signal processing techniques from data collected on a GNSS board to predict shading conditions of raw GNSS data.

The totality of this report is the recommendation for continued development where a conclusion will be provided on the effectiveness of using GNSS data to predict shading conditions.

## 2. System description

The project can be clearly divided in two main lines of work, the first relating to the hardware aspect of the project, and everything that went into correctly interfacing with it in order to collect a robust dataset. It encompasses tasks such as mounting the board, interfacing with the Arduino MEGA, and collecting data.

The second, in turn, includes the data analytics part of the project, and covers tasks such as performing feature selection to choose the best variables, training models using different algorithms and choosing the best performing one, fine-tuning the final model, and evaluating said model according to set criteria.

Firstly, the physical design of the system is shown in Figure 1. It consists of the SparkFun GNSS Multi-Band Magnetic Antenna, connected with its proprietary cable to the GPS-RTK-SMA ZED-F9P board. These components were bought together and are responsible for the gross of the data collection process. The ZED-F9P is a very advanced

board capable of detecting location with accuracy down to a few centimeters, and was particularly recommended by John Deere, given its similarities to the models present in John Deere equipment.

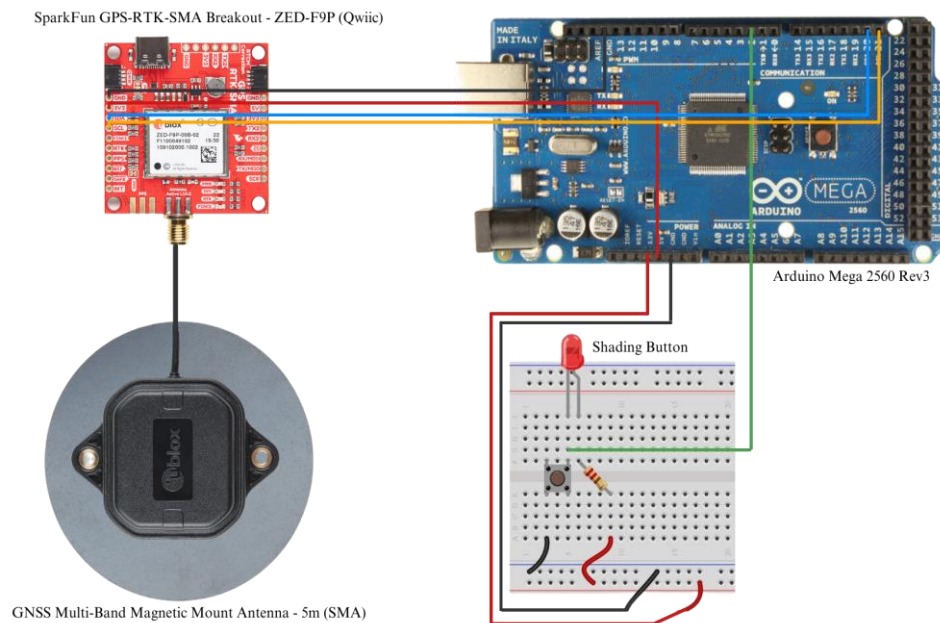


Figure 1 - Diagram of Physical System Design

The design also includes an Arduino MEGA 2560 Rev3 (Arduino, 2023) connected to the board, which is able to extract and collect all of the information it produced. The Arduino board is then connected to both a computer, to program it and receive the information, and to a breadboard with a button on it, which is used to manually input the target variable.

The data collection process involved deploying the Arduino code to the GPS board to gather readings from relevant variables. Due to the time-consuming nature of processing readings, the data collection rate was approximately one reading every 20 seconds, resulting in around 180 new values per hour-long session. Instead of writing the readings to a .csv file, the approach of copying and pasting from the console to a .txt file was adopted to manage the data output.

A total of 149 variables were initially collected, but due to the increased collection time, a refined set of 10 to 25 variables was chosen to balance model complexity and accuracy. The preliminary data collection involved a schedule focused on a single shading source, trees, to obtain 76 records over two 80-minute cycles. The purpose was to have enough data for feature engineering.

The data collection schedule was then expanded to include different types of shading in collaboration with a John Deere representative. This resulted in a varied dataset of 1,678 records, split into 1,342 training samples and 336 validation samples. Additionally, three additional datasets were collected for additional tests to be conducted during model evaluation, including unshaded, tree-shaded, and cardboard box-shaded scenarios.

Once the data collection process was relatively advanced, a Cat Boost model (ArcGis Pro 3.1, s.f.) was trained on the training samples after performing feature engineering and selecting the final variables. The Scikit-Learn library was used for these processes.



The model was tested on the testing dataset, resulting in an accuracy score of 95.65%, indicating that the model is on the right track.

In the methodology, the dataset was divided into training and validation sets using an 80-20 split. The numerical variables were normalized using the Scikit-Learn standard scaler, which converts them to values between 0 and 1. Normalization is important for fair representation, improved performance, faster convergence, and accurate interpretation of features.

The training sample was then used to train the Cat Boost model implemented in Scikit-Learn (Scikit-Learn, n.d.). Hyperparameter tuning was deemed unnecessary as Cat Boost has well-optimized default hyperparameters. Its built-in features, such as gradient-based learning and ordered boosting, make it less sensitive to hyperparameter variations, reducing the need for manual tuning.

### 3. Results

While multiple models were developed, only the final Cat Boost model is subject to this evaluation. The most commonly used metric, accuracy, is introduced, but due to the specific nature of the problem, ranking models based solely on accuracy was insufficient. Therefore, additional metrics were considered.

Recall, which measures the sensitivity of the model, was introduced to prioritize minimizing false negatives over false positives. The area under the curve (AUC) was also incorporated as an evaluation metric. The AUC represents the model's ability to rank positive instances higher than negative instances, providing an overall performance summary.

To combine these metrics, the shading score metric was created. It is a simple average of accuracy, recall, and AUC, with equal weight given to each metric. The score ranges between 0 and 1. While AUC and accuracy scores are expected to be similar due to the balanced dataset, the inclusion of recall with a 33% weight emphasizes its importance in achieving accurate results.

To provide a comprehensive analysis, four different tests were conducted on the model. The first test, considered the most crucial, evaluated the model's performance on the validation sample. The remaining tests assessed the model's performance on unshaded datasets, shaded datasets with trees as the source, and shaded datasets with a cardboard box as the source. As the shading score formula could only be applied to the first test, the hit-rate (accuracy) was used for the remaining tests.

In the general test, the confusion matrix showed an accuracy of 71.13% and a recall of 76.1%. The AUC score, calculated using the ROC curve, was 71.38%. Combining these metrics, the shading score was determined to be 72.87%, falling short of the initial expectations of 85% or later revised expectations of 80%. However, further analysis in the following chapter will consider additional factors before making a final recommendation.

The test on unshaded data yielded an accuracy of 75.7%, indicating that the model correctly predicted non-shaded instances in over three out of every four measurements. In the test with trees as the source of shading, the accuracy dropped to 58.4%, suggesting a general failure as it only surpassed a random guess by 8.4%. The final test using a

cardboard box as the shading source showed a relatively higher accuracy, of 74.74% compared to the tree shade case, likely due to the more potent shading caused by the box.

The confusion matrix for these four tests can be seen in Figure 2.

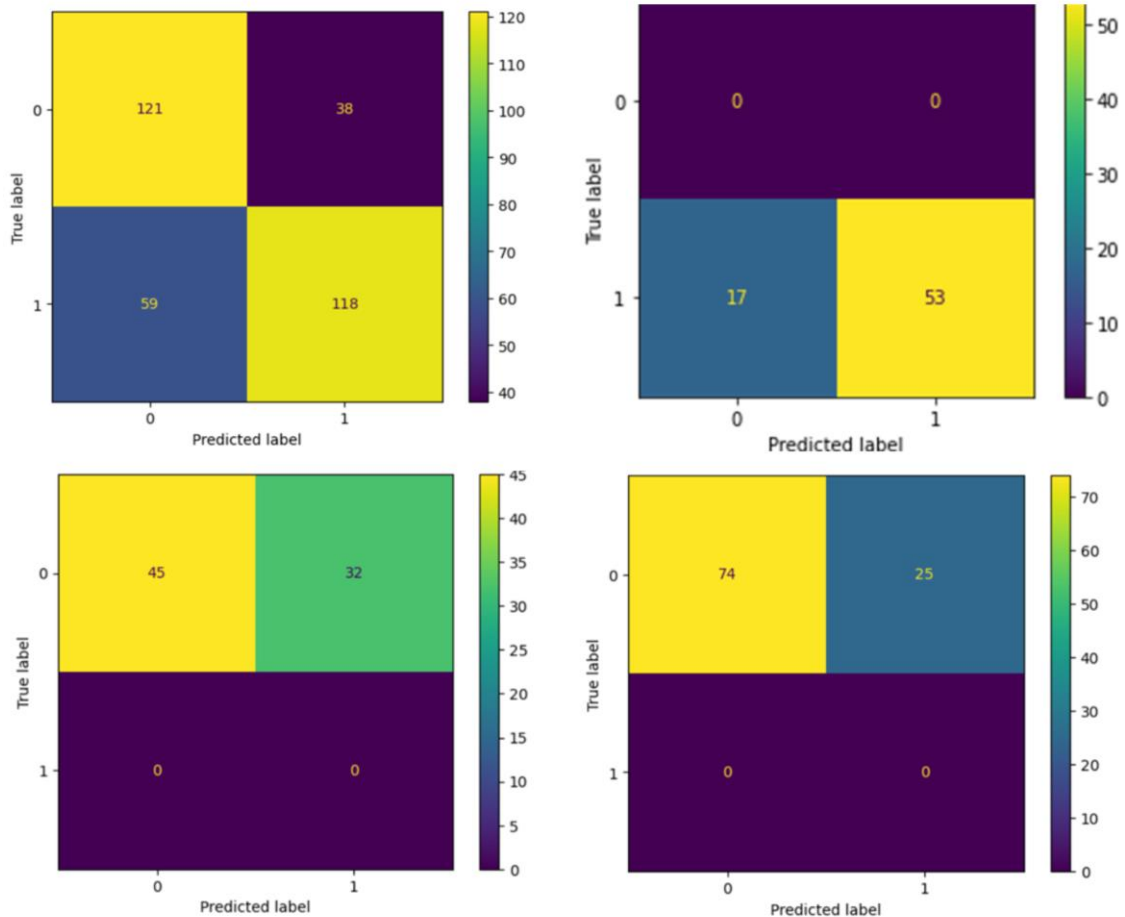


Figure 2 - Confusion Matrix for (Top Left) General Test, (Top Right) Unshaded Test, (Bottom Left) Tree Test and (Bottom Right) Cardboard Test

#### 4. Conclusions

The final results of the project, although not meeting expectations, offer a potential positive recommendation for John Deere. However, it is essential to address the limitations encountered. One major concern is the difficulty in generalizing the model to a wider dataset due to the possibility of location influences within the variables.

The project's focus on a GPS board, inherently designed for providing location data, adds complexity to isolating location influences. Moreover, all data collected originated from a single neighborhood, limiting the model's generalizability without diverse location data for testing.

Insufficient training data is another limitation, as the dataset size of 1,678 samples may not adequately capture the model's complexity and variables. Additionally, the lack of proof of representativity introduces uncertainty regarding the model's real-time shading detection capabilities, as it was trained and tested on specific environmental conditions.

Given these limitations, the final recommendation suggests not pursuing shading detection using only machine learning on GNSS data, unless John Deere is willing and able to follow an approach that circumvents the aforementioned limitations. The team proposes exploring alternative approaches to address the identified limitations, such as calibrating the model to mitigate location influences and collecting data from diverse locations.

These alternatives present promising avenues for future research, emphasizing the importance of considering constraints when developing machine learning models for shading detection.

## **5. References**

All references are included in Chapter 9.

# DETECCIÓN DE SOMBRA EN SISTEMAS GNSS CON JOHN DEERE

**Autor: Pardo de Santayana Navarro, Antonio.**

Director: Register, Dr. Leonard Franklin.

Entidad Colaboradora: John Deere

## RESUMEN DEL PROYECTO

Este proyecto pretende solucionar la interrupción de la comunicación entre los tractores John Deere y los satélites causada por las sombras cuando un tractor se desplaza por debajo de un objeto. La solución consistió en desarrollar, entrenar y probar diferentes modelos utilizando técnicas de procesamiento de señales a partir de datos recogidos en una placa GNSS para predecir las condiciones de sombreado de los datos GNSS sin procesar. Basándose en las pruebas y la experimentación, el informe no recomienda utilizar únicamente datos GNSS.

**Palabras clave:** GNSS, GPS, Sombra, Machine Learning, John Deere

### 1. Introducción

Este informe pretende ofrecer un resumen y una evaluación exhaustivos de un dispositivo de detección de sombras, desarrollado en la Universidad de Texas en Austin en colaboración con John Deere. Este proyecto pretende identificar la interrupción de la comunicación entre el satélite y el tractor que se produce cuando un tractor se desplaza por debajo de un objeto.

El Sistema Global de Navegación por Satélite (GNSS), un grupo que incluye a todos los miembros de una red de satélites que utiliza la trilateración para determinar la posición (Inside GNSS, 2013), es la forma en que el tractor se comunica con el satélite. Para determinar cuándo un tractor se desplaza por debajo de un objeto y se rompe la conexión entre ambos se introducen técnicas de Machine Learning.

La solución presentada consiste en desarrollar, entrenar y probar diferentes modelos que integran modelos de clasificación binaria a partir de datos recogidos en una placa GNSS para predecir las condiciones de sombreado de los datos GNSS sin procesar.

El entregable final de este informe es una recomendación elaborada a partir de las conclusiones sacadas del proyecto para John Deere, sobre si continuar o no con esta línea de investigación en un futuro, aclarando los principales obstáculos y criterios en los que centrarse.

### 2. Diseño del sistema

El proyecto puede dividirse claramente en dos líneas principales de trabajo, la primera relativa al aspecto hardware del proyecto, y todo lo que supuso interconectarse correctamente con él para recoger un conjunto de datos robusto. Abarca tareas como el montaje de la placa, la interconexión con el Arduino MEGA y la recogida de datos.

La segunda, por su parte, incluye la parte de análisis de datos del proyecto, y abarca tareas como realizar la selección de características para elegir las mejores variables,

entrenar modelos utilizando diferentes algoritmos y elegir el que mejor funcione, afinar el modelo final y evaluar dicho modelo según los criterios establecidos.

En primer lugar, la Figura 1 muestra el diseño físico del sistema. Consiste en la Antena Magnética Multibanda GNSS de SparkFun, conectada con su cable propietario a la placa GPS-RTK-SMA ZED-F9P. Estos componentes se compraron juntos y son responsables del grueso del proceso de recogida de datos. La ZED-F9P es una placa muy avanzada capaz de detectar la ubicación con una precisión de hasta unos pocos centímetros, y fue especialmente recomendada por John Deere, dadas sus similitudes con los modelos presentes en los equipos John Deere.

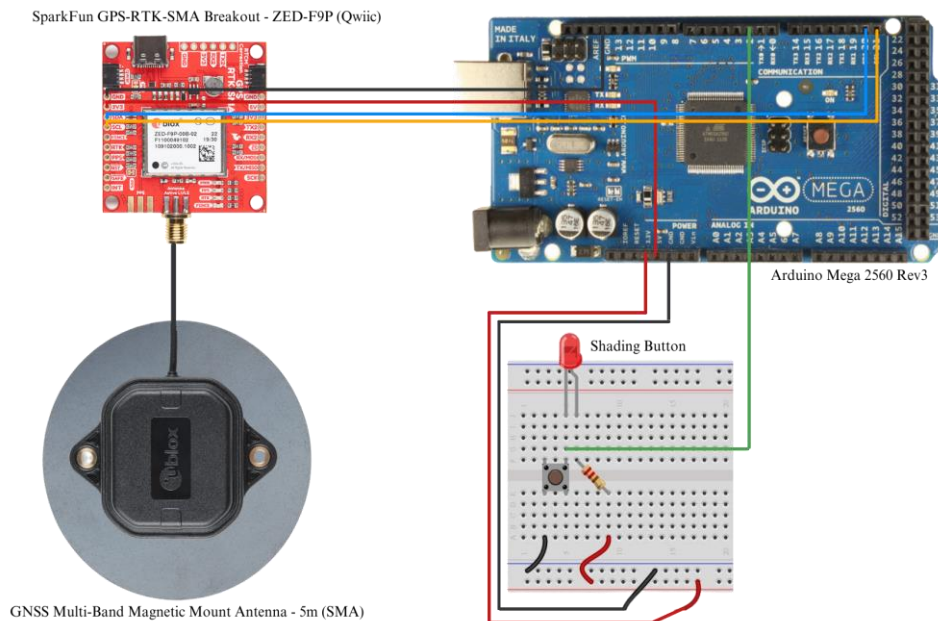


Figura 1 - Diseño Físico del Sistema

El diseño también incluye un Arduino MEGA 2560 Rev3 (Arduino, 2023) conectado a la placa, que es capaz de extraer y recoger toda la información que produce. La placa Arduino se conecta tanto a un ordenador, para programarla y recibir la información, como a una protoboard con un botón, que se utiliza para introducir manualmente la variable objetivo.

El proceso de recolección de datos trata de desplegar el código Arduino en la placa GPS para guardar registros de las variables pertinentes. Debido a que el procesamiento de las lecturas requiere mucho tiempo, el ritmo de recogida de datos fue de aproximadamente una lectura cada 20 segundos, lo que resulta en unos 180 valores nuevos por sesión de una hora de duración. En lugar de escribir los registros en un archivo .csv, se adoptó el método de copiar y pegar desde la consola a un archivo .txt para gestionar la salida de datos.

Inicialmente se recogieron un total de 149 variables, pero debido al aumento del tiempo de recogida, se eligió un conjunto refinado de 10 a 25 variables para equilibrar la complejidad y la precisión del modelo. La recogida preliminar de datos incluyó un programa centrado en una única fuente de sombra, los árboles, para obtener 76 registros a lo largo de dos ciclos de 80 minutos. El objetivo era disponer de datos suficientes para llevar a cabo un proceso de feature engineering.

A continuación, se amplió el programa de recogida de datos para incluir diferentes tipos de sombreado decididos en colaboración con un representante de John Deere. El resultado fue un conjunto de datos de 1.678 registros, divididos en 1.342 muestras de entrenamiento y 336 muestras de validación. Además, se recopilamos tres conjuntos de datos para realizar pruebas adicionales durante la evaluación del modelo, incluyendo escenarios sin sombra, con sombra de árboles y con sombra de cajas de cartón.

Una vez que el proceso de recogida de datos estaba relativamente avanzado, se entrenó un modelo Cat Boost (ArcGis Pro 3.1, s.f.) con las muestras de entrenamiento tras realizar ingeniería de características y seleccionar las variables finales. Para estos procesos se utilizó la biblioteca Scikit-Learn. El modelo se probó en el conjunto de datos de validación, obteniéndose una puntuación de precisión del 95,65%, lo que indica que el modelo iba por buen camino.

Para el entrenamiento, el conjunto de datos se dividió en conjuntos de entrenamiento y validación utilizando una división 80-20. Las variables numéricas se normalizaron utilizando el standard scaler de Scikit-Learn, que las convierte en variables discretas que toman valores entre 0 y 1.

A continuación, la muestra de entrenamiento se utilizó para entrenar el modelo Cat Boost implementado en Scikit-Learn (Scikit-Learn, n.d.). El ajuste de hiperparámetros se consideró innecesario, ya que Cat Boost tiene hiperparámetros por defecto bien optimizados. Sus características integradas, como el aprendizaje basado en gradientes y el refuerzo ordenado, lo hacen menos sensible a las variaciones de hiperparámetros, eliminando la necesidad de ajuste manual.

### **3. Resultados**

Aunque se desarrollaron múltiples modelos, sólo el modelo final Cat Boost ha sido evaluado en esta última sección. Para evaluar los modelos, la métrica más popular es la precisión, que hace referencia a “accuracy” en inglés, no a “precision” (puede dar lugar a confusión). Sin embargo, surge la necesidad de implementar también algunas métricas adicionales para capturar mejor el funcionamiento del modelo.

Por ejemplo, la métrica “recall” (que carece de traducción directa) mide la sensibilidad del modelo, para dar prioridad a minimizar los falsos negativos frente a los falsos positivos, lo cual dada la naturaleza del problema en cuestión es lógico. El área bajo la curva ROC (AUC) también se ha incorporado como medida de precisión. El AUC representa la capacidad del modelo para clasificar las instancias positivas por encima de las negativas, proporcionando un resumen general del rendimiento.

Para combinar estas métricas, se crea un nuevo valor llamado “shading score”, que recoge las tres métricas mencionadas anteriormente. Se trata de la media aritmética entre la precisión, el “recall” y la AUC, con el mismo peso asignado a cada uno de ellos. La puntuación toma valores entre 0 y 1, como sus componentes. Aunque se espera que las puntuaciones de AUC y precisión sean similares debido al conjunto de datos equilibrado, la inclusión del “recall” con una ponderación del 33% cambia el foco del modelo ligeramente.

Para proporcionar un análisis exhaustivo, se realizaron cuatro pruebas diferentes con el modelo. La primera prueba, considerada la más importante, tomó en cuenta el rendimiento del modelo en la muestra de validación designada inicialmente. Las pruebas

restantes evaluaron el rendimiento del modelo en tres conjuntos de datos adicionales: uno de ellos incluyendo solo valores sin sombra, el segundo con valores sombreados (utilizando árboles como la fuente de sombra), y el tercero con registros también sombreados, pero utilizando una caja de cartón como fuente de sombra. Como la fórmula de la métrica “shading score” sólo podía aplicarse a la primera prueba, se utiliza la tasa de aciertos (precisión) para las pruebas restantes.

En la prueba general, la matriz de confusión mostró una precisión de 71,13% y un “recall” de 76,1%. La puntuación AUC, calculada mediante la curva ROC, es del 71,38%. Con todo esto se saca finalmente el “shading score”, que da 72,87%, por debajo de las expectativas iniciales de 85% o de las expectativas revisadas posteriormente de 80%. Sin embargo, en el capítulo siguiente se analizarán otros factores antes de formular una recomendación final.

La prueba con datos no sombreados resulta en una precisión del 75,7%, lo que indica que el modelo predijo correctamente instancias no sombreadas en más de tres de cada cuatro registros de prueba. En la prueba con árboles como fuente de sombra, la precisión descendió al 58,4%, lo que sugiere un fallo general, ya que sólo superó en un 8,4% a una predicción aleatoria.

La prueba final, en la que se utilizó una caja de cartón como fuente de sombra, mostró una precisión relativamente mayor, del 74,74%, en comparación con el caso de la sombra de los árboles, probablemente debido a la mayor intensidad de la sombra causada por la caja. Las matrices de confusión de cada una de las cuatro pruebas se observan en la Figura 2.

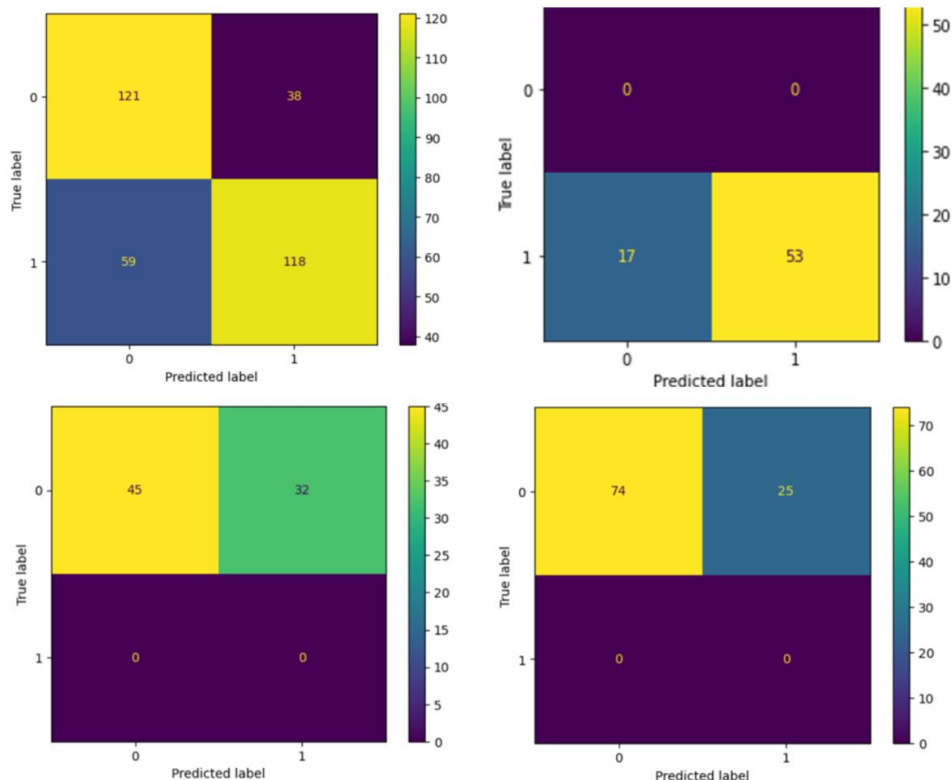


Figura 2 - Matrices de confusión para (Arriba Izquierda) la prueba general, (Arriba Derecha) la prueba sin sombra, (Abajo Derecha) la prueba con árboles y (Abajo Izquierda) la prueba con la caja de cartón

#### 4. Conclusiones

Los resultados finales del proyecto, aunque no llegan a satisfacer totalmente las expectativas, dan lugar a una conclusión para John Deere algo ambigua. Para llegar a una conclusión concreta, es esencial abordar las limitaciones encontradas. Una de las principales preocupaciones relacionadas con el proyecto es la dificultad de generalizar el modelo a un conjunto de datos recogido en otra parte del mundo.

Esto se debe a la posibilidad de influencias de localización dentro de las variables, o más en concreto la incapacidad de demostrar que los datos utilizados en el entrenamiento no dependen del lugar donde se toma la muestra.

El hecho de que el proyecto se centrara en una placa que trabaja en el marco de referencia del sistema GNSS, diseñado intrínsecamente para proporcionar datos de localización, añade complejidad al aislamiento de las influencias de localización. Además, todos los datos recogidos procedían de un único barrio, lo que limita la capacidad de generalizar del modelo sin datos de localización diversos para las pruebas.

La insuficiencia de datos de entrenamiento es otra limitación, ya que el tamaño del conjunto de datos de 1.678 muestras puede no captar adecuadamente la complejidad y las variables del modelo, que incluso después del proceso de feature engineering siguen siendo 24. Además, la falta de pruebas de representatividad introduce incertidumbre respecto a las capacidades de detección de sombras en tiempo real del modelo, ya que se entrenó y probó en condiciones ambientales específicas.

Dadas estas limitaciones, la recomendación final es principalmente negativa. El equipo propone explorar enfoques alternativos para abordar las limitaciones identificadas, como calibrar el modelo para mitigar las influencias de la ubicación y recopilar datos de diversas ubicaciones. Estas alternativas presentan vías prometedoras para futuras investigaciones y subrayan la importancia de tener en cuenta las limitaciones a la hora de desarrollar este tipo de modelos.



# Index

<b>Chapter 1. Introduction</b> .....	<b>7</b>
1.1 Motivation .....	7
1.2 General Introduction.....	7
<b>Chapter 2. Description of used technology</b> .....	<b>9</b>
2.1 GNSS Technology.....	9
2.1.1 Overview.....	9
2.1.2 GNSS Measurements Modelling.....	10
2.1.3 Real-time Kinematic Positioning.....	11
2.2 Relevant Standards .....	13
2.2.1 IS-200-GPS Standard.....	13
2.3 Data Analytics .....	13
2.3.1 u-Blox Framework.....	13
2.3.2 Arduino.....	14
2.3.3 Cat Boost.....	15
<b>Chapter 3. Prior Art</b> .....	<b>17</b>
3.1 Introduction .....	17
3.2 Prior Art Exclusive of Patent Information.....	17
3.2.1 Urban Positioning on a Smartphone: Real-time Shadow Matching using GNSS and 3D City Models.....	17
3.2.2 GNSS Position Error Estimated by Machine Learning Techniques with Environmental Information Input.....	19
3.2.3 Improving GNSS Positioning using Neural Network-based Corrections.....	20
3.2.4 Estimating Sunlight Using Signal Strength from Smartphone .....	21
3.3 Prior Art Based on Patent Information.....	22
3.3.1 Patent JP2015068768A.....	22
3.3.2 Patent US20100283674A1 .....	23
3.3.3 Patent US20210124059.....	25
3.3.4 Patent US7541975B2 .....	28
3.4 Conclusion.....	28
<b>Chapter 4. Project Definition</b> .....	<b>30</b>

4.1	Justification .....	30
4.1.1	<i>Potential Future Developments</i> .....	30
4.2	Objectives .....	31
4.3	Methodology .....	32
4.3.1	<i>Hardware and Interfacing</i> .....	32
4.3.2	<i>Data Analytics</i> .....	33
4.4	Planification and Economic Estimates .....	34
4.4.1	<i>Planification</i> .....	34
4.4.2	<i>Economic Estimates</i> .....	35
4.5	Safety and Ethical Considerations .....	36
4.5.1	<i>Health and Safety</i> .....	36
4.5.2	<i>Environmental Impact and Sustainability</i> .....	37
4.5.3	<i>Privacy and Data Security</i> .....	37
4.5.4	<i>Intellectual property</i> .....	38
<b>Chapter 5.</b>	<b><i>System Design</i></b> .....	<b>39</b>
5.1	Introduction .....	39
5.2	Components .....	40
5.2.1	<i>GNSS Multi-Band Magnetic Mount Antenna</i> .....	40
5.2.2	<i>SparkFun GPS-RTK-SMA Breakout - ZED-F9P</i> .....	40
5.2.3	<i>Arduino 2560 MEGA Rev 3</i> .....	41
5.3	Data Collection .....	42
5.3.1	<i>Initial Data Collection</i> .....	43
5.3.2	<i>Data Collection Schedule</i> .....	44
<b>Chapter 6.</b>	<b><i>Model design</i></b> .....	<b>46</b>
6.1	Introduction .....	46
6.2	Feature Engineering .....	46
6.2.1	<i>Preliminary Data Collection</i> .....	46
6.2.2	<i>Initial Cat Boost Model</i> .....	47
6.3	Final Model .....	48
6.3.1	<i>Variable Explanation</i> .....	48
6.3.2	<i>Model Feature Importance Values</i> .....	50
6.3.3	<i>Methodology</i> .....	52

6.3.4 Additional Model Information.....	53
6.4 Conclusion.....	56
<b>Chapter 7. Model Evaluation .....</b>	<b>57</b>
7.1 Introduction .....	57
7.2 Evaluation Techniques .....	57
7.2.1 Shading Score.....	58
7.3 Model Testing.....	59
7.3.1 General test .....	59
7.3.2 Test on Unshaded Data .....	61
7.3.3 Test on Tree Data.....	61
7.3.4 Test on Cardboard Box Data .....	63
7.4 Conclusion.....	64
7.4.1 Model Evaluation on Imbalanced Datasets.....	64
<b>Chapter 8. Conclusions &amp; Future Works .....</b>	<b>66</b>
8.1 General Conclusions.....	66
8.1.1 Recognized Limitations .....	66
8.2 Final recommendation.....	67
8.2.1 Possible Solutions to Limitations .....	67
8.3 Possible future work.....	68
8.3.1 Further Data Collection .....	69
8.3.2 Implementation.....	70
8.3.3 Further Statistical Analysis .....	71
<b>Chapter 9. References.....</b>	<b>72</b>
<b>ANNEX I: PROJECT ALIGNMENT WITH SDGs .....</b>	<b>75</b>
<b>ANNEX II Arduino Code for Data Collection .....</b>	<b>80</b>

## *Figure Index*

Figure 1 - Diagram of Physical System Design .....	8
Figure 2 - Confusion Matrix for (Top Left) General Test, (Top Right) Unshaded Test, (Bottom Left) Tree Test and (Bottom Right) Cardboard Test .....	10
Figure 3 - GNSS basic observables .....	10
Figure 4 - Layout of pseudo range measurements.....	11
Figure 5 - Typical RTK Cluster.....	12
Figure 6 - Gradient Boosting Algorithm Methodology.....	15
Figure 7 - Diagram of connections between smartphones and satellites.....	18
Figure 8 - Visual representation of algorithm .....	18
Figure 9 - First Stage in ML Process.....	19
Figure 10 - Second Stage in ML Process .....	20
Figure 11 - Visual Representation of Corrections .....	20
Figure 12 - Overview of Shading Detection Pipeline.....	22
Figure 13 - Diagram of car passing under object .....	23
Figure 14 - Structure of proposed Kalman filter .....	24
Figure 15 - Receiver configuration in the device .....	26
Figure 16 - (Left) Single distribution peak from direct signal. (Right) Multipeak indeterminable signal from multipath or NLOS signals.....	27
Figure 17 - Diagram of Sensor View.....	28
Figure 18 - Overview of hardware design .....	39
Figure 19 - GNSS Multi-band magnetic mount antenna.....	40
Figure 20 - SPARKFUN GPS-RTK-SMA BREAKOUT - ZED-F9P .....	41
Figure 21 - Arduino MEGA 2560 Rev 3 .....	42
Figure 22 - Example of data collection log .....	43
Figure 23 - Final Variables chosen.....	47
Figure 24 - Confusion Matrix for Feature Selection Model.....	48
Figure 25 - Evolution of Loss in Preliminary Model Training .....	54
Figure 26 - AUC Relationship with ROC curve.....	58

---

Figure 27 - Confusion Matrix for the Validation Dataset .....	60
Figure 28 - Confusion Matrix for Unshaded Test .....	61
Figure 29 - Confusion Matrix for Shaded by Trees Test.....	62
Figure 30 - Test under tree branches .....	62
Figure 31 - Confusion Matrix for Shaded by Cardboard Box Test.....	63
Figure 32 - Test Under Cardboard Box .....	64
Figure 33 - Mounted Data Collection System Example.....	70
Figure 34 - SDG 9 .....	75
Figure 35 - SDG 13 .....	76
Figure 36 - SDG 2 .....	77
Figure 37 - SDG 17 .....	78

---

## *Table Index*

Table 1 - Project Planification .....	35
Table 2 - Bill of Materials .....	36
Table 3 - Preliminary Data Collection Schedule .....	44
Table 4 - Data Collection Schedule .....	45
Table 5 - Data Collection Results .....	45
Table 6 - Attribute Importance Values .....	51
Table 7 - Possible Loss Functions .....	55

## **CHAPTER 1. INTRODUCTION**

### ***1.1 MOTIVATION***

When a tractor is moving through a field, at some point it is expected to encounter a line of trees, or in more rare cases, a bridge or a tunnel. When said obstacle partially blocks the tractor's connection to the GNSS constellations, the tractor's GPS will be unable to accurately pinpoint its location. In order to overcome this problem, what John Deere tractors currently do is to take certain checkpoints of its location, such that when a tractor enters or leaves a shaded area it can better calculate its trajectory.

The key issue tackled by this project is that, if those checkpoints could be taken at the precise moment when the tractor enters shading, it can more accurately and efficiently calculate its trajectory, taking the entry and exit points.

Given the sensitive nature of John Deere's data and the scale of the operation were an ML model to be deployed on John Deere's tractors, what this project intends to do is to collect a varied sample of GNSS data and run certain ML methodologies in order to provide a recommendation to John Deere as to whether pursue this line of research further, as it currently is in very early stages.

### ***1.2 GENERAL INTRODUCTION***

Ensuring accurate positioning and navigation is crucial for modern agricultural machinery, such as tractors, to optimize performance and productivity. However, obstacles like trees, bridges, and tunnels can obstruct the connection between the tractor's GPS system and the global navigation satellite system (GNSS) constellations, leading to inaccuracies in determining the tractor's location.

To address this challenge, John Deere, a renowned manufacturer of agricultural equipment, wants to conduct research on a technique where the tractor takes specific location checkpoints to improve trajectory calculations when entering or leaving shaded areas.

This project aims to explore the possibility of leveraging machine learning methodologies to predict shading. By collecting a diverse sample of GNSS data and applying ML techniques, the project seeks to assess the feasibility and potential benefits of deploying an ML model on John Deere tractors.

Given the sensitive nature of John Deere's proprietary data and the scale of their operations, this research initiative represents an early-stage exploration. The objective is to provide valuable insights and recommendations to John Deere regarding the viability of pursuing this line of research.

In this project, the objective is to contribute to the advancement of precision agriculture and provide valuable guidance to John Deere and their research teams as they work to bring innovation to the sector.



## **Chapter 2. DESCRIPTION OF USED TECHNOLOGY**

### ***2.1 GNSS TECHNOLOGY***

#### **2.1.1 OVERVIEW**

GNSS consists of a constellation of satellites orbiting at approximately 20,000 kilometers above the earth's surface, continually transmitting signals which enable users to calculate their three-dimensional position with global coverage (ESA, s.f.). In accordance with the wiki created by the reputable GNSS Science Support Centre (GSSC), the fundamentals, various receiver systems, and applications of GNSS will be reviewed in this section.

Coordinate systems are observable or model data that produce repeatable points of reference for GNSS, and function within the framework of reference systems and reference frames (Soffel & Langhans, 2012). Theoretical models that comply with physical criteria, set by the International Earth Rotation and Reference Systems Service (IERS), are referred to as reference systems.

There are three basic GNSS observables: pseudo range, carrier phase, and Doppler shift. Pseudo range or pseudo distance is the estimated or calculated distance between satellite and receiver that is not matching the actual geometric range due to synchronism errors between satellite and receiver clocks.

The time for data to travel, or traveling time  $\Delta T$ , is propagated from the satellite antenna to the phase center of the receiver. This measurement can be used to determine the maximum correlation between satellite and receiver by shifting a coded replica by  $\Delta T$  of the signal from the receiver until the maximum correlation is determined as seen in Figure 3.

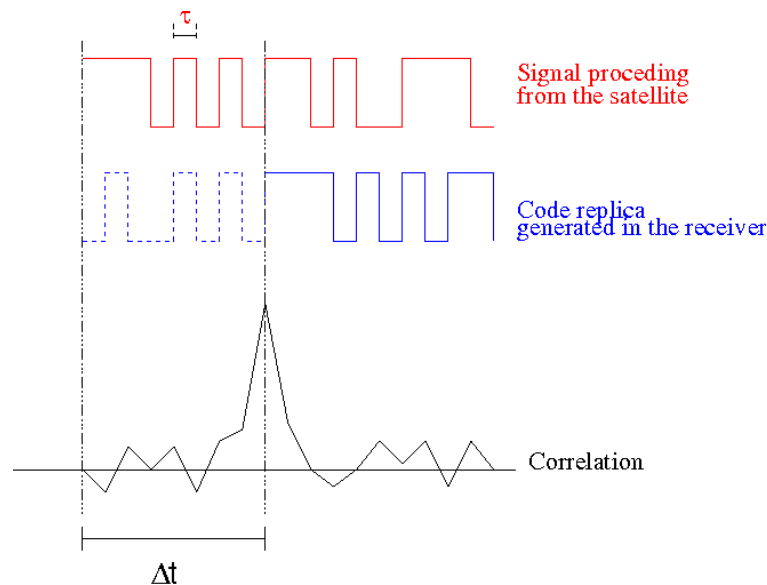


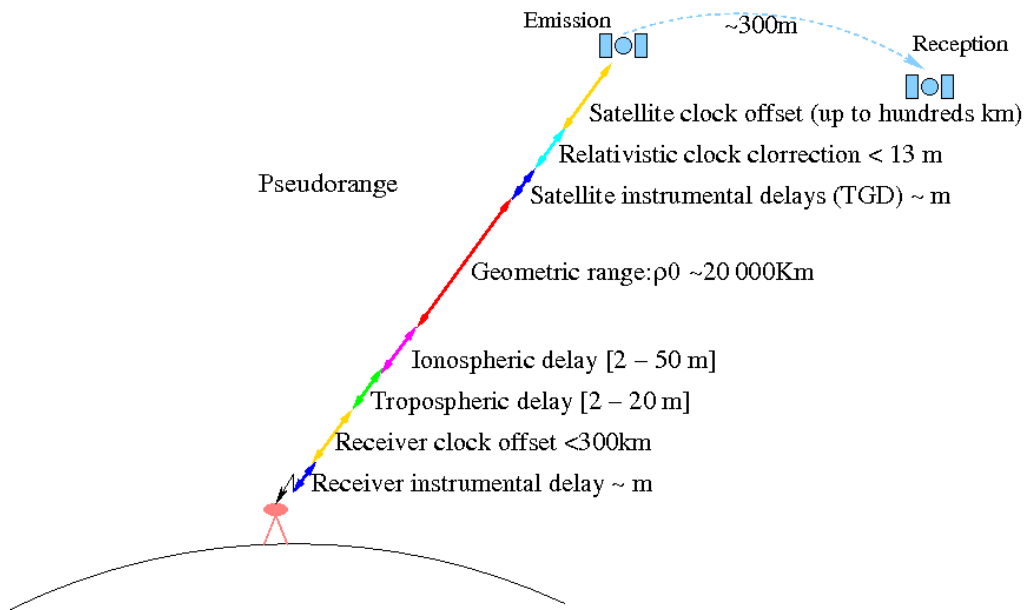
Figure 3 - GNSS basic observables

### 2.1.2 GNSS MEASUREMENTS MODELLING

When performing GNSS or any type of location-based positioning system, there is communication that takes place between a receiver and a satellite (ESA, s.f.). The point of the receiver is to measure the transmission time, also referred to as phase, of the signal that is emitted from the satellite.

Due to the distance between the two systems, there is a delay that takes place in the receiving of the signal. Two forms of GNSS have phase measurements which are code and carrier, code being a method in which the pseudo-random code and code from the satellite are compared, while carrier refers to a highly precise measurement of the range between the satellite and the receiver.

While most of the delays affect both forms of common GNSS/GPS techniques, some only affect the carrier forms like the phase wind-up effect. The wind-up effect only affects carrier methods because it is a change that takes place depending on the orientation of the satellite. A good way to imagine how this would affect the carrier method is to think of the carrier method as a piece of string that connects the satellite to the receiver.



*Figure 4 - Layout of pseudo range measurements*

This project will be focusing on the pseudo-range code method. In Figure 4, the layout of the various sources of delays is shown. For each section of the layout, represented by a different color, various variables are taken into account. As this is merely an introduction to the GNSS system there is no need to delve deeper into the equation that combines these delay sources into one.

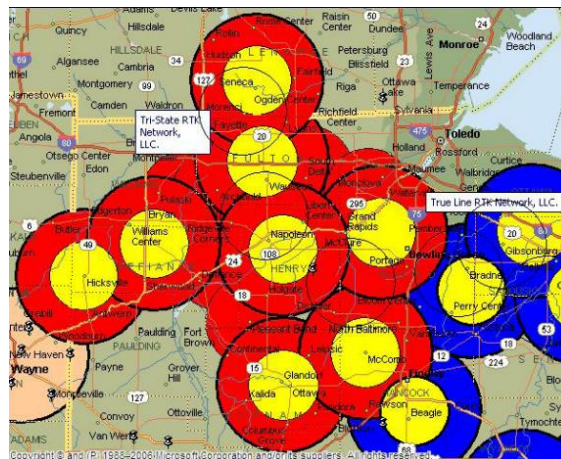
### 2.1.3 REAL-TIME KINEMATIC POSITIONING

RTK positioning was developed in the mid-1990s and it is a method of positioning that provides measurements in real-time with high accuracy (centimeter-level) using GNSS signals. This is possible by establishing communications between two systems, a reference (base) receiver and a rover receiver. The reference receiver transmits its raw measurements and corrections to the rover, and the rover must resolve ambiguities to give an accurate estimation of its position (Gakstatter, 2009).

The technique described above is traditional RTK, involving only one base signal and one rover, which has a limited range between the single base station and its rover, which is why multi-reference stations (Network RTK) are currently preferred. One of the biggest advantages of this implementation is the increased reliability of the service (Fotopoulos &

Cannon, 2001). Moreover, it improves the quality of corrections as one correction can be compared to others, creating what is called a virtual reference station, a nonexistent reference station situated closer to the rover than all existing stations.

This idea is born from the concept of RTK clusters, which consist of multiple reference stations managed by a single entity (Gakstatter, 2009). As Figure 5 shows, a typical RTK cluster is organized as such, with users in the yellow areas obtaining accuracy levels of an inch, while users in the red areas experience a 1–2-inch accuracy. However, in RTK clusters the user must decide which reference station to use, which is why Network RTK has become more popular, as RTK correction is based on most or all reference stations instead of one.



*Figure 5 - Typical RTK Cluster*

In the US, the development of Network RTK has vastly reduced the cost of implementing a new positioning service as the number of stations necessary drop from around 30 to approximately 5 to 10 in every 10,000 square kilometers (Wanninger, 2004). Currently, as stated in commercially operated RTK clusters and networks could be accessed for prices in the order of \$125 and \$500 a month, respectively, in 2009.

## **2.2 RELEVANT STANDARDS**

### **2.2.1 IS-200-GPS STANDARD**

The IS-200-GPS Standard (Dunne, 2018), defines the requirements for the interface between the Space Segment of the Global Positioning System and the User Segment of the GPS for radio frequency link 1 and link 2. These two links permit space vehicles to provide earth coverage signals for the GPS navigation system.

The standard includes information such as signals structures used. For example, the L1 signal has two carrier components modulated in quadrature with each other, with BPSK modulation, while modulation for L2 is chosen by ground command. As will be seen in further sections, this thesis takes full advantage of both of these bands with the GNSS Multi-Band Magnetic Mount Antenna.

Overall, the standard provides the framework for communications between satellites and ground stations, which act as the base for all GPS services.

## **2.3 DATA ANALYTICS**

### **2.3.1 U-BLOX FRAMEWORK**

U-blox is a leading provider of positioning and wireless communication technologies, specializing in global navigation satellite systems (GNSS) and cellular modules. The company offers a range of products and solutions for various industries, including automotive, agriculture, transportation, and IoT.

At its core, u-blox focuses on delivering precise and reliable positioning technologies. They design and manufacture GNSS receivers, which receive signals from satellite constellations such as GPS, GLONASS, Galileo, and BeiDou, to determine accurate location, velocity, and time information. These GNSS receivers integrate advanced algorithms and signal processing techniques to enhance positioning accuracy and reliability even in challenging environments.

In addition to GNSS technology, u-blox provides cellular modules that enable wireless communication capabilities. These modules support various cellular standards, including 2G, 3G, 4G/LTE, and emerging 5G networks.

Furthermore, u-blox offers software solutions, including positioning software, protocol stacks, and system-level software. These software components enable efficient integration of u-blox products into customer applications, ensuring optimal performance and functionality.

It is relevant to this project because the SparkFun ZED-F9P board used in the project works with u-blox technology, and relies on the SparkFun u-blox GitHub repository (Clark) for interfacing.

### **2.3.2 ARDUINO**

Arduino is an open-source platform used for all sorts of robotics and general electronics projects. It consists of two main aspects, a microcontroller and an integrated development environment (IDE) that allows the user to write and compile .ino files. The language is a simplified version of C++, and is proprietary to Arduino.

The microcontroller (hardware part of the system) works by connecting components to input and output pins, such as buttons, LEDs, sensors, and displays. By writing code, you can control and read data from these components, enabling you to build a wide range of projects.

Once the code is written, it is compiled and uploaded to the Arduino board through a USB connection. The microcontroller then executes the code, usually being based on a scan loop, as many other microcontroller applications.

Overall, Arduino simplifies the process of creating electronic projects by providing an accessible platform that combines hardware and software, bringing an attractive product to beginners who want to learn about microcontrollers and professionals looking for simplicity in their projects.

### 2.3.3 CAT BOOST

Cat Boost (ArcGis Pro 3.1, s.f.) is a tool developed by the company Yandex, and is one of the most potent gradient boosting algorithms available today. One of Cat Boost's unique capabilities is its capacity to effectively handle categorical variables without the need for manual preprocessing. The key machine learning concept behind it is known as boosting.

Boosting is an ensemble learning technique in machine learning that combines multiple weak models to create a strong predictive model. It works by sequentially training weak learners, such as decision trees, on subsets of the data, with each subsequent learner focusing on the mistakes made by the previous ones.

The weak learners' predictions are weighted and combined to make the final prediction. Boosting adapts to the data by assigning higher weights to misclassified instances, improving overall performance. The evolution of a boosting model can be found in Figure 6.

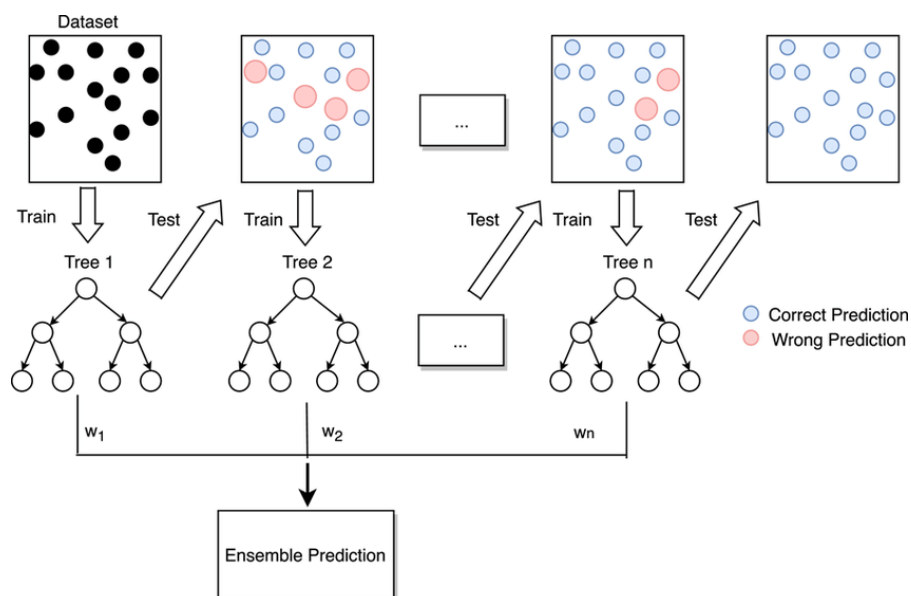


Figure 6 - Gradient Boosting Algorithm Methodology

However, contrary to conventional gradient boosting algorithms, which translate categorical features into numerical representations, Cat Boost makes use of a cutting-edge method known as "ordered boosting." This method uses an algorithm that interacts directly with

categorical features, preserving their fundamental characteristics and avoiding the loss of important learning information.

Cat Boost uses gradient-based one-hot encoding to effectively handle high-cardinality categorical features. By efficiently capturing the statistical significance of various categories while maintaining the algorithm's overall performance, this technique avoids the common problem of dimensionality explosion.

Cat Boost also uses a number of regularization techniques to prevent overfitting. These consist of the random permutations that add randomness to training to enhance generalization. In order to incorporate information about target variables into the encoding process and increase the algorithm's predictive power, ordered target statistics are also used.

These, accompanied by an overall better performance, are the main reasons Cat Boost is the chosen tool for the final model implementation.



## **Chapter 3. PRIOR ART**

### ***3.1 INTRODUCTION***

Although the research surrounding the combination of tractors, GNSS data and Machine Learning methodologies is in its early stages, there are some relevant findings that portray certain advances that are relevant to the scope of the project. Each of them will be thoroughly reviewed in this section, accompanied by an explanation that links it to the thesis.

These examples come mainly in the form of academic papers and patents, with publishing dates ranging from 2009 to 2021, in order to properly capture the current state of the art.

### ***3.2 PRIOR ART EXCLUSIVE OF PATENT INFORMATION***

#### **3.2.1 URBAN POSITIONING ON A SMARTPHONE: REAL-TIME SHADOW MATCHING USING GNSS AND 3D CITY MODELS**

In this paper (Wang, Grover, & Ziebart, 2013), research Ziebart looked for a way to improve the quality of GNSS communication with cell phones in urban cities, as the buildings blocked the signals as shown in Figure 7.

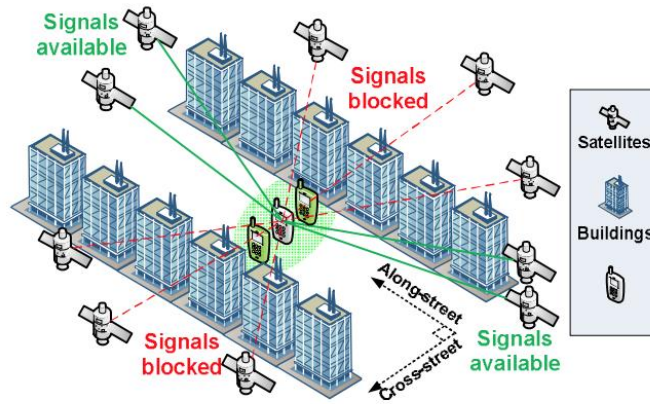


Figure 7 - Diagram of connections between smartphones and satellites

The machine learning algorithm they chose to focus on was a modified version of k-nearest neighbors, which is a data classification method that determines the likelihood of a certain element to be part of one of the groups that were previously classified. A visual representation of this modified kNN algorithm is shown in Figure 8.

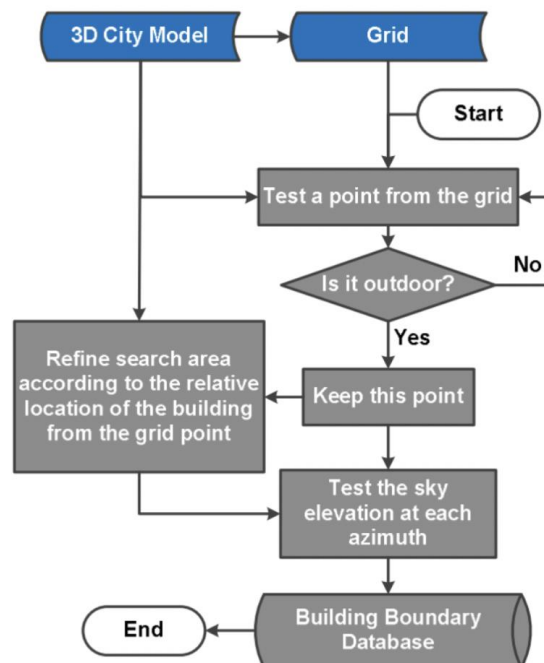


Figure 8 - Visual representation of algorithm

As far as the project is concerned, the Machine Learning methodology employed is relevant to the project, as it deals with similar data and its aim is practically the same, although with a different methodology.

### 3.2.2 GNSS POSITION ERROR ESTIMATED BY MACHINE LEARNING TECHNIQUES WITH ENVIRONMENTAL INFORMATION INPUT

This paper (Kuratomi, 2019) goes into depth about using machine learning techniques mainly involving decision trees to better estimate errors in the Global Positioning System and provides insight into the variables/features in the data that greatly affect the error in the location estimation process of the GPS.

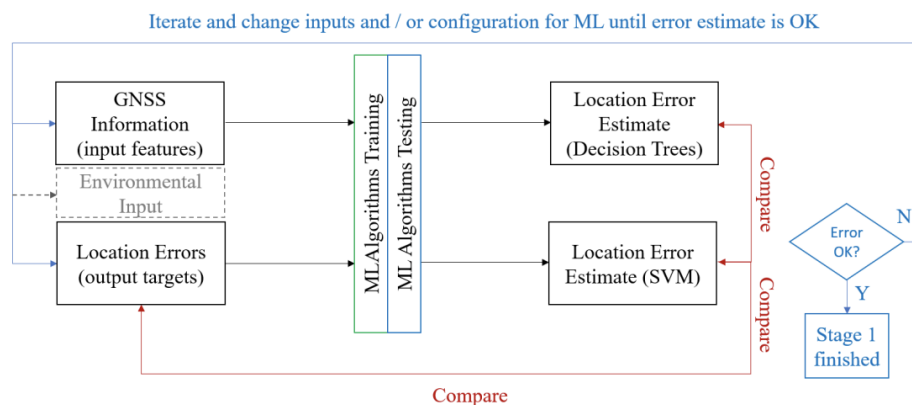
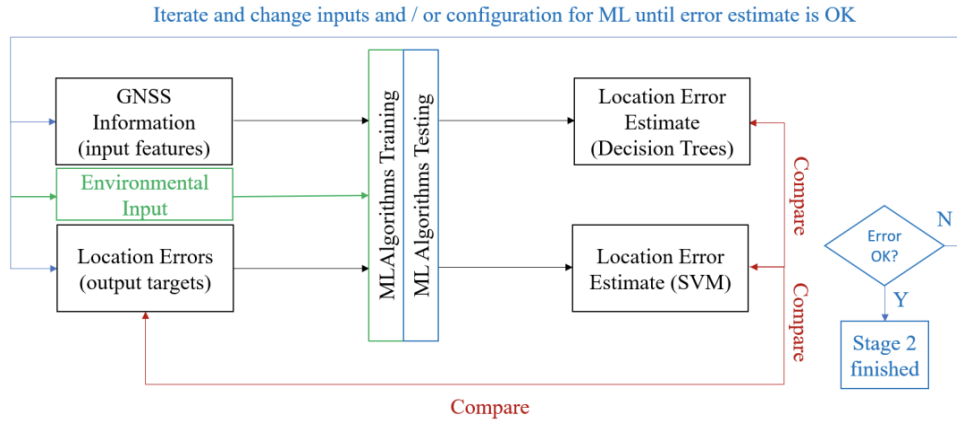


Figure 9 - First Stage in ML Process

First, he uses machine learning algorithms on two datasets corresponding to a truck and a lawnmower to minimize the prediction error close to the location error of the truck itself, as shown in Figure 9.

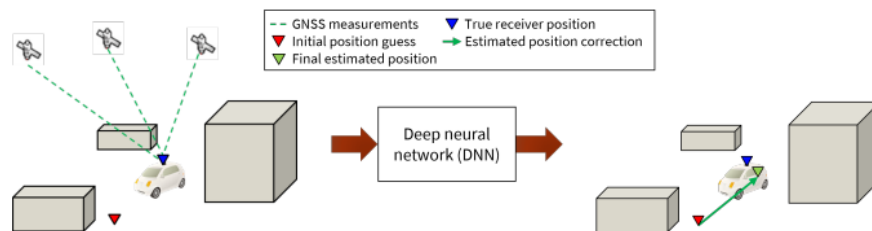


*Figure 10 - Second Stage in ML Process*

Finally, Figure 10 shows how the author provides environmental obstruction to see how this affects the receiver and the dataset and minimizes the error by seeing which variables are causing the most error. This allows the model to correct for the estimated error improving the overall prediction.

### 3.2.3 IMPROVING GNSS POSITIONING USING NEURAL NETWORK-BASED CORRECTIONS

This paper (Kanhere, Gupta, Shetty, & Gao, 2021) proposes utilizing Deep Neural Networks (DNNs) for GNSS positioning corrections, hoping to improve accuracy on positioning by adding said corrections to an initial guess. A general understanding of the aim of the paper can be seen in Figure 11.



*Figure 11 - Visual Representation of Corrections*

The main challenges facing the paper are wide variations in data due to the nature of GNSS and overfitting to available data, and they utilize set-based deep learning methods to tackle the entropy between data formats. They also add a data augmentation strategy which could prove relevant to the project.

While it does not attempt to predict shading, and DNNs are currently not being considered in this project, given their complicated deployment, the methodologies used are undoubtedly relevant, as they provide further insight applying Machine Learning to GNSS data.

### **3.2.4 ESTIMATING SUNLIGHT USING SIGNAL STRENGTH FROM SMARTPHONE**

In this paper (Y. Nishiyama, 2022), the authors attempt to come up with an accessible method to estimate UV exposure to users. The way they approach this is by trying to solve the same problem tackled in this project - using GNSS data from devices to estimate whether the user is in shade or not. They firstly collect a dataset of 4 days' worth of GNSS data + labels associated with whether the device is exposed to direct sunlight or not.

They then used features from the GNSS data (GNSS signal strength, azimuth and elevation of the satellite and the Sun, for both current time and preceding/following timepoints) as inputs to 12 different classification models. The entire pipeline for this process is shown in Figure 12 below.

Using 4-fold cross validation, they found that the radial basis function (RBF) kernel SVM was tied with quadratic discriminant analysis (QDA) for the best performance in accuracy, recall, precision, and F1 score. However, all models except for one managed to achieve at least 90% in all metrics, with these features as inputs.

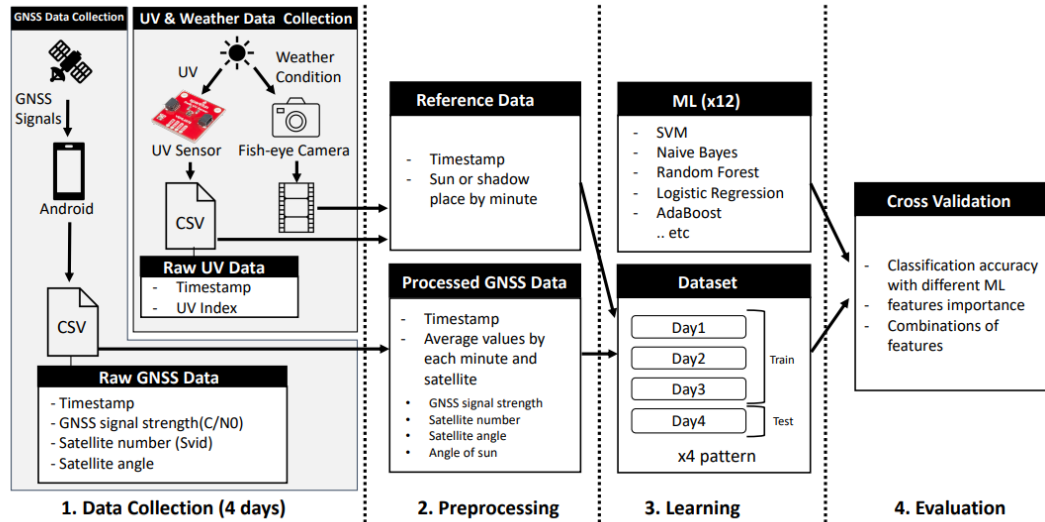


Figure 12 - Overview of Shading Detection Pipeline

This is very applicable to the project at hand, as the paper is essentially tackling the same problem, just in a different manner. The setup here (testing classification models using features from GNSS data, and verifying the best performing ones using cross-validation) is one that will probably be implemented, with classification models being the key focus.

### 3.3 PRIOR ART BASED ON PATENT INFORMATION

#### 3.3.1 PATENT JP2015068768A

This patent (Japan Patent n° JP2015068768A, 2013) was filed in 2013 in Japan and discussed a system that could be used to help with the interruption between a signal and receiver that takes place by a structure such as a bridge Figure 13. The solution that was discovered was a positioning device that takes the signal from the satellite and device and performs a distance calculation. This distance calculation can be used to calculate the integer bias of a carrier phase, leading to more accurate results.

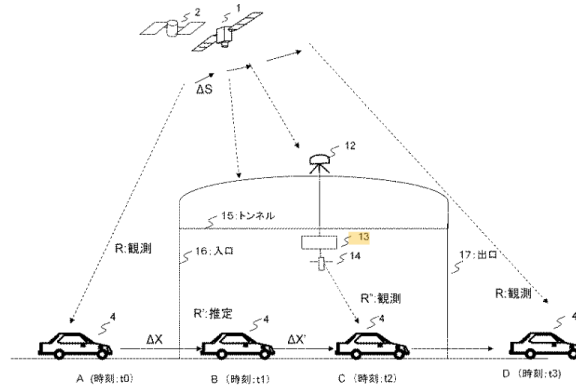


Figure 13 - Diagram of car passing under object

In contrast with the objectives of this projects regarding how to determine these shaded areas, the patent shows a method using distance calculation and their proposed positioning device. They propose a system in which their positioning device tracks the amount of time that has passed while the vehicle is experiencing the communication issue, and uses that to determine the movement of the vehicle.

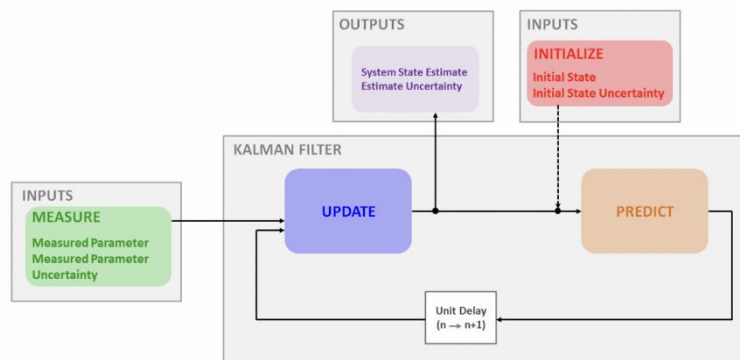
To this end, they use a gyro sensor that works as a distance sensor, allowing them to use acceleration to track the distance traveled by the car. Then they can use these values to track the movement of the satellite and the movement of the car, they can determine the carrier phase or the range between the satellite and the vehicle (Japan Patente n° JP2015068768A, 2013).

While this attempt to determine the interruption between satellite and receiver does not involve machine learning, it shows a basic understanding of how to deal with the communication errors that can take place with positioning and signals and which elements are important with it. The proposed solution involved the measurement model, a model that will be heavily involved in the project.

### 3.3.2 PATENT US20100283674A1

This patent (USA Patente n° US20100283674A1, 2010) is an American patent filed in 2015 that proposes a system to compensate and determine the shadow level of a given GNSS

region. It is based on a Kalman filter, an algorithm that produces estimates of data using multiple measurements. Kalman filters often used aircraft and ship positioning, which, as the system in the project at hand needs to do, takes into account real time and dynamic positioning, with a structure as shown in Figure 14.



*Figure 14 - Structure of proposed Kalman filter*

The patent looks to solve the same problem presented in this thesis, stating that there is a disadvantage for GNSS systems when satellite signal is interrupted. As data for the Kalman filter, it uses GNSS signals while also applying Dead Reckoning (DR), combining the two positioning techniques. This technique is further explained in (Rashid & Turuk, 2015).

The algorithm, as its corresponding flow chart suggests (no figure included as it is in Korean) first takes GNSS and DR measurements, and then matches them in a GNSS-DR processor. After that, it weighs the result against its estimated degree of shading (which comes from the map itself), and generates a second result. It then derives a third position information from the previous two and gives that as an output.

One drawback to the idea presented is that the measurements required for DR are not included in GNSS receivers, so if this were to be implemented additional hardware would be required, which could come in the form of the L26-DR module for GNSS receivers.



Although the idea proposed in this patent gives a rather different approach to the key problem, it is certainly relevant to the final design, as the design process taken can be viewed as parallel to the project at hand. DR may also be included in future works of the design.

### **3.3.3 PATENT US20210124059**

This patent (USA Patent No. US20210124059, 2021) and entails GNSS receiving methods to differentiate between direct waves from satellite GNSS signals or NLOS and multipath GNSS signals. GNSS signals are typically received from multiple paths, due to being reflected off of glass, metal, and wet surfaces before they are interpreted by the receiver. This is known as multipath interference and involves interpreting direct and indirect signals.

However, NLOS signals involve situations where direct signals are entirely blocked and only a reflected signal is received. Highly developed techniques are employed to reduce the interference from multipath to reduce the measurement model errors. With NLOS reception the errors are often in all respects different in receiver interpretation, as mentioned in (Inside GNSS, 2013). This patent introduces a novel and non-obvious object to provide a GNSS receiver easily capable of determining direct wave or NLOS signals from a satellite.

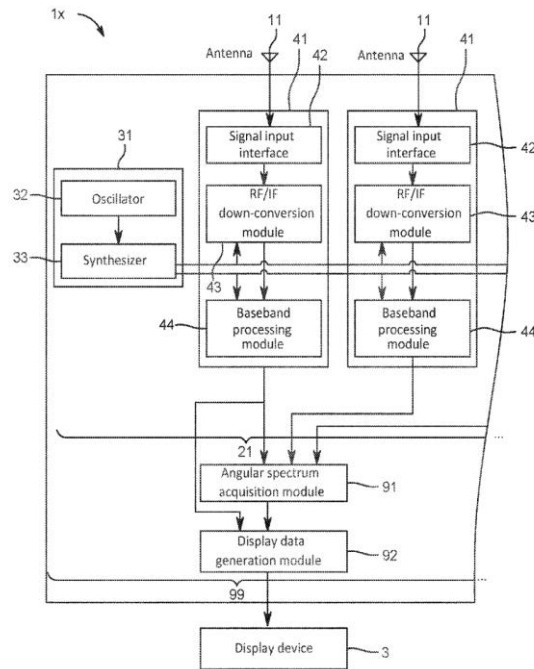
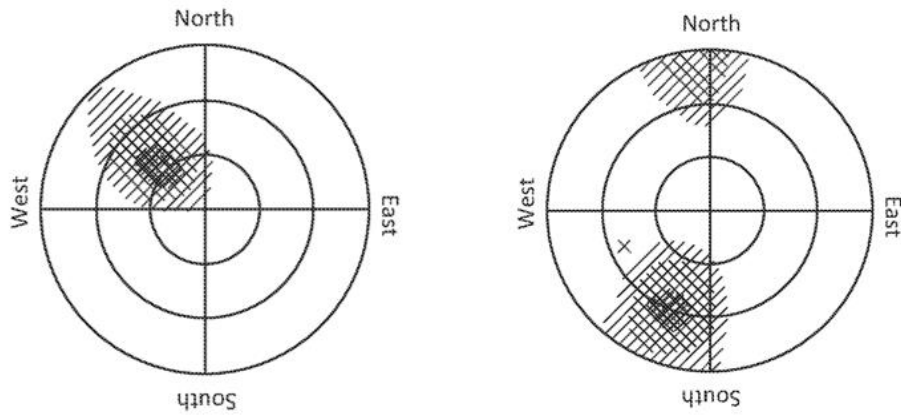


Figure 15 - Receiver configuration in the device

As seen in Figure 15, this configuration of a GNSS receiver is able to determine whether a received signal is direct wave or NLOS “based on the angular spectrum with respect to the estimation of the arrival direction of the GNSS signal” (USA Patent No. US20210124059, 2021). A property of angular spectrum exploited to differentiate between direct and NLOS is direct GNSS signals show a distribution with a single peak in a certain direction whereas NLOS will have zero or many peaks in undeterminable directions as seen in Figure 16. Within this configuration multipath interference is reduced and is not an issue, in fact would preferably include a multipath signal.



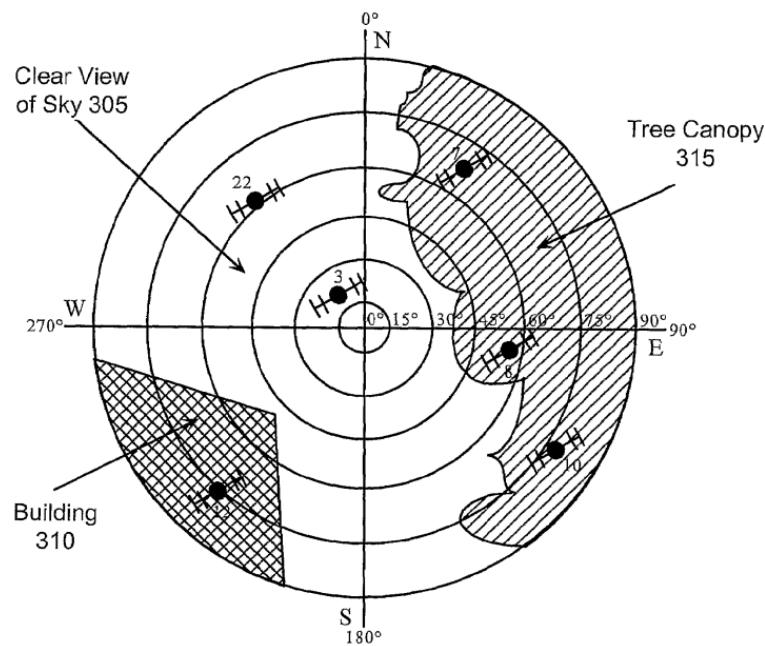
*Figure 16 - (Left) Single distribution peak from direct signal. (Right) Multiplexed signal from multipath or NLOS signals*

This patent includes hardware configuration to ensure proper detection of NLOS signals. The configuration requires a plurality of reception modules, a clock signal from a common clock, angular spectrum acquisition module, display data generation module to display angular spectrum, and as a key to detection having at least two antennas (USA Patent No. US20210124059, 2021).

This is applicable to the project as its focus is to predict NLOS signals and being able to train on data conditions preceding an NLOS occurrence will be crucial in during the modeling phase.

### 3.3.4 PATENT US7541975B2

This patent (Sever & Alison, 2009) discusses a system that uses a skyward facing sensor to classify points in its hemispherical view as sky, partial sky, or no sky, as shown in Figure 17.



*Figure 17 - Diagram of Sensor View*

Pseudo range and phase data from GNSS satellites determined to be in a region of sky can be considered reliable and used with confidence in a positioning solution. Pseudo range and phase data from GNSS satellites determined to be in a region of partial sky can be considered suspect and can therefore only contribute to a position solution with limited confidence and decreased accuracy (Sever & Alison, 2009).

### 3.4 CONCLUSION

In conclusion, the problem at hand has been approached on multiple occasions, although in a slightly altered format, presenting several instances of prior art that serve as guidance to the final system design. This prior art comes both in patent and patent-exclusive form,

providing a wide range of examples where similar problems have been tackled by experts on the matter.

It is clear how the current state of the art justifies the existence of the project at hand, as it shows how several advances have been made in the general field, but none in the way attempted in this bachelor's thesis.

## **Chapter 4. PROJECT DEFINITION**

### **4.1 JUSTIFICATION**

Given the state of the art presented above, it is clear the project is necessary, as it will give John Deere guidance in terms of their future research and development spending. If the final recommendation of the report is positive, it is likely that John Deere will spend time and resources delving deeper into the topic. However, a negative recommendation may lead them to pursue different objectives or employ a different methodology toward the same objective.

All in all, while the scope of the project is limited to John Deere's interests and pure scientific curiosity, it is important to recognize how, while the issue of shading currently relevant (as evidenced by the amount of prior art tackling it), it has never quite been addressed in the context of tractors and rural areas.

#### **4.1.1 POTENTIAL FUTURE DEVELOPMENTS**

In the future, these lines of research could have broader implications beyond John Deere's immediate interests. For example, if John Deere successfully develops innovative solutions to mitigate shading in rural areas, it could potentially lead to improvements in agricultural productivity and efficiency on a global scale.

Although quite marginally, if implemented, the project could potentially contribute to making farming more efficient, and therefore increasing farming yield. Moreover, the research conducted by John Deere could lead to advancements in related fields and industries. This is because machine learning is currently highly demanded in all fields, and advances in GPS technologies in agriculture could be used in other sectors.

Given the importance of agriculture and the increasing need for sustainable farming practices, addressing the issue of shading in tractors and rural areas could have far-reaching

implications. However, these implications will be further discussed through the framework of SDGs, found in ANNEX I: PROJECT ALIGNMENT WITH SDGs.

In conclusion, while the project initially focuses on addressing shading in the context of tractors and rural areas for John Deere's specific needs, it has the potential to make a significant change. This section is further developed in Parte I8.3.

## **4.2 OBJECTIVES**

Key objectives of the project can be broken down in the three key parts of the project, which are hardware, software and data analysis. As for the hardware portion of the project the objectives are the following

- To set up and utilize the GPS RTK SMA Kit F9P purchased correctly.
- To design a system that accomplishes two main objectives:
  - Facilitate data pipeline from the antenna to the Arduino MEGA board
  - Ensure a safe, easy and mobile way to collect data for extended periods of time, under several different conditions.
- To create an easy way to manually input the shading variable

In the software area, which mainly focuses on moving the data.

- To interface correctly with the GPS RTK SMA Kit F9P, and extract key information to a Python script.
- To successfully preprocess all the data to fit the desired models.
- To carefully validate data and ensure its quality and integrity.

Finally, in the data analysis aspect of the project the main objectives are the following

- To perform feature engineering avoiding calibration models and unimportant variables.
- To experiment with several different modelling techniques, tuning hyperparameters in order to maximize performance on each of them.

- To correctly evaluate each model, and pick out the highest performing one, ensuring it does not suffer from statistical skew due to overfitting to training data.
- To test the final chosen model under different shading materials to obtain an accurate evaluation.
- To use the results obtained to make an insightful final recommendation.

### **4.3 METHODOLOGY**

Given the previous section gave a full breakdown of the key objectives of the problem, the methodology can be broken down in two sections, combining the hardware and hardware interfacing aspects in one and leaving the data analytics part of the project separate.

Left out of this section are all reports and deliverables due to the University of Texas at Austin, which in a way guided the development of the project towards completing goals set in collaboration with technical and writing TAs and the project's director.

#### **4.3.1 HARDWARE AND INTERFACING**

In the hardware aspect, the first step is ordering the SparkFun GPS-RTK-SMA Breakout - ZED-F9P board, as it takes around 3 months to arrive from Denver, Colorado, where SparkFun is based. It is important to receive the board with enough time, as data collection is critical part of the project higher amounts of data collected will improve prediction capabilities.

Once the board arrives the system is mounted and the interfacing process begins. Once it is possible to both get information out of the board and store it somewhere the data collection part begins, and will continue for a few months in order to maximize the amount of available data. The schedule for data collection is explained in 5.3.2.



## **4.3.2 DATA ANALYTICS**

### ***4.3.2.1 Data Collection***

Firstly, it will be necessary to collect a smaller dataset with all possible variables for the board. The reason behind this is that the board has a total of 149 available variables, and in order to make a robust and relevant model it is imperative to lower that number to a more reasonable level, avoiding calibration variables, and also irrelevant variables.

To reduce the number of variables a feature engineering process will be used on the preliminary dataset, and from then, once the final variables for the model are selected, the actual data collection process will begin. However, it is possible that some of these final features are dropped along the process, while it is impossible to add new features to past readings.

### ***4.3.2.2 Model Training***

Once the data collection process has reached a certain stage, the group will begin to develop a few preliminary models, to test different approaches and attempt to choose the best one. Model training will be conducted as is commonplace in machine learning methodologies. Given the problem at hand is a binary classification one, which is rather common, there are plenty of options regarding algorithm selection, which is why developing the preliminary models to find the best performers is a key aspect of the project.

After this step, the model will be fine-tuned in order to maximize performance, following machine learning principles, such as maintaining separate training and evaluation sets, rigorously.

### ***4.3.2.3 Model Evaluation***

Although this step is undoubtedly related to the previous one, it is different in the sense that it only takes into account one final model and the tests performed on it. There are many ways to evaluate a binary classification algorithm, and they will be examined and explained further in Chapter 7.

The final model will be tested not only with the testing dataset, but also with special testing samples that measure the performance of the model under a given shading condition, such as different types of shade. These follow-up tests will have a lesser impact on the final recommendation but will nevertheless be useful to evaluate the performance of the model.

Once there is sufficient feedback on the performance of the model, and its hyperparameters have been fine-tuned to their best degree, the final recommendation will be constructed and presented to John Deere.

## **4.4 PLANIFICATION AND ECONOMIC ESTIMATES**

### **4.4.1 PLANIFICATION**

The overall planification of the project can be captured in Table 1, where the expected and actual time ranges for key objectives is included.

<b>Task</b>	<b>Expected Date</b>	<b>Actual Date</b>
<b>Initial project assignment</b>	September 2022	October 2022
<b>Hardware arrival</b>	November 2022	January 2023
<b>Hardware interfacing</b>	November 2022 – January 2023	January - February 2023
<b>Model development</b>	September 2022 – March 2023	September 2022 – March 2023
<b>Data collection</b>	December 2022 – February 2023	January 2023 – April 2023
<b>Final Report Writing</b>	March – April 2023	March – April 2023

<b>Final Presentation – USA</b>	April 2023	April 2023
<b>Thesis Writing</b>	May – June 2023	June 2023

*Table 1 - Project Planification*

As can be seen in Table 1, a legal issue between John Deere and The University of Texas at Austin regarding the intellectual property of all John Deere sponsored projects delayed the actual project assignments. The key hardware component for the project (GPS board) also took longer than expected to arrive as it met some specifications set by John Deere.

Despite these initial setbacks, the project was carried out in time and there were no issues on the US presentation and report. As for the final presentation in Madrid, sudden knee surgery slightly delayed thesis writing, but the defense is expected to be carried out in July 2023, according to the standards set by ICAI.

#### **4.4.2 ECONOMIC ESTIMATES**

The economic cost of this project was mostly covered by The University of Texas at Austin, institution where the project was carried out, and the costs are reflected in the bill of materials, included in Table 2.

<i>Bill of Materials</i>	<i>Item</i>	<i>Qty [Units]</i>	<i>Price [\$]</i>
	SparkFun GPS-RTK-SMA Breakout Kit	1	324.95
	Arduino MEGA 2560 Rev3	1	49.99
	QWICC cable	1	4.99

	Poster materials	1	8.74
	(UT Austin only)		
	Total		388.67

*Table 2 - Bill of Materials*

The set limit of the project in the University of Texas at Austin was of \$500, so the two largest expenses incurred were covered by this budget.

There were additional hardware requirements for the project, such as cables, a button, a breadboard, and many others. This equipment was supplied by The University of Texas at Austin free of charge, and are therefore not included in Table 2. All software used in this project was also free, hence the lack of software purchases in Table 2.

## **4.5 SAFETY AND ETHICAL CONSIDERATIONS**

While the use case for this project seems fairly benign, it is important to keep in mind certain considerations regarding the project. At this level, it is merely a research project that does not even use real user data, but if John Deere does decide to pursue this line of research certain situations might arise where it is important to take these considerations into account.

### **4.5.1 HEALTH AND SAFETY**

The design of this project does not immediately suggest issues related to health and safety, yet there are certain considerations to be examined. For example, GPS technologies can be critical in safety hazards, such as workplace accidents, which especially in the conditions these systems are used in (large fields and plantations), can provide first responders with a precise location of the target.

Even if John Deere mostly focuses on agricultural products, they do also manufacture certain off-road vehicles which might one day be involved in rescue missions. However, these concerns are barely tangent to the scope of the project and should not be given more thought.

## **4.5.2 ENVIRONMENTAL IMPACT AND SUSTAINABILITY**

It is important to note that the location history requires only periodic internet connection as not all farming locations will have sustainable connection to the mobile server. However, to counteract this, John Deere devices' remote location history can store its data locally and update as operations continue until a stable internet connection is acquired.

Additionally, since most John Deere automobiles use fuel and release emissions, it is important to note, if possible, to limit the distance traveled by tractors, maybe by using past history of traveled paths on a farm to limit the distance thus reducing the emissions released into the atmosphere.

However, this concern relates more to John Deere's business in general than to this project in particular. Furthermore, if implementation of this research ever occurs, vehicles will be more efficient and consume less resources.

## **4.5.3 PRIVACY AND DATA SECURITY**

One of the concerns of GPS tracking that has existed since its conception is privacy and security. Although this project is trying to improve one form of GPS that does not directly impact location data, it is important to note the problems that can arise from GPS tracking.

When a company possesses information about the location of a person, there is a risk of that information being leaked or that the company profits off of this data. The consumer must instead put their trust into the company and the company must respect the consumer's privacy and only use the location information for the uses it is intended to be used for.

This is the key reason behind John Deere not facilitating access to their databases for the purposes of this project. On the other hand, it gave birth to the data collection and hardware interfacing aspects of the project.

#### **4.5.4 INTELLECTUAL PROPERTY**

This thesis, as well as the overall project carried out in the University of Texas at Austin is bound by an intellectual property assignment and a non-disclosure agreement by John Deere. Any potentially patentable inventions and discoveries were required to be disclosed. Furthermore, the presentation of this thesis is done with the explicit approval of the John Deere legal team.

## Chapter 5. SYSTEM DESIGN

### 5.1 INTRODUCTION

In this section, the hardware and hardware interfacing aspects of the project will be covered, while the data analysis part will be left to the following chapter. The general hardware mapping of the final design can be found in Figure 18. Each component and its relevance to the project will be explained in detail in following sections.

This setup is meant to simulate what a John Deere tractor would have equipped on it, changing of course the Arduino with its proprietary CPU. It was purchased following their recommendations, and also introduces the constraint of not adding any additional hardware to the model, which greatly affected final results.

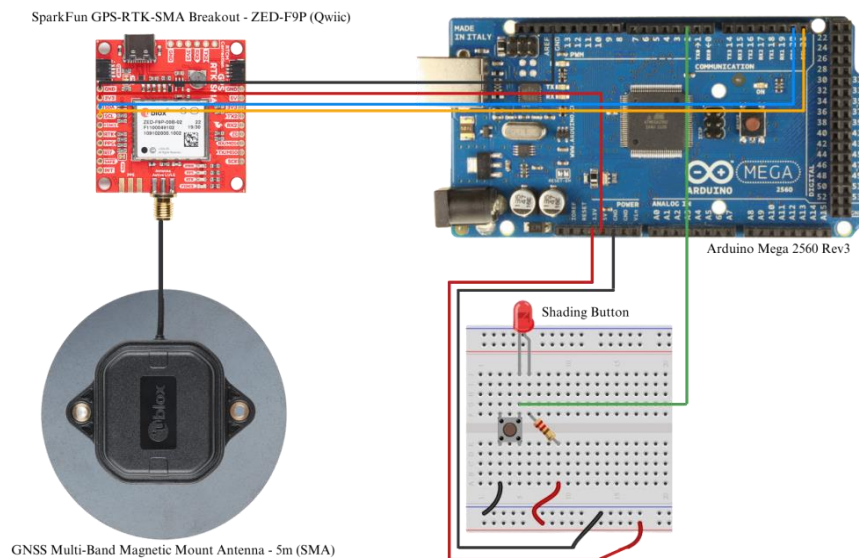


Figure 18 - Overview of hardware design

## 5.2 COMPONENTS

### 5.2.1 GNSS MULTI-BAND MAGNETIC MOUNT ANTENNA

In contrast to conventional GNSS/GPS antennas, the ANN-MB-00 GNSS (uBlox, 2018) multiband antenna is built to receive both the traditional L1 GPS band and the more recent L2 GPS band. The ANN-MB-00, featured in Figure 19, also comes with a magnetic base as well as mounting holes for additional anchoring.



*Figure 19 - GNSS Multi-band magnetic mount antenna*

This antenna features a high-performance multi-band RHCP dual-feed patch antenna element and a 5m SMA connection. It also supports GPS, GLONASS, Galileo, and BeiDou. It offers a quick, simple, and dependable multi-band antenna solution for the most recent u-blox F9 platform, which is the one used in the product, but it may also be used with any GPS/GNSS receiver that can take advantage of the L1/L2 dual reception.

It is offered by Sparkfun, the recommended provider of these items by John Deere, at a price tag of \$72.95, and was purchased within the GPS-RTK-SMA Breakout Kit. It is connected to the SparkFun GPS-RTK-SMA Breakout - ZED-F9P board through its included cable.

### 5.2.2 SPARKFUN GPS-RTK-SMA BREAKOUT - ZED-F9P

The most important piece of hardware in the project is undoubtedly this board (uBlox, 2023). The latest in a long line of potent RTK boards utilizing the ZED-F9P module from u-blox,



the SparkFun GPS-RTK-SMA raises the bar for high-precision GPS. It is based on the SparkFun GPS-RTK2 designs.



*Figure 20 - SPARKFUN GPS-RTK-SMA BREAKOUT - ZED-F9P*

The ZED-F9P is the pinnacle module for high accuracy GNSS and GPS location solutions, including RTK, and is capable of achieving three-dimensional accuracy of 10mm. Its physical aspect is shown in Figure 20. It is priced at \$274.95 on the SparkFun website, and was also purchased in the bundle with the antenna.

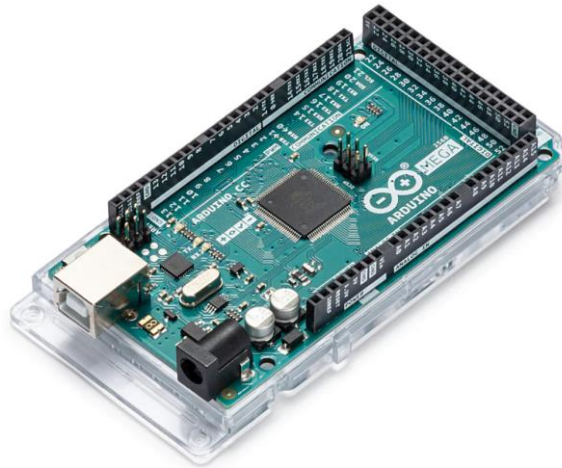
### ***5.2.2.1 Relevant pins & connections***

While initially the aim was to use a QWICC cable, which is able to establish an I2C or SPI communication with the Arduino using one single cable, the change from an Arduino to an Arduino MEGA meant those connections had to be done manually using 4 different cables. For this, the SCL and SDA pins were used to make the connection (apart from ground and a voltage source). The board was also connected to the antenna through its own cable, as can be seen in Figure 18.

### **5.2.3 ARDUINO 2560 MEGA REV 3**

Initially, the University of Texas at Austin provided a generic Arduino board to interface with the F9P board. However, it was found during the data collection process that the memory required to properly implement the uBlox library surpassed the Arduino boards

memory capacity. As the uBlox modules are strictly necessary to interface with the board there was no alternative but to purchase the MEGA version, seen in Figure 21, at a price tag of \$49.99.



*Figure 21 - Arduino MEGA 2560 Rev 3*

### ***5.2.3.1 Relevant Pins & Connections***

Relevant connections in the Arduino include the I2C connection to the F9P board, through pins SDA and SCK (Arduino, 2023). Also, it is connected to a breadboard which includes, as seen in Figure 18, a button and a LED light attached to it, in order to manually input the shading variable. The Arduino is also connected to a laptop, which receives its information.

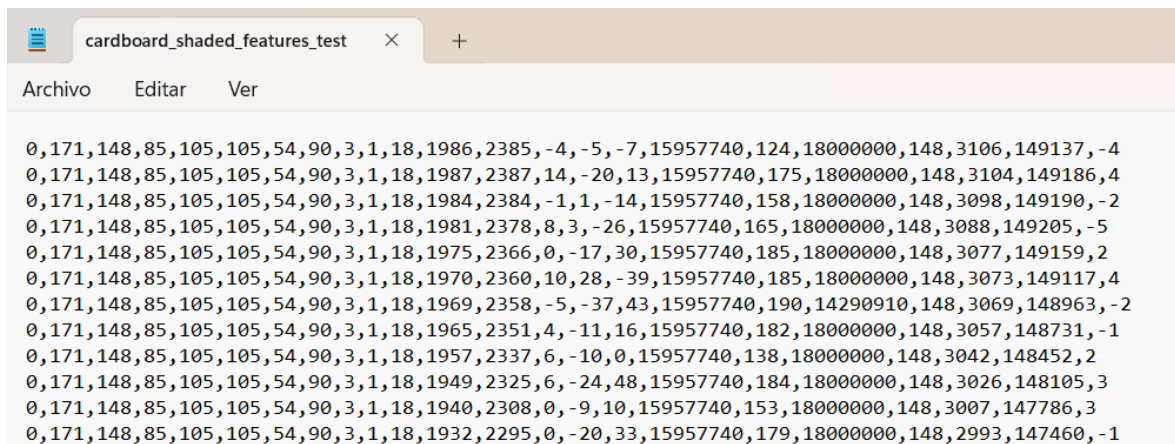
## ***5.3 DATA COLLECTION***

Deploying the Arduino code found in ANNEX II to the board, it is possible to collect data readings from the relevant variables (the process to select them will be commented later in Chapter 6. ). As readings take a little time to process, it is only possible to get one new reading every 20 seconds or so, which means that an hour-long data collection session would yield around 180 new values to the dataset.

For this reason, instead of writing the code to dump the readings into a csv file and use more memory it was decided to simply copy and paste from the console into a txt file, because

even if the performance gain was scarce, the amounts of data output by the board was always manageable.

At this point it might seem reasonable to include a figure showing an example of readings, but the board was returned to the University of Texas at Austin prior to the writing of this report. However, Figure 22 shows an example of one of the log files data was pasted on for use in Python. This particular example was collected using a cardboard box as the shading method.



```

cardboard_shaded_features_test
Archivo Editar Ver
0,171,148,85,105,105,54,90,3,1,18,1986,2385,-4,-5,-7,15957740,124,18000000,148,3106,149137,-4
0,171,148,85,105,105,54,90,3,1,18,1987,2387,14,-20,13,15957740,175,18000000,148,3104,149186,4
0,171,148,85,105,105,54,90,3,1,18,1984,2384,-1,1,-14,15957740,158,18000000,148,3098,149190,-2
0,171,148,85,105,105,54,90,3,1,18,1981,2378,8,3,-26,15957740,165,18000000,148,3088,149205,-5
0,171,148,85,105,105,54,90,3,1,18,1975,2366,0,-17,30,15957740,185,18000000,148,3077,149159,2
0,171,148,85,105,105,54,90,3,1,18,1970,2360,10,28,-39,15957740,185,18000000,148,3073,149117,4
0,171,148,85,105,105,54,90,3,1,18,1969,2358,-5,-37,43,15957740,190,14290910,148,3069,148963,-2
0,171,148,85,105,105,54,90,3,1,18,1965,2351,4,-11,16,15957740,182,18000000,148,3057,148731,-1
0,171,148,85,105,105,54,90,3,1,18,1957,2337,6,-10,0,15957740,138,18000000,148,3042,148452,2
0,171,148,85,105,105,54,90,3,1,18,1949,2325,6,-24,48,15957740,184,18000000,148,3026,148105,3
0,171,148,85,105,105,54,90,3,1,18,1940,2308,0,-9,10,15957740,153,18000000,148,3007,147786,3
0,171,148,85,105,105,54,90,3,1,18,1932,2295,0,-20,33,15957740,179,18000000,148,2993,147460,-1

```

*Figure 22 - Example of data collection log*

### 5.3.1 INITIAL DATA COLLECTION

The GPS board was able to produce 149 different variables in total, of which the objective was to keep between 10 and 25, which would produce a model complex enough to make accurate predictions and robust enough not to overfit to training data.

The issue with collecting this preliminary data is that, as a total of 149 variables were being used in the process, the board took much longer to collect them. With the final variables it took about 20 seconds to collect a single row, while with the full set of variables it took around 2 minutes, a very noticeable 6x increase.

### 5.3.1.1 Preliminary Data Collection Schedule

For this, a rudimentary version of the data collection schedule (Table 4) was created, using simply one shading source, which was trees, as it is believed that it was the most realistic shading source relevant for John Deere's purposes. Said schedule is found on Table 3.

<i>Shading</i>	<i>Time [minutes]</i>
Unshaded and static	40
Shaded with trees and static	40

Table 3 - Preliminary Data Collection Schedule

In total, two of the 80-minute cycles were executed, and the number of records obtained was 76, which were estimated to be enough to conduct the feature engineering. The full methodology of the process, along with the resulting data is described further in detail in 6.2.1.

### 5.3.2 DATA COLLECTION SCHEDULE

In order to make the most of data collection sessions and obtain a varied dataset, the group decided to collect data in cycles, including different types of shading. This schedule, found in Table 4, was made in collaboration with a John Deere representative who recommended certain shading types.

<i>Shading</i>	<i>Time [minutes]</i>
Unshaded and static	40
Unshaded and moving	40
Shaded with trees and static	20

Shaded with cardboard box and walking	20
Shaded in car and moving	20
Shaded under table (static)	20

*Table 4 - Data Collection Schedule*

With this schedule, based in 160-minute cycles, the dependent variable (shading), is balanced and varied. The result of the data collection is a final dataset of  $n=1,678$ , split into 1342 training samples and 336 validation samples, as per the 80-20 split already mentioned. In addition, for the extra tests conducted in 7.3, three additional datasets were collected. Their contents can be found in Table 5.

<i>Dataset</i>	<i>Size [n]</i>
General	1,678
2 <sup>nd</sup> test – unshaded	70
3 <sup>rd</sup> test – trees	77
4 <sup>th</sup> test – box	99

*Table 5 - Data Collection Results*

## **Chapter 6. MODEL DESIGN**

### ***6.1 INTRODUCTION***

This section will focus on the data analytics part of the process, describing some of the most relevant aspects of the methodology used, while final results will be left for the next chapter. This includes all content related to model evaluation, leaving this chapter focused in feature engineering, preprocessing and developing the final model.

This is the only section of this project where inputs from other team members mean it is not possible to go into all detail in certain parts as it would be appropriating their work. However, my personal contributions (especially in the later stages) are also significant and therefore relevant to this thesis. The main relevant aspect missing from this section is the comparison-based research that led to Cat Boost being the chosen tool for the final model.

### ***6.2 FEATURE ENGINEERING***

#### **6.2.1 PRELIMINARY DATA COLLECTION**

Before even starting the gross data collection process, it was necessary to firstly perform some form of feature engineering, as the GPS is able to return 149 variables in total, which not only were too many for the model, but also made the data collection process much slower, as explained in 5.3.1. The aim in this process was to get the number of features down to a reasonable number, in the range from 10 to 30.

The methodology followed for this process was an initial collection of 76 samples, divided into 53 training samples and 23 testing samples. Then, all variables considered to be related to location or time were dropped, as the intent of this project is to build a generalist model, not a calibration one. What this means is that the model must be able to perform shading detection anywhere in the world, and not rely on location data to make a prediction.

## 6.2.2 INITIAL CAT BOOST MODEL

After this step, a Cat Boost model was trained on the training samples, and feature engineering was performed on its variables. Then, the least important features in the dataset were dropped, resulting in with the final variables shown in Figure 23. A thorough description of Cat Boost and its usefulness can be found in Parte I2.3.3.

Attribute	Importance
FixType	18.442698
Elipsoid	13.782453
GnssFixOk	13.762762
HorizontalAccEst	8.319986
PDOP	6.230230
SIV	5.187164
VerticalAccEst	4.836642
SpeedAccEst	4.030833
PositionAccuracy	3.455373
ElipsoidHp	3.251041
NorthingDOP	3.132855
VerticalDOP	3.048282
Heading	2.653978
HeadingAccEst	1.622575
NedEastVel	1.318406
HorizontalAccuracy	1.135533
NedNorthVel	1.094238
GeometricDOP	0.841857
VerticalAccuracy	0.820027
HorizontalDOP	0.803515
EastingDOP	0.781525
PositionDOP	0.532897
TimeDOP	0.484009
NedDownVel	0.431122

Figure 23 - Final Variables chosen

All variables are broken down and explained in 6.3.1. Furthermore, the refined model was tested on the testing dataset and the confusion matrix shown in Figure 24 was obtained, resulting in an accuracy score of 95.65%. This value, despite not being particularly relevant to the overall model, shows the model is pointed somewhat in the right direction.

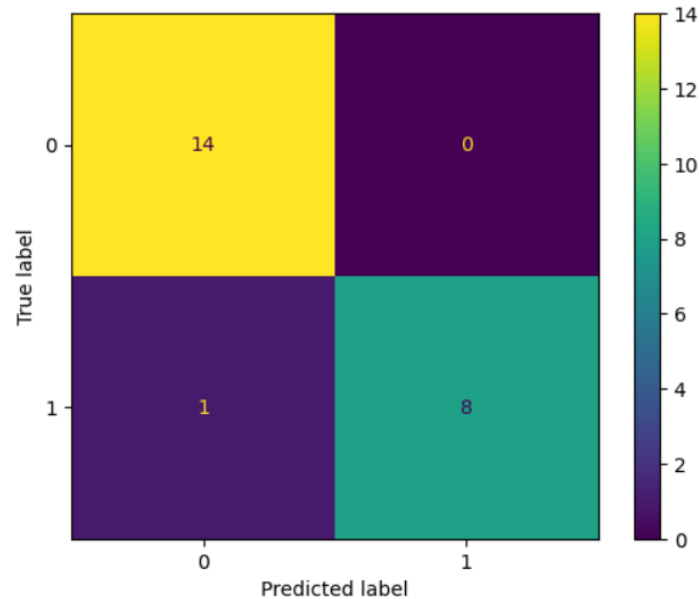


Figure 24 - Confusion Matrix for Feature Selection Model

## 6.3 FINAL MODEL

After the feature training process, and the elimination of all variables that were influenced by calibration factors, such as latitude, longitude, altitude or time, the final model was trained using all available samples from the data collection. In the following sections the final 22 variables of the model will be explained alongside the methodology followed throughout the process.

### 6.3.1 VARIABLE EXPLANATION

In the following bullet points, each variable used in the final model will receive a brief explanation.

1. FixType: Indicates the type of fix obtained from the GPS receiver, representing the quality and reliability of the position fix. It categorizes the fix as either valid or invalid based on the available satellite signals and their strength.



2. GnsFixOk: Indicates whether a valid Global Navigation Satellite System (GNSS) fix has been obtained. It signifies whether the receiver has successfully acquired and locked onto GNSS satellite signals for accurate positioning.
3. HorizontalAccEst: Represents the estimated accuracy of the horizontal position.
4. PDOP (Position Dilution of Precision): Indicates the dilution of precision in the position estimation due to the geometric configuration of the satellites. A lower PDOP value indicates a better satellite geometry, leading to improved position accuracy.
5. SIV (Satellites in View): Represents the number of satellites in view of the GPS receiver at a given time. It indicates the availability of satellite signals and can be used to assess the reliability of the position fix.
6. VerticalAccEst: Represents the estimated accuracy of the vertical position.
7. SpeedAccEst: Indicates the estimated accuracy of the GPS receiver's speed measurement.
8. PositionAccuracy: Represents an overall measure of the accuracy of the position fix obtained from the GPS receiver. It combines information from various sources, such as satellite geometry, signal quality, and receiver capabilities, to provide an estimation of the overall position accuracy.
9. NorthingDOP: Represents the dilution of precision specifically in the northing or y-coordinate estimation.
10. VerticalDOP: Indicates the dilution of precision specifically in the vertical or z-coordinate estimation.
11. Heading: Represents the direction in which the GPS receiver is moving.
12. HeadingAccEst: Indicates the estimated accuracy of the heading measurement.
13. NedEastVel: Represents the velocity or speed of the GPS receiver in the eastward direction.
14. HorizontalAccuracy: Represents the estimated accuracy of the horizontal position fix.
15. NedNorthVel: Represents the velocity or speed of the GPS receiver in the northward direction.

16. GeometricDOP: Indicates the geometric dilution of precision, which represents the influence of satellite geometry on the accuracy of the position estimation.
17. VerticalAccuracy: Represents the estimated accuracy of the vertical position fix.
18. HorizontalDOP: Represents the dilution of precision specifically in the horizontal position estimation.
19. EastingDOP: Indicates the dilution of precision specifically in the easting or x-coordinate estimation.
20. PositionDOP: Represents the dilution of precision in the overall position estimation.
21. TimeDOP: Indicates the dilution of precision in the time synchronization between the GPS receiver and satellite signals.

### 6.3.2 MODEL FEATURE IMPORTANCE VALUES

As Cat Boost does not implement any sort of linear or logistic regression, there are no model coefficients. However, the feature importance values used in the feature engineering process can serve as accurate descriptors of the weight each variable holds in the model. The importance values are expressed in percentages, and the table is shown in Table 6.

<i>Index</i>	<i>Attribute</i>	<i>Importance [%]</i>
0	GeometricDOP	0.799674067
1	PositionDOP	1.355290658
2	TimeDOP	1.912957378
3	VerticalDOP	1.615021837
4	HorizontalDOP	1.442351034
5	NorthingDOP	1.033227545

6	EastingDOP	0.738594482
7	FixType	6.86240713
8	GnssFixOk	13.4831487
9	SIV	4.126137132
10	HorizontalAccEst	22.50747491
11	VerticalAccEst	6.801200587
12	NedNorthVel	2.146814374
13	NedEastVel	4.298854127
14	NedDownVel	1.026027504
15	Heading	8.009403578
16	SpeedAccEst	8.866057875
17	HeadingAccEst	1.7581399
18	PDOP	4.082316065
19	PositionAccuracy	3.917693599
20	HorizontalAccuracy	1.313558335
21	VerticalAccuracy	1.903649175

*Table 6 - Attribute Importance Values*

The importance values shown in Table 6 are quite similar to those obtained in the feature engineering process, which are shown in Figure 23. This makes sense since the variables used in the final process are practically the same, only these values have considerably more training set depth.

The main difference comes from the elimination of the “Elipsoid” variable, which was found to be location related, as it refers to the altitude above the ellipsoid that is used to represent the Earth in certain GPS systems.

### **6.3.3 METHODOLOGY**

For all processes described in this section, the Python library used was Scikit-Learn, one of the leading machine learning libraries currently available in Python. As is the standard with machine learning models, samples are first divided in training and validation sets. As the dataset in use is reasonably large, it is possible to do an 80-20 split, as 20% of data should be enough to properly validate data.

Data is then normalized using the Scikit-Learn standard scaler, which converts all numerical variables into values between 0 and 1. Variable scaling is important in machine learning to ensure fair representation of features, enhance model performance, speed up convergence, provide balanced regularization, and enable accurate interpretation. Scaling equalizes variable influence, improves optimization, and makes it easier to compare the impact of different variables in the final model.

Once the data is ready, the training sample is fed to Scikit-Learn’s own Cat Boost implementation for model training. Hyperparameter tuning was found not to be necessary in this case, due to the fact that Cat Boost has well-optimized default hyperparameters. Its built-in features, like gradient-based learning and ordered boosting, make it less sensitive to hyperparameter variations, making manual tuning less crucial for achieving good performance.

However, even if hyperparameter tuning as such is not necessary, certain details of the model commented in the following section did include some research to decide aspects such as the loss function of the model or its learning rate.

### **6.3.4 ADDITIONAL MODEL INFORMATION**

The CatBoostClassifier object used by the Scikit-Learn implementation of Cat Boost gives the user some additional information to better understand the model training process. Moreover, it offers ample room for model tuning, even if, as mentioned in prior sections, numerical hyperparameters do not need to be tuned by hand in Cat Boost.

#### ***6.3.4.1 Number of Trees or Iterations***

One of these hyperparameters, which is not discussed below, is the number of trees, sometimes also known as iterations. Here, the default set by the Cat Boost object is of  $n=1,000$  trees, and after conducting experiments with other values ranging from  $n=100$  to  $n=2,000$ , the conclusion was that  $n=1,000$  was indeed the best value.

In order to visualize how we came to this conclusion, Figure 25 shows a plot between the natural logarithm of the loss function evolved as iterations passed. This experiment was conducted on one of the preliminary models, and therefore the final loss value obtained is far inferior to the one in the final model, but it still shows clearly why 1,000 trees is the correct amount, as the loss seems to converge at that value, and including more would likely lead to overfitting to the training data and not show a considerable improvement.

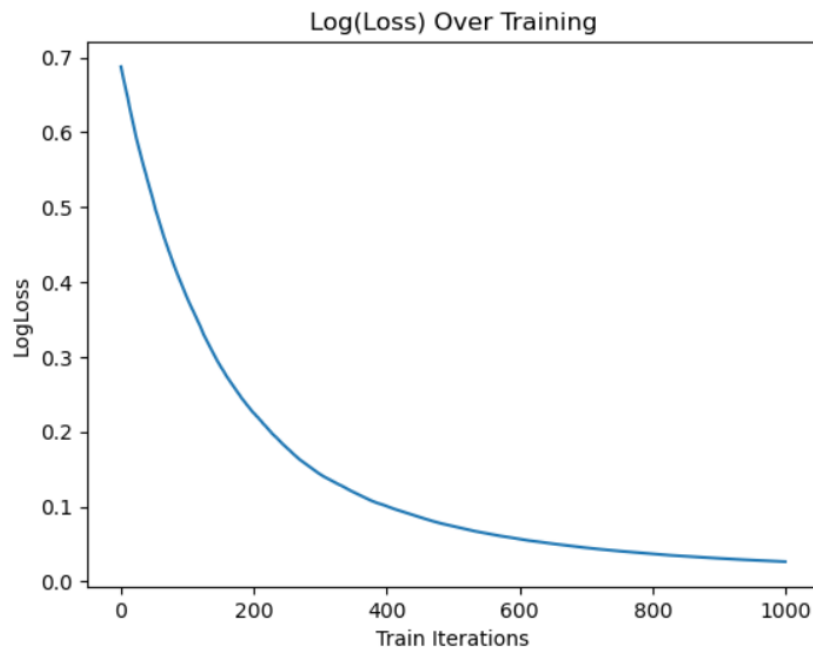


Figure 25 - Evolution of Loss in Preliminary Model Training

### 6.3.4.2 Loss Function

It is relevant to point out the loss function used here is not related to the model evaluation aspect of this project, but refers to the internal mechanism used in the training of the model. In Table 7, some of the possible loss functions offered for classification problems by the Cat Boost algorithm, along with their respective formulas. All the figures containing the formulas in Table 7 are sourced directly from (Yandex, s.f.).

<i>Loss Function</i>	<i>Formula</i>
Cross Entropy	$\frac{-\sum_{i=1}^N w_i (t_i \log(p_i) + (1 - t_i) \log(1 - p_i))}{\sum_{i=1}^N w_i}$

Logloss	$-\frac{\sum_{i=1}^N w_i (c_i \log(p_i) + (1 - c_i) \log(1 - p_i))}{\sum_{i=1}^N w_i}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F score	$(1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall}$
F1 score	$2 \frac{Precision \cdot Recall}{Precision + Recall}$
MCC	$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$

Table 7 - Possible Loss Functions

After some research, precision, recall, F score, F1 score and MCC were dropped as candidates for the model, as for classification models as powerful as CatBoost it is recommended to use either cross entropy or the logloss metric. To decide between the options, they were both implemented on the preliminary models, and logloss was found to work best, so it was included in the final model. This is why in Figure 25 it appears on the Y-axis.

### ***6.3.4.3 Learning Rate***

Given Cat Boost is based on a gradient descent approach, it is imperative to establish a learning rate. It is the most important hyperparameter in any of these models, as having it too low or too high can lead to the model not converging. At first, the approach was to try different values and see what worked best.

However, it was quickly discovered that this process is not necessary after all, as the Cat Boost library is able to obtain the optimal value, by simply passing a null object (“None” in Python) to the CatBoostClassifier constructor method. In the case of the final model, the final learning rate determined by the algorithm was of 0.002939. Considering the industry standard is usually set between values of 0.0001 and 0.001, the value is quite reasonable.

## ***6.4 CONCLUSION***

Without commenting the concrete results of the model training, as it would overlap slightly with the following sections, it is possible to draw certain conclusions from the training process. It is undoubtedly one of the more creative aspects of these kinds of projects, as there are practically unlimited algorithms and methodologies to follow, each combination of them resulting in different models with their advantages and disadvantages.

In the case of this project, these decisions were made mainly following what the group considered to be the best practices in each case, in the train-test split and the scaling of variables, for example. In cases where the decision was not clear an experimental approach was always preferred, as it leads to a model that will be better tailored to deal with the particular dataset used in the problem.



## Chapter 7. MODEL EVALUATION

### 7.1 INTRODUCTION

In this section, different evaluation techniques for binary classification models will be analyzed. Although only the final Cat Boost model is presented in Chapter 6, many others were developed in order to find the highest performing one. For this, it was not enough to simply rank them by accuracy, which is the most common statistic, because of the specific nature of the problem. However, only the results for the final model will be included in the

### 7.2 EVALUATION TECHNIQUES

In any classification problem, the most commonly seen metric for determining whether a model works or not is accuracy or hit-rate, described in (1). In a model with little or no context, it provides a general understanding on how many times a prediction will be right

$$(1) \textit{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

However, the requirements posed by John Deere made it clear that false positives, which are instances where shading is detected yet is not present, are more favorable than false negatives, cases where shading is present yet goes undetected. That is why the measure of recall (2) is also introduced, which is designed to measure the sensitivity of the model.

$$(2) \textit{Recall} = \frac{TP}{TP + FN}$$

As the equation shows, higher values of false negatives negatively impact the recall metric. Also, more complex metrics can be introduced to the model to give a more detailed measure of performance. This is the reason behind incorporating the area under curve, or AUC, into the analysis. It is a widely used evaluation metric for binary classification problems. It

quantifies the performance of a model by measuring the area under the ROC curve. The ROC curve plots the true positive rate against the false positive rate at various classification thresholds.

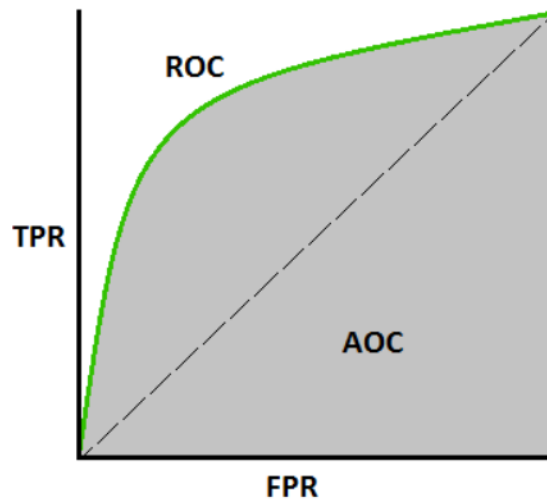


Figure 26 - AUC Relationship with ROC curve

As Figure 26 indicates, the ROC curve plots the relationship between the false positive rate (FPR) and true positive rate (TPR), and the AUC is the area under said curve. In an ideal scenario, the curve is a square that covers the entire two-dimensional space from 0 to 1, resulting in a perfect AUC score of 1.

AUC represents the probability that the model will correctly rank a randomly selected positive instance higher than a randomly selected negative instance. It provides a concise summary of the model's discriminative power, with higher AUC values indicating better overall performance.

### 7.2.1 SHADING SCORE

In order to combine the 3 metrics discussed above, the shading score metric was created. It has no relevance outside of this project, and it was created to combine all metrics into one. The final formula for the shading score (3) was determined somewhat arbitrarily, but with the approval of a John Deere representative.

$$(3) \textit{Shading Score} = \frac{\textit{Accuracy} + \textit{Recall} + \textit{AUC}}{3}$$

It is clear it just consists of a simple average of all the three mentioned variables, which have equal weight and take values between 0 and 1. As the dataset at hand is relatively balanced, due to the rigorous data collection schedule, it is heavily expected that AUC and accuracy scores will be quite similar, but giving 33% of the weight to the recall metric seems like a reasonable amount, as it is still very important to get results right.

### **7.3 MODEL TESTING**

In order to provide a more detailed solution, it was decided to carry out 4 different tests on the model. The first is a general test, done on the validation sample, which can be regarded as the one to be expected in these situations. The second was testing the model on a purely unshaded dataset, collected after the model training. The third and fourth tests followed the same methodology, only with a shaded dataset. The difference between them is that one used trees as its shading source and the other used a cardboard box.

As can be easily seen with the formula for the shading score (3), only the first test can use this metric, as the other three would have either no positives or no negatives, so for these tests the performance will be measured with the hit-rate (accuracy).

It is important to note that in all tests, the shading variable is active-low, which means a 0 means shading while a 1 means not shading. This is simply due to the button used in data collection, as it was also active-low and it was pressed during shaded conditions. This should not affect at all the obtained results as these values are just labels.

#### **7.3.1 GENERAL TEST**

For the general test, which is considered the most important one, the confusion matrix in Figure 27 is obtained. It is considered to be more important than the others because it assesses the model as a whole, and has a significantly higher amount of testing data, n=336 (Table 5).

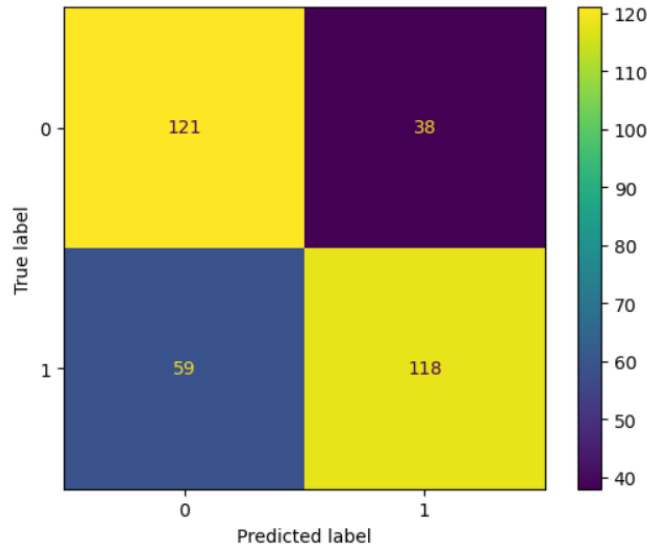


Figure 27 - Confusion Matrix for the Validation Dataset

### 7.3.1.1 Result Calculation on First Test

In order to calculate the shading score metric, it is necessary to first compute the three metrics that make it up. Accuracy and recall can be easily calculated from the confusion matrix, by using formulas (1) and (2), respectively. The accuracy comes out to 71.13%, and the recall is 76.1%, taking into account the active-low characteristic of the dependent variable.

Calculating the AUC is more complicated, as it involves calculating the ROC curve, which in turn is obtained by calculating the true positive rate (TPR) and false positive rate (FPR) at every possible threshold. Luckily, Python provides the tools to do all of this in just one line, with the Scikit-Learn function `roc_auc_score()` (Scikit-Learn, n.d.), which in this case outputs the value 71.38%. As expected from a balanced dataset, the AUC score and the accuracy of the model are almost identical.

These three variables are now used as the input for formula (3) to get a shading score of 72.87%. The value obtained does not meet the set expectations, which at first were of 85% but were later changed to 80%, as the amount of data was considerable and the main

hypothesis upon which this thesis is based indicates that shading should be clearly detectable using GNSS data. However, as the next chapter will explain in more detail, this does not mean the final recommendation should be fully negative, as there are other factors at play.

### 7.3.2 TEST ON UNSHADED DATA

For the test on data collected on open sky, the confusion matrix shown in Figure 28 was obtained, from the testing sample of  $n=70$  instances, as mentioned in Table 5.

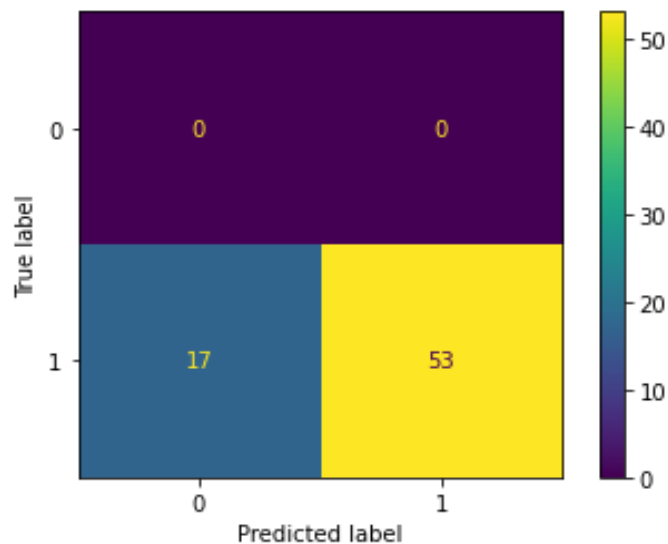
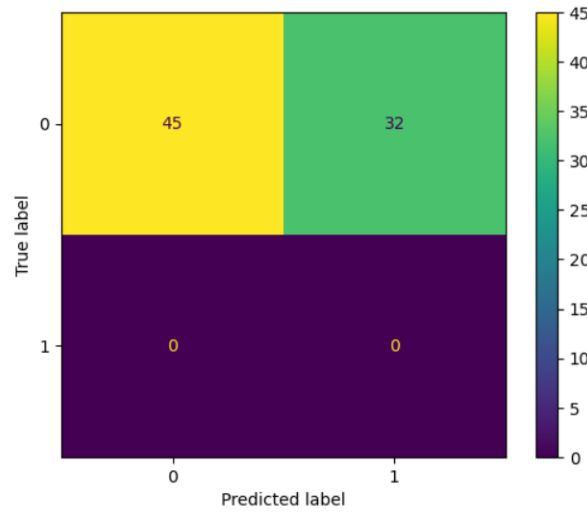


Figure 28 - Confusion Matrix for Unshaded Test

As can be seen, the accuracy of the model on this dataset is of 75.7%, which means the model was able to predict non-shaded instances a little over three out of every four measurements.

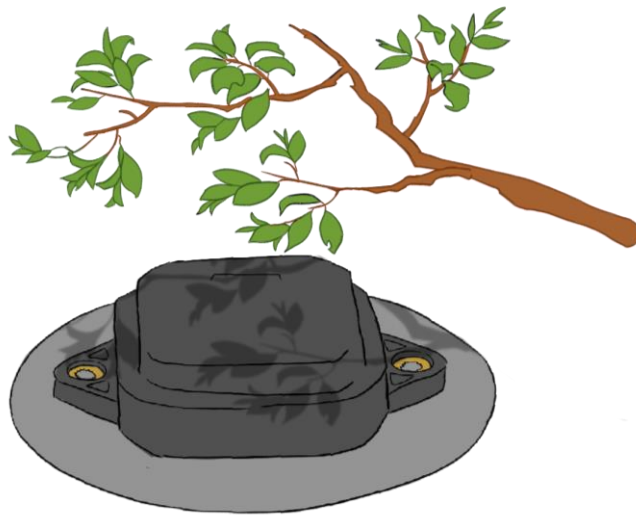
### 7.3.3 TEST ON TREE DATA

The confusion matrix for the third test, which tested the model with newly collected data of the board sitting under trees, is seen in Figure 29.



*Figure 29 - Confusion Matrix for Shaded by Trees Test*

In this scenario, the accuracy drops to 58.4%, which can be interpreted as failure in general, as it is only 8.4% superior to a random guess, which would yield a 50% hit rate. These results will be commented in Chapter 8. A visual representation of the test is shown in Figure 30.



*Figure 30 - Test under tree branches*

### 7.3.4 TEST ON CARDBOARD BOX DATA

For the final test, conducted with a sample size of  $n=99$ , a little higher than the other two for no particular reason, the methodology followed was the same as in other cases. The resulting confusion matrix is shown in Figure 31.

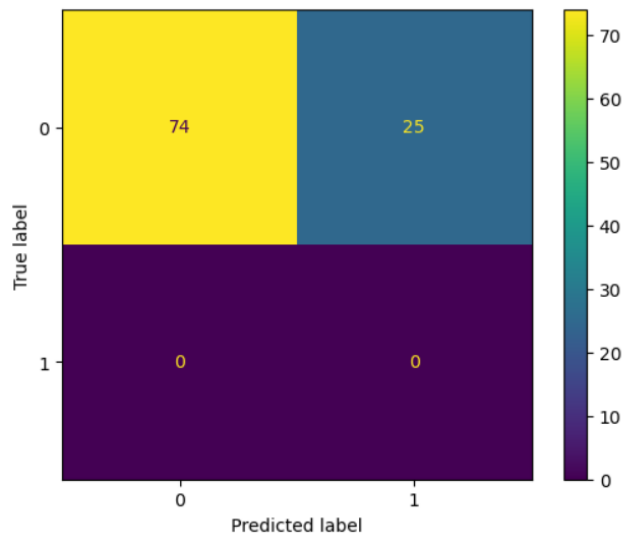
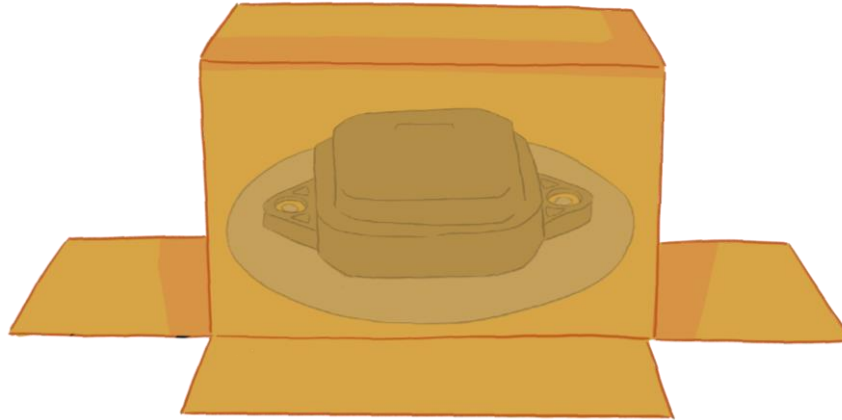


Figure 31 - Confusion Matrix for Shaded by Cardboard Box Test

In this test, there is a relative increment in accuracy in relation to the tree shade case, which can be easily explained by the fact that the shading created by a cardboard box is more potent than the one created by trees, as trees leave some space for GNSS signals to reach the antenna, while the cardboard box blocks it even more.

These results will be commented and explained in detail in the following chapter. In Figure 32, a diagram showing how the test was conducted can be seen.



*Figure 32 - Test Under Cardboard Box*

## **7.4 CONCLUSION**

Although the specific conclusions regarding the obtained data are left to Conclusions & Future Works, it is also relevant to remark the importance of a well-defined model evaluation system. The key aspect is that it must always be aligned with the objectives of the developer.

It is easy to lose sight of the purpose of the model among the many intricacies of the model development phase, and also to attempt to implement convoluted evaluation mechanisms just for the sake of complexity, so it is crucial to always keep in mind the purpose of the model.

### **7.4.1 MODEL EVALUATION ON IMBALANCED DATASETS**

In the case of this project, the created metric of shading score captures the objective of the model perfectly. It can be argued that the inclusion of the AUC metric is rather absurd, as it is mentioned above that in most cases it coincides with the accuracy. However, this is only the case in balanced datasets, such as the one we collected (purposely achieving this balance between shade and no shade).



In the real world, and by extension in a hypothetical dataset used by John Deere, tractors are set to experience shading in only a small fraction of the instances. In this case, the inclusion of the AUC metric helps to compensate this effect.

Furthermore, the current design takes into account the possibility of working with an imbalanced dataset not only by including the AUC in its evaluation framework, but also by using Cat Boost. Cat Boost is an ensemble method for machine learning, which combine different predictions from different models to make their final predictions.

Other techniques to combat an imbalanced dataset are certain preprocessing techniques such as oversampling, which works by creating synthetic cases of the underrepresented case from data from present cases, and undersampling, which just eliminates instances of the overrepresented class.

## Chapter 8. CONCLUSIONS & FUTURE WORKS

### 8.1 *GENERAL CONCLUSIONS*

Given the final results achieved in the previous chapter, the final model, while not quite meeting expectations, could be used in a positive recommendation, as the data John Deere has to work with could very likely be used to train a more accurate model, but first it is important to take into consideration some limitations in the project at hand.

#### 8.1.1 RECOGNIZED LIMITATIONS

##### 8.1.1.1 *Calibration Issues*

The largest issue found with the project is that of generalizing to a wider dataset. As much as all location or time related variables have been avoided throughout the model training part of the project, it is impossible to say for certain that all of the variables included in the final model do not contain any location data.

This is mainly because the project is about a GPS board, and its main purpose is to provide location data, so it is complicated to obtain information that is verifiably free of location influences.

Also, while the data collection process ensured the data collected was as varied as possible, all collected data in the end comes from around the same neighborhood in Austin, Texas. This probably influenced the final model in ways that are impossible to quantify unless it was possible to collect data and test the model elsewhere.

##### 8.1.1.2 *Insufficient Training Data*

Even though the data collection process took place during most of the project's overall duration, 1,678 samples are very likely not enough, given the amount of variables and overall complexity of the model. In (Melvin, 2021), it is argued that, as a rule of thumb, no

less than 1,000 samples per class should be used in any problem. Of course, this is only a rule of thumb and does not apply to simpler models, but removing the testing dataset, the number of samples per class collected comes out to a total of 671 samples.

### ***8.1.1.3 Representativity***

Another issue source of uncertainty found in the project comes from the lack of proof of representativity. What this entails is that, even if the model was generalist (no calibration data was taken into account), trained on a larger sample size and received high testing accuracy scores, there is no way of proving that a tractor that is making real time predictions will be able to detect shading properly.

This is due to the fact that the model on data collected in a specific environment, with varied sources of shading to attempt to cover most shading scenarios, yet until the model is tested out in real conditions, that uncertainty will always be present

## **8.2 FINAL RECOMMENDATION**

The final recommendation given to John Deere was that they should not pursue this specific line of research, which involves shading detection using only machine learning methodologies on GNSS data, unless they were willing to pursue one of the two alternatives proposed by the team.

These proposals were set in order to overcome the limits mentioned above. While the first limitation is a little harder to completely solve, correcting the other two is perfectly feasible given the right conditions, and it is the unanimous opinion of the group that both alternatives would make for interesting and fruitful lines of research.

### **8.2.1 POSSIBLE SOLUTIONS TO LIMITATIONS**

#### ***8.2.1.1 Implementing a Larger Training Set***

Firstly, the most important aspect to take into account is John Deere's proprietary database, where GNSS information is stored. As information in the database is not labeled in terms of

shading, in order to continue research, they would either have to classify existing data or collect new and labeled data.

Both of these initiatives are highly ambitious and costly if John Deere were to pursue them but it is nevertheless a possibility, especially the second part, as it is clear John Deere would be able to collect data at an exponentially higher rate than was achieved in this project.

Moreover, a larger dataset opens the door to a Deep Learning approach to the problem, were a model based on something like convolutional neural networks could work very well in the problem at hand. DL architectures are also highly deployable, with vast research on the deployment of these systems in Edge AI devices through frameworks such as ONNX.

#### ***8.2.1.2 Consider Adding a Light Sensor***

The second proposed condition is adding a light sensor to select products, either more high-end editions or more important equipment. A machine learning model that had access to GNSS data, and also was able to take into account changes in lightning would have a much better chance to accurately predict shading than one which simply relied on GNSS data.

This light sensor approach, if deemed accurate, could then, in a more distant future, be used to create a training dataset upon which a GNSS-only approach could also be used. However, this line of research would be conducted in a longer time frame and there is no point in considering it at the moment.

### ***8.3 POSSIBLE FUTURE WORK***

There are two ways in which this section can be interpreted. One is the next steps John Deere as a company can take with the final recommendation. This interpretation is already well explained in the previous section, and therefore not relevant to this one. The second interpretation is the next steps could an individual researcher take in order to build upon this project, which will be the focus of this section.

### **8.3.1 FURTHER DATA COLLECTION**

The first to do would surely be collecting more data, as it has been made clear throughout this report that data is relevant issue to the project and it always helps to have more of it.

Also, there is definitely room for optimization in the data collection process, as currently the board readings are printed to the Arduino IDE standard output, and have to be manually copied and pasted into a text file for analysis. However, this was done not only for commodity's sake, but also to reduce the memory usage of the code, as before the Arduino MEGA was purchased, memory usage was one of the biggest challenges faced, as pointed out in section 5.2.3.

It would also be recommendable to perform another feature engineering round, either from the initial 149 variables or with the final model, with the aim of constructing a more robust model that could include other different variables.

Finally, in the data collection process followed in this project, the group simply walked around or stayed put with the board connected to the Arduino, which in turn was connected to a laptop for data collection. This system proved to be rather uncomfortable and unstable, as small movements in the equipment often resulted in a loose cable and therefore a loss of connectivity.

This is why the group considered implementing a mounted system, which would look similar to the one shown in Figure 33, except all connected to a laptop (it is also possible to do it without the laptop but that was not considered). The figure below shows a static system, but there are also examples where the system can be mounted on a backpack for mobile data collection.



*Figure 33 - Mounted Data Collection System Example*

This of course is slightly outside of this project's scope, as it has little to do with John Deere's objective and would take a considerable amount of time to mount. However, for a smaller project where data collection has to be conducted by the researchers, it would be rather useful.

### **8.3.2 IMPLEMENTATION**

The second possible step would be to deploy the model for real time prediction on an Edge device, such a microcontroller or even a RaspberryPi. This step was initially included in the scope of the project, yet was quickly discarded by the faculty mentor present during the development of the project.

Of course, the implementation would be quite rough compared to what it would look like on John Deere systems, but it would be quite helpful to see the model making predictions in real time (maybe displaying the current prediction). This would have to clear benefits, to point out the flaws of the model, as it would be clear to see where the model makes the

wrong predictions, and to improve the presentation of the project, as an audience would have a better understanding of the system.

### **8.3.3 FURTHER STATISTICAL ANALYSIS**

Even if the machine learning aspect of the project was one of the most important ones, the ever-growing field of statistical inference is incredibly vast, and there are still many ways to optimize and improve the current data analytics setup.

Firstly, to ensure the quality of our data and robustness of the model, certain tests could be added to the model. While most statistical tests are thought for simple regressions, the classification problem at hand can also be tested in different ways, such as McNemar's test for robustness, as seen in (Dietterich, 1998).

Moreover, Cat Boost is just one of many possible algorithms, and although it did prove to be the best performers among the ones tested, it is perfectly possible that it is not the absolute best approach, especially considering the growing deployment capacity of usually heavier models.

Nowadays, usually deep learning architectures such as VGG-16 have been succeeded by ones like MobilenetV1 and are considered strong contenders for any sort of machine learning related project (Busch, Corradi, Ninic, Thermou, & Bennets, 2021). They might not be able to improve upon Cat Boost currently, but in the future, they are to be kept in mind.

## Chapter 9. REFERENCES

- ArcGis Pro 3.1. (s.f.). *ArcGis Pro*. Obtenido de <https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-catboost-works.htm>
- Arduino. (16 de May de 2023). *ATmega2560-Arduino Pin Mapping*. Obtenido de <https://docs.arduino.cc/hacking/hardware/PinMapping2560>
- Busch, J., Corradi, T., Ninic, J., Thermou, G., & Bennets, J. (2021). Deep Neural Networks for Visual Bridge Inspections and Defect Visualisation in Civil Engineering. *ResearchGate*.
- Clark, P. (s.f.). *GitHub*. Obtenido de [https://github.com/sparkfun/SparkFun\\_ublox\\_GNSS\\_Arduino\\_Library](https://github.com/sparkfun/SparkFun_ublox_GNSS_Arduino_Library)
- Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, vol. 10 n. 7, 1895-1923. doi:10.1162/089976698300017197.
- Dunne, M. J. (22 de Mayo de 2018). *Interface Specification IS-GPS-200J*. Obtenido de <https://www.gps.gov/technical/icwg/IS-GPS-200J.pdf>
- ESA. (s.f.). *GNSS Fundamentals*. Obtenido de Navipedia: <https://gssc.esa.int/navipedia/index.php>
- Fotopoulos, G., & Cannon, M. E. (2001). An overview of multi-reference station methods for CM-level positioning. *GPS Solutions*, 1-10.
- Gakstatter, E. (2009). RTK Networks – What, Why, Where? *USSLS/CGSIC Meeting*. Obtenido de <https://www.gps.gov/cgsic/meetings/2009/gakstatter1.pdf>
- Inside GNSS. (2013, November 18). *Inside GNSS*. Retrieved from <https://insidegnss.com/multipath-vs-nlos-signals/>



- Kanhere, A. V., Gupta, S., Shetty, A., & Gao, G. (2021). Improving GNSS Positioning using Neural Network-based Corrections. *Navigation*.
- Kirk, G. R. (2010). *USA Patente n° US20100283674A1*.
- Kowada, S., & Hashimoto, K. (2021). *USA Patent No. US20210124059*.
- Kuratomi, A. (2019). *GNSS Position Error Estimated by Machine Learning Techniques with Environmental Information Input*. KTH Royal Institute of Technology.
- Melvin, R. L. (28 de June de 2021). *The University of Alabama at Birmingham*. Obtenido de <https://sites.uab.edu/periop-datascience/2021/06/28/sample-size-in-machine-learning-and-artificial-intelligence/>
- Rashid, H., & Turuk, K. (2015). Dead reckoning localisation technique for mobile wireless sensor. *IET Digital Archive*.
- Scikit-Learn. (n.d.). *Scikit-Learn*. Retrieved from [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc\\_auc\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html)
- Sever, M., & Alison, M. (2009). Enhanced GNSS signal processing.
- Soffel, M., & Langhans, P. (2012). Celestial Reference System. *Space-Time Reference Systems*, 155-174. doi:10.1007/978-3-642-30226-8\_7
- Soga, H. e. (2013). *Japan Patente n° JP2015068768A*.
- uBlox. (10 de December de 2018). *ANN-MB series Multi-band, high precision GNSS antennas Data Sheet*. Obtenido de [https://cdn.sparkfun.com/assets/1/8/f/f/0/ANN-MB\\_DataSheet\\_\\_UBX-18049862\\_.pdf](https://cdn.sparkfun.com/assets/1/8/f/f/0/ANN-MB_DataSheet__UBX-18049862_.pdf)
- uBlox. (24 de March de 2023). *ZED-F9P-02B Datasheet*. Obtenido de [https://cdn.sparkfun.com/assets/f/8/d/6/d/ZED-F9P-02B\\_DataSheet\\_\\_UBX-21023276.pdf](https://cdn.sparkfun.com/assets/f/8/d/6/d/ZED-F9P-02B_DataSheet__UBX-21023276.pdf)

- Wang, L., Grover, P., & Ziebart, M. (2013). GNSS Shadow Matching: Improving Urban positioning accuracy using a 3D city model with optimized visibility scoring scheme. *Navigation*, 195-207.
- Wanninger, L. (11 de June de 2004). *Introduction to Network RTK*. Obtenido de <http://www.wasoft.de/e/iagwg451/intro/introduction.html>.
- Y. Nishiyama, K. H. (2022, August 9). *Estimating Sunlight Using GNSS Signal Strength from Smartphone*. Retrieved from <https://arxiv.org/pdf/2208.08858.pdf>
- Yandex. (s.f.). *Cat Boost*. Obtenido de <https://catboost.ai/en/docs/concepts/loss-functions-classification>

## **ANNEX I: PROJECT ALIGNMENT WITH SDGs**

This annex provides an analysis of the project aimed at addressing communication disruption between John Deere tractors and satellites caused by shading when a tractor travels beneath an object. The project focuses on developing, training, and testing different models using signal processing techniques from data collected on a Global Navigation Satellite System (GNSS) board to predict shading conditions of raw GNSS data. This annex aims to highlight the project's connection to the Sustainable Development Goals (SDGs).

The project aligns with SDG 9, which focuses the development of resilient infrastructure, promotion of sustainable industrialization, and fostering innovation. By addressing the communication disruption issue in John Deere tractors caused by shading, the project contributes to the improvement of infrastructure reliability and efficiency in the agricultural sector. Figure 34 shows SDG 9.



*Figure 34 - SDG 9*

Efficient communication is vital for modern agricultural practices to ensure optimal productivity and resource management. By mitigating the shading-induced communication disruption, the project directly contributes to enhancing the infrastructure of John Deere tractors. Reliable communication infrastructure enables farmers to make informed decisions,

optimize operations, and reduce resource wastage, thereby supporting sustainable agricultural practices.

Furthermore, the project utilizes signal processing techniques and data analysis to develop and train models capable of predicting shading conditions. This approach demonstrates innovation in the agricultural machinery sector by leveraging advanced technologies to address operational challenges. The innovative solutions proposed in this project can inspire further research and development efforts to enhance communication systems in other agricultural equipment and contribute to technological advancements in the industry.

The project also has a connection to SDG 13, which calls for urgent action to combat climate change and its impacts. While the project's primary goal is to address communication disruption, its indirect impact on reducing environmental impacts in the agricultural sector aligns with SDG 13, which is shown in Figure 35.



*Figure 35 - SDG 13*

Improved communication and accurate positioning systems can optimize tractor routes and reduce fuel consumption. By enabling efficient and optimized operations, the project contributes to reducing greenhouse gas emissions associated with agricultural activities. This aligns with the broader objective of SDG 13 to promote sustainable practices that mitigate climate change.

The project's success in predicting shading conditions and enhancing communication systems helps farmers adapt to climate change impacts. By providing accurate data and facilitating informed decision-making, the project supports climate-resilient agriculture. Farmers can adjust their practices based on real-time weather information, leading to better water management, reduced vulnerability to extreme weather events, and increased agricultural resilience in the face of climate change.

In addition, the project aligns with SDG 2, which aims to end hunger, achieve food security, improve nutrition, and promote sustainable agriculture. By addressing communication disruption between tractors and satellites in the agricultural sector, the project indirectly contributes to SDG 2 (Figure 36) by supporting sustainable agriculture and food production.



*Figure 36 - SDG 2*

The main reason for this comes with improving productivity. The reliable communication established between John Deere tractors and satellites enables farmers to access essential information and data to improve agricultural productivity. By predicting shading conditions and providing accurate positioning, farmers can optimize planting, irrigation, and harvesting practices, resulting in increased crop yields and food production. This contributes to achieving food security and reducing hunger, especially in regions heavily reliant on agriculture for sustenance.

Moreover, the project's focus on sustainable farming practices aligns with SDG 2. Through improved communication systems, farmers can make informed decisions about resource management, such as applying fertilizers and pesticides more efficiently and conserving water. By promoting sustainable agriculture, the project helps protect natural resources, minimize environmental degradation, and support long-term food production capacity.

The project also aligns with SDG 17, pictured in Figure 37, which highlights the importance of global partnerships and collaboration to achieve the Sustainable Development Goals. The project's success relies on cooperation among different stakeholders, including technology developers, agricultural machinery manufacturers, farmers, and satellite service providers.



*Figure 37 - SDG 17*

Firstly, the project showcases the power of partnerships and collaboration to address complex challenges. Collaboration between researchers, technology experts, and agricultural industry stakeholders is crucial for developing innovative solutions and overcoming technical barriers. By engaging different actors and fostering collaboration, the project exemplifies the spirit of SDG 17 and demonstrates how partnerships can drive progress towards sustainable development.

Also, the project's implementation involves the sharing of knowledge, expertise, and best practices among various stakeholders. Collaboration enables the exchange of ideas, lessons learned, and technological advancements. This knowledge sharing process helps build

capacity among farmers and industry professionals, empowering them to adopt improved communication systems and enhance agricultural practices. By promoting knowledge sharing and capacity building, the project contributes to the achievement of SDG 17's objective of strengthening partnerships for sustainable development.

In conclusion, the project addressing communication disruption between John Deere tractors and satellites caused by shading demonstrates a multifaceted connection to the Sustainable Development Goals. Through its alignment with SDG 2 (Zero Hunger) in terms of improving agricultural productivity and promoting sustainable farming practices, SDG 9 (Industry, Innovation, and Infrastructure) through enhancing infrastructure reliability and promoting innovation, and SDG 17 (Partnerships for the Goals) in terms of collaborative approaches and knowledge sharing, the project contributes to the broader agenda of achieving sustainable development.

By enhancing food security, supporting sustainable agriculture, improving infrastructure reliability, fostering innovation, and promoting partnerships, the project showcases the potential for technology-driven solutions to address complex challenges and drive progress towards the SDGs.

The successful implementation of this project can serve as a model for similar initiatives, fostering progress towards the SDGs in the agricultural sector and beyond. It demonstrates the interconnectedness of the SDGs and the importance of adopting holistic approaches to achieve sustainable development and build a more resilient and inclusive future.

## ANNEX II ARDUINO CODE FOR DATA COLLECTION

```
#include <SparkFun_u-blox_GNSS_v3.h>
#include <sfe_bus.h>
#include <u-blox_Class_and_ID.h>
#include <u-blox_GNSS.h>
#include <u-blox_config_keys.h>
#include <u-blox_external_typedefs.h>
#include <u-blox_structs.h>

#include <Wire.h> //Needed for I2C to GNSS

#include <SparkFun_u-blox_GNSS_v3.h>

const int buttonPin = 2;
int buttonState = 0;

SFE_UBLOX_GNSS myGNSS;
void setup()
{
  Serial.begin(115200);
  delay(10000);
  pinMode(buttonPin, INPUT);
  Serial.println();
  Wire.begin(); // Start I2C

  while (myGNSS.begin() == false) //Connect to the u-blox module using Wire port
  {
    Serial.println(F("u-blox GNSS not detected at default I2C address.
Retrying..."));
    delay (1000);
  }

  myGNSS.setI2COutput(COM_TYPE_UBX); //Set the I2C port to output UBX only (turn
off NMEA noise)
  //make header
  Serial.print('PacketCfgSpaceRemaining');
}

void loop()
{
  if (myGNSS.getPVT() == true)
  {
    buttonState = digitalRead(buttonPin);

    if (buttonState == HIGH) {
      // turn LED on:
    }
  }
}
```



```
Serial.print("1"); Serial.print(',');
} else {
  // turn LED off:
  Serial.print("0"); Serial.print(',');
}

Serial.print(myGNSS.getGeometricDOP()); Serial.print(',');
Serial.print(myGNSS.getPositionDOP()); Serial.print(',');
Serial.print(myGNSS.getTimeDOP()); Serial.print(',');
Serial.print(myGNSS.getVerticalDOP()); Serial.print(',');
Serial.print(myGNSS.getHorizontalDOP()); Serial.print(',');
Serial.print(myGNSS.getNorthingDOP()); Serial.print(',');
Serial.print(myGNSS.getEastingDOP()); Serial.print(',');
Serial.print(myGNSS.getFixType()); Serial.print(',');
Serial.print(myGNSS.getGnssFixOk()); Serial.print(',');
Serial.print(myGNSS.getSIV()); Serial.print(',');
Serial.print(myGNSS.getHorizontalAccEst()); Serial.print(',');
Serial.print(myGNSS.getVerticalAccEst()); Serial.print(',');
Serial.print(myGNSS.getNedNorthVel()); Serial.print(',');
Serial.print(myGNSS.getNedEastVel()); Serial.print(',');
Serial.print(myGNSS.getNedDownVel()); Serial.print(',');
Serial.print(myGNSS.getHeading()); Serial.print(',');
Serial.print(myGNSS.getSpeedAccEst()); Serial.print(',');
Serial.print(myGNSS.getHeadingAccEst()); Serial.print(',');
Serial.print(myGNSS.getPDOP()); Serial.print(',');
Serial.print(myGNSS.getPositionAccuracy()); Serial.print(',');
Serial.print(myGNSS.getElipsoid()); Serial.print(',');
Serial.print(myGNSS.getElipsoidHp()); //Serial.print(',');
Serial.println();

}
}
```