IJESRT: 13(10), October, 2024 ISSN: 2277-9655

International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

Chief Editor

Dr. J.B. Helonde

Executive Editor

Mr. Somil Mayur Shah

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

GNSS AMBIGUITY RESOLUTION USING MACHINE LEARNING APPROACH

Yakubu Issaka[1], Kenneth Junior Appau[*2], Philip Agyekum Antwi[3], Francis Ferguson-Howard[4], Ebenezer Ankomah Gyamera[5]
[* 1, 2, 3, 4]Geomatic Engineering Department, University of Mines and Technology, Tarkwa, Ghana
[5]Department of Soil Science, School of Agriculture, University of Cape Coast, Ghana

DOI: 10.5281/zenodo.14001390s

ABSTRACT

In an effort to increase positioning accuracy in difficult circumstances, this study examined the use of machine learning (ML) approaches to improve GNSS ambiguity resolution. Performance indicators, such MSE, RMSE, MAE, and R^2, were used to assess two machine learning models: K-Nearest Neighbours (KNN) and Gradient Boosting Decision Tree (GBDT). GBDT achieved R^2 score of 1.00, the lowest MSE (0.000 meters), RMSE (0.0010 meters), and MAE (0.0007 meters) from the training phase, making it the best performance. With an MSE of 0.000 meters, an RMSE of 0.0049 meters, a MAE of 0.0031 meters, and a R^2 of 1.00, KNN also demonstrated impressive performance. GBDT maintained its exceptional accuracy throughout the testing phase, with an MSE of 0.000 meters, RMSE of 0.0010 meters, MAE of 0.0007 meters, and an R^2 of 1.00. KNN performed competitively, with an MSE of 0.000 meters, RMSE of 0.0050 meters, MAE of 0.0032 meters, and R^2 of 1.00. These comprehensive results demonstrate the usefulness of machine learning approaches, notably GBDT and KNN, in greatly enhancing GNSS ambiguity resolution. Such developments are critical for overcoming the obstacles given by urban canyons, dense vegetation, and other obscuring settings that have previously hampered GNSS location technologies.

KEYWORDS: GNSS, Ambiguity Resolution, Machine Learning (ML); K-Nearest Neighbours (KNN); Gradient Boosting Decision Tree (GBDT)

1. INTRODUCTION

Two kinds of direct measurements are often possible with GPS namely; the carrier phase and the pseudo-range measurements. The carrier phase measurements, because of their low noise level, may be applied to a wide range of high-precision accuracy tasks such as kinematic positioning, static survey, and attitude determination. The carrier phase measurements, however, are unclear due to an unspecified integer cycle count. For high-precision kinematic positioning, this well-known integer ambiguity resolution problem necessitates a laborious initialization procedure [1]. The integer ambiguity can be resolved in several ways. They may be broadly classified into two categories: motion-based ambiguity resolution [2] and search-based ambiguity resolution [3]. Motion-based techniques must gather data for a duration that includes discernible shifts in the visible GPS constellation or an apparent rotation of the platform.

Motion-based GNSS ambiguity resolution techniques leverage the dynamics of the receiver to enhance the resolution process. These techniques use the receiver's motion information to resolve carrier phase ambiguities. Examples of such methods include Kalman Filter-Based Dynamic Model [5]. Inertial Aided Ambiguity Resolution [6], Motion Constraints (Zero-Velocity Updates) [7], Dynamic Network RTK [8], Carrier Phase-Based Relative Positioning with Motion Constraints [9], Dynamic Model-Based Smoothing [10], Multi-Sensor Fusion [1]1, Time-Differenced Carrier Phase (TDCP) Method [12].

The search-based approaches merely employ observations from a single epoch to determine the most likely combination of ambiguities, albeit occasionally the amount of noise may cause it to be incorrect. Examples of these methods include the LAMBDA Method (Least-squares Ambiguity Decorrelation Adjustment) [13], MLAMBDA (Modified Lambda) [14], C-LAMBDA (Integer Bootstrapping) [15], Integer Least-Squares (ILS) Method [16], Sequential Conditional Least-Squares Ambiguity Resolution (SCLAR) [17], Integer Gauss-Markov (IGM) Method [18], GNSS Ambiguity Search by Filtering (GASF) [19], Ambiguity Function Method (AFM)

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[20]. The methods discussed earlier are traditional or classical. Remarkably, this study does not explore the traditional integer search or motion-based technique; numerous papers have thoroughly discussed this strategy, including [5, 6, 7, 8, 13, 14, 15, 18, 19, 20].

A relatively new area of research called "machine learning" (ML) uses statistical algorithms to learn from and analyse data to forecast or make decisions based on it. In this study, the integer in carrier phase measurements—which is necessary for precise positioning —can be found using machine learning (ML) techniques using GNSS data. This is so that the ML algorithm may understand the link between the GNSS measurements and the integer ambiguity by training or calibrating it using input data, such as simulated or experimental data. Once trained, the method may be used to find the real-time integer ambiguity [21].

The Integer Ambiguity Resolution problem has not been fully tackled using existing machine-learning techniques. The reason for such might be that most machine-learning approaches need a lot of data and a lot of processing power to train their algorithms. Nonetheless, a substantial amount of research supports the application of Machine Learning for both terrestrial and aerial navigation that employ satellite data. Neural networks, for example, have been used in [22] to assess geoid unevenness using GPS data. A particular kind of recurrent neural network called an LSTM (Long Short-Term Memory) network has been used to detect abnormalities in satellite data [23] and to rectify GPS signals [24]. By using data from Inertial Navigation Systems (INS) to correct for GPS signal losses.

Therefore, this research aims to estimate ambiguity integer resolution using machine learning methods (k-Nearest Neighbours (KNN) and Gradient Boosting Decision Tree (GBDT)). Here, estimated integer ambiguity resolution values determined by a classical technique; Least-squares Ambiguity Decorrelation Adjustment (LAMBDA), is chosen to train the machine learning methods to estimate integer values. LAMBDA is selected because it is a well-known and often applied technique in the field of Global Navigation Satellite System (GNSS) ambiguity resolution. The resilience and effectiveness of this approach in resolving integer ambiguities, which is essential for obtaining high-precision GNSS location, have drawn special attention.

2. MATERIALS AND METHODS

Data Structure

The dataset comprises of approximately 1000 observation epochs collected from multiple GNSS receivers and satellites. Each epoch includes raw GNSS observations. The data was obtained through extensive field measurements using high-precision GNSS receivers, ensuring a diverse and comprehensive dataset for training and evaluating the machine learning models.

The dataset is labelled with known integer ambiguities, enabling supervised learning methods to effectively learn and predict integer ambiguities. In training the model, 70% of the data was used and for training the model, and the remaining 30% for testing.

Machine Learning Methods

Machine Learning (ML), in general, is the study and development of intelligent agents, where an intelligent agent is a system that examines its environment and takes actions to maximise its chances of success. Many real-world problems need the agent to work with incomplete or confusing facts. Using ML techniques has many advantages over traditional development and implementation strategies.

These include quick access to collected knowledge (e.g. knowledge-based systems), easy-to-implement prototypes without deep expert knowledge (e.g., artificial neural networks (ANNs)), or systems that can learn (e.g., evolutionary optimisation algorithms).

k-Nearest Neighbour (KNN)

The KNN algorithm views the instances as points in a metric space and the descriptive qualities as its dimensions. The categorized instances are maintained in the training phase unprocessed. A new example is classified by calculating the distance (along the descriptive qualities) between it and all training examples, then assigning the new example to the class of the closest training example [25].

It is also possible to use the KNN technique for regression in addition to classification. It is actually applicable to any kind of output. If the examples only have continuous attributes, they can be viewed as points in a Euclidean space, and the Euclidean distance measure can be applied. Given two examples x = (x_1..., xn) and y = (y_1...yn), their Euclidean distance is calculated in Equation (i).

$$distance(\mathrm{x},\mathrm{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad \text{(i)}$$

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

It should be noted that this ignores scale variations across characteristics and does not support discrete attributes. Equation (ii) computes the distance using a broader definition.

$$distance(x,y) = \sqrt{\sum_{i=1}^{n} w_i \times (x_i - y_i)^2} \quad \text{(ii)}$$

where w_i is a non-negative weight value assigned to attribute Ai and the difference between attribute values is defined in Equation (iii) as follows;

$$distance(x,y) = \begin{cases} |x_i - y_i| & \text{if attribute } A_i \text{ is continues} \\ 0 & \text{if attribute } A_i \text{ discrete and } x_i = y_i \\ 1 & \text{otherwise} \end{cases} \quad \text{(iii)}$$

After normalizing continuous qualities, the weights enable the varying significance of the traits to the job at hand to be taken into consideration. The KNN approach is more accurate and resilient when combined with the more generic KNN method. When estimating the target value for a new example, it uses the k closest training instances together with their target values. By majority voting, the new example's class is decided upon for categorization. According to Pugelj and Džeroski [25], the prediction in regression is the mean of the class values for the k training instances.

Gradient Boosting Decision Tree (GBDT)

Boosting algorithms combine weak learners, i.e., learners slightly better than random, into a strong learner in an iterative way [26]. Gradient boosting is a boosting-like algorithm for regression. Given a training dataset, the goal of gradient boosting is to find an approximation of the function which maps instances to their output values by minimizing the expected value of a given loss function. Gradient boosting builds an additive approximation of function as a weighted sum of functions which is expressed in Equation (iv)

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x) \quad \text{(iv)}$$

Where ρ_m is the weight of the m^{th} function, $h_m(x)$. These functions are the models of the ensemble (e.g., decision trees). The approximation is constructed iteratively. First, a constant approximation of $F^*(x)$ is obtained as expressed in Equation (v).

$$F_0(x) = \arg min \sum_{i=1}^{N} L(y_i, \alpha) \quad \text{(v)}$$

Where, $L(y_i, \alpha)$ is the loss function. Subsequent models are expected to minimize as presented in Equation (vi).

$$\left(\rho_m, h_m(x)\right) \arg min \sum_{i=1}^{N} L\left(y_i, F_{m-1}(x_i) + \rho h(x_i)\right) \quad \text{(vi)}$$

Model Evaluation

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. This is important because it provides the opportunity to assess the efficacy of a model. In this study, the following statistical evaluators were used for the model performance evaluation:

R-squared (R^2)

R-squared is a measure of how well the predicted values of the model fit the actual data. It represents the proportion of variance in the target variable that the model can explain. R^2 ranges from 0 to 1, with higher values indicating a better fit. An R^2 value close to 1 indicates that the model explains a large portion of the variance in the data, while a value close to 0 indicates a poor fit. The mathematical equation for finding the coefficient of determination (R^2) is expressed in Equation (vii).

$$R^2 = 1 - \frac{RSS}{TSS} \quad \text{(vii)}$$

where, RSS is the sum of squared residuals (the sum of the squared differences between the predicted and actual values), and TSS is the total sum of squares (the sum of the squared differences between the actual values and the mean of the actual values).

Root Mean Squared Error (RMSE)

RMSE is a measure of the average error between the predicted and actual values. It is calculated by taking the square root of the mean of the squared differences between the predicted and actual values. RMSE is commonly used to evaluate the accuracy of a model's predictions, with lower values indicating better accuracy. The mathematical formula is expressed in Equation (viii) as

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}} \quad \text{(viii)}$$

where N is the number of samples, $\hat{x}_i$ is the predicted value, and x_i is the actual value.

Mean Absolute Error (MAE)

As expressed in Equation (ix), MAE measures the average absolute difference between the predicted and actual values. It is calculated by taking the mean of the absolute differences between the predicted and actual values. MAE is another commonly used indicator of prediction accuracy, with lower values indicating better accuracy.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \quad \text{(ix)}$$

where, y_i is the predicted value, x_i is the true value, and n is the number of data points.

3. RESULTS AND DISCUSSION

In this study, two (2) machine learning models (K-Nearest Neighbour (KNN) and Gradient Boosting Decision Tree (GBDT) were used to predict the integer ambiguity values. MATLAB R2024a was used in writing scripts for the models' development and implementation. GBDT was scripted in Python and implemented MATLAB. Before training the models, it was necessary to normalize data to avoid reduction in network speed and accuracy. In this present work, the GNSS data were normalized between -1 and 1.

Models Developed

In this section, we discuss the parameters utilized in our K-Nearest neighbours (KNN) model, as outlined in Table 1, Additionally, a detailed overview of the Gradient Boosting parameters used in the study, as specified in Table 2. These parameters significantly influence the performance and behaviour of the models.

KNN Model Parameters

The KNN model used in our work considers the five nearest data points in the feature space while generating predictions, with Number of Neighbours (k) = 5. The selection of k achieves a balance between variation and bias. While a bigger k could unduly smooth out the decision boundaries (high bias), thus missing significant patterns, a lesser k might lead to a model that is excessively sensitive to noise (high variance). Based on empirical research, we found that selecting k = 5 yielded the best results on our dataset, balancing sensitivity to the local structure of the data with generalization. The distance between data points was calculated using the Euclidean distance metric.

The Euclidean metric is a suitable choice for our data as it accurately captures the geometric separation between points in a continuous feature space. When features have comparable scales, this measure makes it easier to determine who the closest neighbours are. We used a weighting technique based on distance in our KNN model. With this method, neighbours are given weights depending on their inverse distance, with closer neighbours being given more weight. In noisy datasets where far neighbours may be less significant or even deceptive, this weighting strategy can improve the effect of closer neighbours on the prediction, which can result in more robust and accurate predictions.

Table 1. K-Nearest Neighbours Parameters

Model	Parameter	Value
KNN	Number of Neighbours	5
	Metric	Euclidean
	Weight	Distance
	Cross Validation	5-fold

Gradient Boosting Decision Tree Parameters

We used Extreme Gradient Boosting (XGBoost), which is a well-known gradient boosting method that is renowned for its scalability and speed. By utilizing ten trees, the model was better able to represent intricate patterns. With a learning rate of 1, every tree's contribution to the final model is regulated. Though careful adjustment is needed to avoid overfitting, a greater learning rate might result in faster convergence. Because the model training is repeatable, the outcomes are reliable and consistent. By limiting the complexity of the model, the maximum depth of each trees was set at 5. This helps prevent overfitting.

Regularization was achieved using a LAMBDA value of one, which helped to control overfitting by penalizing more complicated models. The proportion of training instances utilized is one, which means that all

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

training data is used for each iteration, resulting in maximal data usage for training. Furthermore, the fractions of features for each split, level, and tree are set to one, implying that all characteristics are examined for each split, level, and tree respectively. This technique guarantees that the model considers all accessible characteristics, hence encouraging complete learning.

Table 2. Gradient Boosting Decision Tree Parameters

Model Parameters	Value
Method	Extreme Gradient Boosting (XGBoost)
Number of trees	10
Learning rate	1
Replicable training	Yes
Limit depth of individual trees	5
Regularization strength	Lambda: 1
Fraction of training Instances	1
Fraction of features for each split	1
Fraction of features for each level	1
Fraction of features for each tree	1
Cross Validation	5-fold

Performance of the Models Developed

Table 4 compares the performance metrics of the machine learning (ML) models for GNSS ambiguity resolution during the training phase. Metrics include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2).

Fig. 1 and Fig. 2 show the trend of ambiguity integer values estimated by both prediction models as compared to the Target using the training and testing data respectively. Fig. 3 and Fig. 4 also show the trend of errors generated by both prediction models using the training and testing data prediction models as compared to the Target respectively.

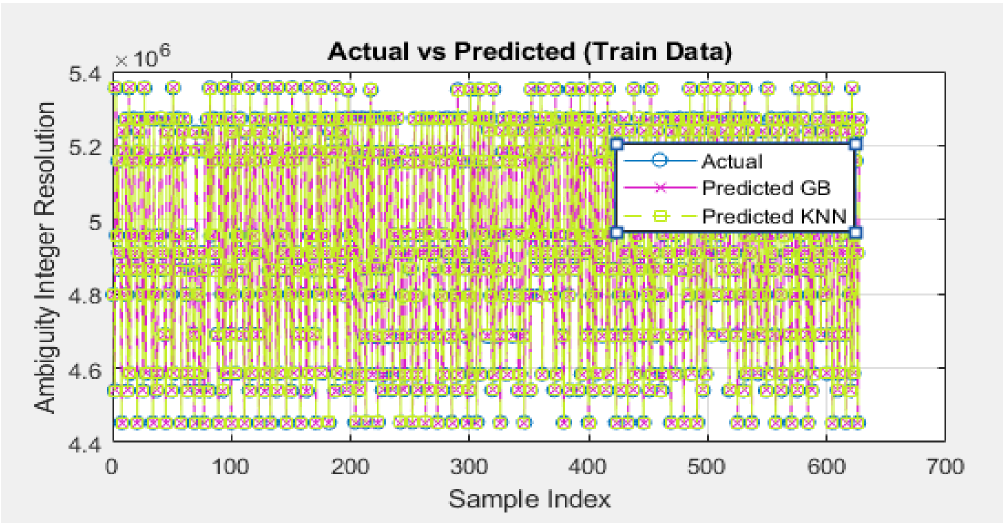

Fig. 1 Actual verse Predicted Integer Ambiguity (Train)

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

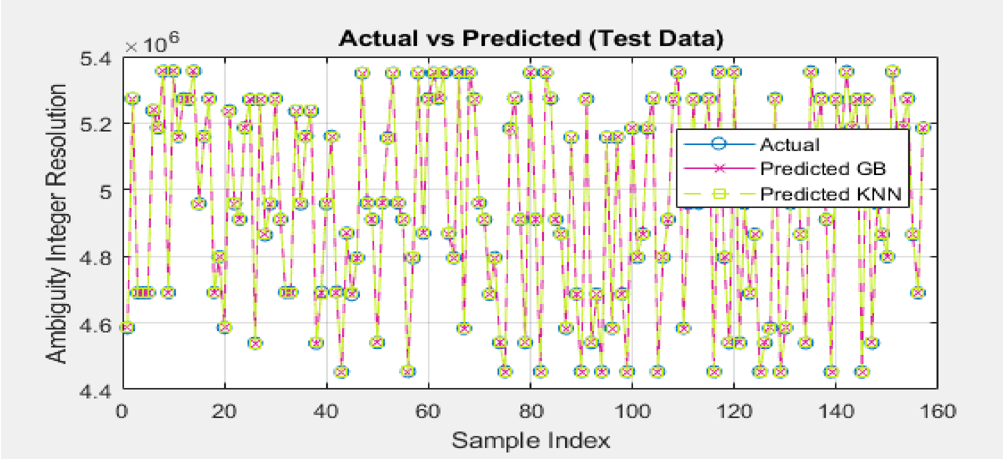

Fig. 2 Actual verse Predicted Integer Ambiguity (Test)

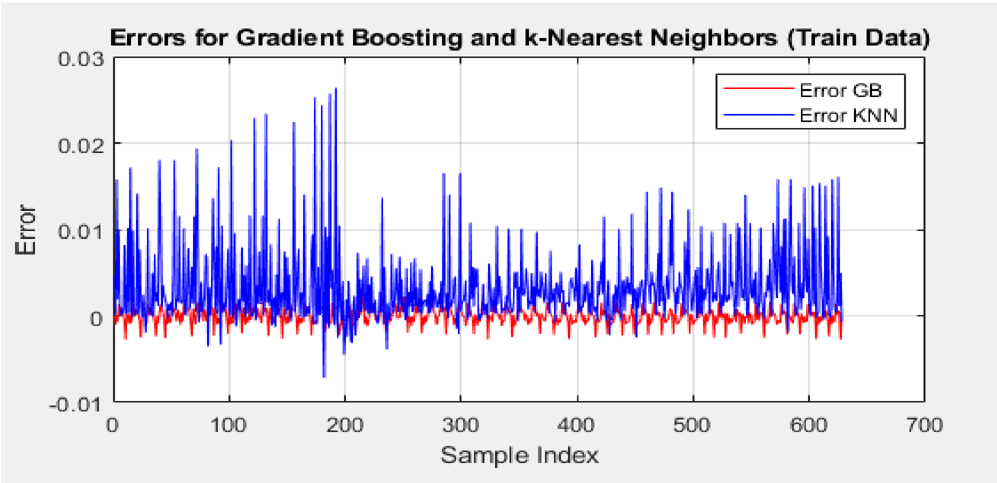

Fig. 3 Error Propagated during Prediction (Train)

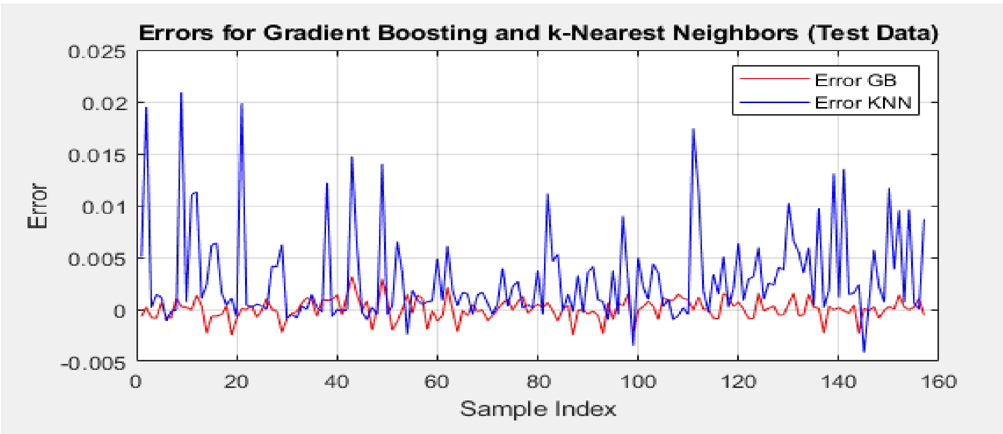

Fig. 4 Error Propagated during Prediction (Test)

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

Table 4 Model Evaluation Comparison for Training

Model	**MSE (m)**	**RMSE (m)**	**MAE (m)**	**R^2**
KNN	0.000	0.0049	0.0031	1.00
GBDT	0.000	0.0010	0.0007	1.00

GBDT exhibits the lowest error metrics among the models, with MSE, RMSE and MAE values of 0.000 meters, 0.0010 meters, and 0.0007 meters respectively. KNN also performs well with slightly higher comparably but still acceptable errors (MSE: 0.000 meters, RMSE: 0.0049 meters, MAE: 0.0031 meters).

All the models achieved a perfect R^2 score of 1.00, indicating excellent fitting to the training data and confirming their ability to predict GNSS ambiguity resolution with high accuracy. Table 5 presents the evaluation results of the same ML models during the testing phase, providing insights into their generalization and performance on unseen data.

Table 5 Model Evaluation Comparison for Testing

Model	**MSE (m)**	**RMSE (m)**	**MAE (m)**	**R^2**
KNN	0.0000	0.0050	0.0032	1.00
GBDT	0.0000	0.0010	0.0007	1.00

Gradient Boosting Decision Tree continued to demonstrate superior performance during the testing phase, maintaining the lowest errors (MSE: 0.000 meters, RMSE: 0.0010 meters, MAE: 0.0007 meters). KNN also performs well, albeit with slightly increased errors compared to the training phase (MSE: 0.0000 meters, RMSE: 0.0050 meters, MAE: 0.0032 meters). The models maintained high R^2 values, indicating robust predictive performance on the testing data.

4. CONCLUSION

Performance Analysis

From Table 4, it is evident that GBDT outperformed k-Nearest Neighbours in terms of accuracy metrics such as MSE, RMSE, and MAE during the training phase. Gradient Boosting Decision Tree, in particular, demonstrates exceptional precision with minimal prediction errors across all evaluated metrics, indicating its robust capability in resolving GNSS ambiguities with high fidelity. k-Nearest Neighbours also performed admirably, showing slight increases in error metrics compared to GBDT but maintaining competitive accuracy and achieving perfect R^2 score, signifying excellent model fitting. Moving to Table 5, which evaluates model performance on unseen testing data, GBDT again emerged as the standout performer with the lowest MSE, RMSE, and MAE values. This reaffirms GBDT's reliability and consistency in predicting GNSS positions accurately under varied conditions. k-Nearest Neighbours continued to demonstrate strong predictive power, albeit with slightly higher errors compared to the training phase, underscoring its capability in real-world applications.

Implications for GNSS Applications

The implications of these results are profound for the advancement of GNSS technology. ML-based approaches, particularly KNN and Gradient Boosting Decision Tree, offer substantial improvements in accuracy and reliability over traditional methods. Such advancements are critical for enhancing navigation systems' performance in challenging environments such as urban areas with high-rise buildings or regions with dense vegetation, where conventional GNSS techniques often struggle due to signal blockages or multipath effects.

In conclusion, the evaluated ML models present compelling evidence of their efficacy in advancing GNSS ambiguity resolution. KNN and Gradient Boosting Decision Tree stand out as reliable choices for achieving precise and reliable position estimates, offering substantial improvements over traditional methods. These findings underscore the transformative potential of ML in overcoming longstanding challenges in satellite navigation, paving the way for more resilient and accurate positioning technologies in both civilian and military applications.

5. ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to the University of Mines and Technology, particularly the Geomatic Engineering Department, for their invaluable support throughout our research. We are especially

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

thankful for the provision of the GPS instruments and their associated equipment, which were essential for our data collection efforts. Additionally, we greatly appreciate access to the Geomatic Engineering Laboratory, which facilitated the processing of our data. The resources and expertise provided by the department have been instrumental in the successful completion of our project. Thank you for your unwavering support and encouragement.

REFERENCES

[1] S. Fan, K. Zhang and F. Wu, "Ambiguity Resolution in GPS-based, Low-cost Attitude Determination", Journal of Global Positioning Systems, vol.2, no.4, pp. 207–214, 2005.
[2] J. L. Crassidis, F. L. Markley, and E. G, "Global Positioning System Integer Ambiguity Resolution without attitude knowledge," Journal of Guidance Control and Dynamics, vol. 22, no. 2, pp. 212-218, 1999
[3] P. G. Quinn, "Instantaneous GPS Attitude Determination", In Proceedings of ION GPS-93, 603-615, Alexandria, VA: Institute of Navigation, 1993.
[4] E. Sutton, "Integer Cycle Ambiguity Resolution under Conditions of Low Satellites Visibility", ION GPS, pp. 91-98, 2002.
[5] B. W. Parkinson and J. J. Spilker, "Global Positioning System: Theory and Applications", Washington, D.C AIAA, vol. 2, 1996.
[6] C. Hide, T. Moore and M. Smith, "Adaptive Kalman filtering for low-cost INS/GPS", Journal of Navigation, vol. 1, no.56, pp. 117-128, 2003.
[7] E. H. Shin, and N. El-Sheimy "A new calibration method for strapdown inertial navigation systems: Measurement Science and Technology", Vol. 11, No. 13, pp. 1714-1722. 2002.
[8] G. Wübbena, A. Bagge and M. Schmitz, "RTK networks based on Geo++® GNSMART–concepts, implementation, results", In Proceedings of the 14th International Technical Meeting of the Satellite, Division of The Institute of Navigation (ION GPS 2001), Salt Lake City, Utah, pp. 368-378, 2001.
[9] Y. Feng and J. Wang, "GPS RTK performance characteristics and analysis", Journal of Global Positioning Systems, Vol. 1, No.7, pp. 1-8, 2008.
[10] M. Ge, G. Gendt, M. Rothacher, C. Shi and J. Liu, "Resolution of GPS carrier-phase ambiguities in precise point positioning (PPP) with daily observations", Journal of Geodesy, Vol. 12, No. 80, pp. 719-732, 2006.
[11] S. Godha and M. E. Cannon, "Integration of DGPS with a low-cost MEMS–based inertial measurement unit (IMU) for land vehicle navigation application", In Proceedings of the 20th International Technical Meeting of the Satellite, Division of The Institute of Navigation (ION GNSS 2007), Fort Worth, Texas, pp. 333-345, 2007.
[12] X. Zhang and P. Li, "Time-differenced carrier phase positioning based on a single dual-frequency GPS receiver", GPS Solutions, Vol. 1, No. 16, pp. 103-113, 2012.
[13] P. J. G. Teunissen, "The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation", Journal of Geodesy, Vol. 2, No. 70, pp. 65-82, 1995.
[14] X. W. Chang, X. Yang, and T. Zhou, "MLAMBDA: A modified LAMBDA method for integer least-squares estimation", Journal of Geodesy, Vol. 9, No. 79, pp. 552-565, 2005.
[15] B. Li and P. J. G. Teunissen, "A new class of GNSS ambiguity resolution approaches based on integer bootstrapping", Journal of Geodesy, Vol. 2, No. 86, pp. 79-93, 2012.
[16] P. J. G. Teunissen, "Least squares estimation of the integer GPS ambiguities", In Invited Lecture, Section IV Theory and Methodology, IAG General Meeting, Beijing, pp. 1-17, 1993.
[17] P. Xu, J Guo and C. Shi, "Fast GPS ambiguity resolution by the method of sequential conditional least-squares adjustment", Journal of Geodesy, Vol. 12, No. 78, pp. 749-756, 2005.
[18] B. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to GPS", IEEE Transactions on Signal Processing, Vol. 11, No. 46, pp. 2938-2952, 1998.
[19] S. Jazaeri and P. J. G. Teunissen, "GNSS ambiguity resolution with filtered real-valued ambiguities", GPS Solutions, Vol. 3, No. 21, pp. 1113-1126, 2017.
[20] C. C. Counselman and S. A. Gourevitch, "Miniature interferometer terminals for earth surveying: ambiguity and multipath with the global positioning system", IEEE Transactions on Geoscience and Remote Sensing, GE, Vol. 4, No. 19, pp. 244 -252, 1981.
[21] de C. Raul and C. Luis, "Neural network-based controller for terminal guidance applied in short-range rockets", IEEE Aerospace and Electronic Systems Magazine, No. 1, pp. 1–11, 2023.

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[22] M. H S. Kavzoglu, “Modelling local GPS and levelling geoid undulations using artificial neural networks”, Journal of geodesy, No. 78, pp. 520–527, 2005.
[23] Z. Zefan, J. Guang, X. Chi, C. Siya, Z. Zhelong and Z. Lu, “Satellite telemetry data anomaly detection using causal network and feature-attentionbased lstm”, IEEE Transactions on Instrumentation and Measurement, No.71, pp. 1–21, 2022.
[24] M. R. Mosavi and F. Shafiee, “Narrowband interference suppression for GPS navigation using neural networks”, GPS solutions, No. 20, pp. 341–351, 2016.
[25] M. Pugelj and S. Džeroski, Predicting Structured Outputs k -Nearest Neighbours Method. Verlag Berlin Heidelberg: T. Elomaa, J. Hollm´en, and H. Mannila (Eds.), pp.262–276, 2011.
[26] C. Bentéjac, A. Csörgő and G. Martínez-Muñoz, “A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review”, 54. 2020.

IJESRT is licensed under a Creative Commons Attribution 4.0 International License.