IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 May 2024, accepted 1 August 2024, date of publication 8 August 2024, date of current version 27 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3440314

RESEARCH ARTICLE

Improving EKF-Based IMU/GNSS Fusion Using Machine Learning for IMU Denoising

ROHAN KUMAR REDDY DAMAGATLA AND MOHAMED ATIA, (Senior Member, IEEE)
Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Corresponding author: Rohan Kumar Reddy Damagatla (rohankumarreddydama@cmail.carleton.ca)

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant RGPIN-2017-06261.

ABSTRACT In the realm of navigation systems, Inertial Measurement Unit (IMU) sensors play a pivotal role. The advent of Micro-Electro-Mechanical System (MEMS) sensors has introduced a lightweight and cost-effective alternative for IMUs. However, MEMS IMUs come with the challenge of larger stochastic errors that accumulate over time, resulting in navigation drifts. To address this issue, the conventional approach involves fusing IMU with the Global Navigation Satellite System (GNSS) for reliable navigation. Nevertheless, this fusion setup fails in providing ubiquitous navigation during GNSS outage scenarios due to persistent IMU errors. In this paper, an efficient methodology is developed to mitigate navigation drifts by eliminating IMU errors using Light Gradient Boosting Machine (LightGBM) and Categorical Boosting (CatBoost) Machine Learning (ML) algorithms. In contrast to existing methodologies that employ high-end and expensive IMUs for training models to denoise low-cost MEMS IMUs, this paper proposes utilizing Inverse Kinematics (IK). This approach helps to derive clean IMU training data from the Position, Velocity, Attitude (PVA) values estimated through the Extended Kalman Filter (EKF) when GNSS is available and reliable. The distinctive advantage of the IK approach lies in its capacity to obtain real-time pseudo error-free IMU data without the necessity for high-end IMUs to train ML models. The proposed method undergoes testing in both Loosely coupled and Tightly coupled EKF scenarios using simulation and real dataset under varying GNSS outage durations. Comparisons are made between the denoised IMU signals and signal processing techniques such as Moving Average (MA) and Savitzky Golay (SG). Additionally, we present a comparative analysis of the proposed algorithms against Convolutional Neural Networks (CNN). Results demonstrate a noteworthy enhancement in position, velocity, and orientation estimation. Furthermore, the computation time required for model training and prediction across various algorithms is analyzed. The outcomes prove the superiority of the proposed tree-based algorithms over the conventional filtering methods and CNN in denoising IMU and improving the navigation results over 90% in all the states compared to the PVA values obtained using raw IMU.

INDEX TERMS CatBoost, convolution neural networks, extended Kalman filter, GNSS, IMU, inverse kinematics, LightGBM.

I. INTRODUCTION

Position, Velocity and Attitude (PVA) states of a vehicle can be obtained using IMU sensors, which incorporate gyroscopes and accelerometers, through IMU mechanization equations [1]. The system using IMU to provide navigation is called Inertial Navigation System (INS). The INS is accurate in short term but becomes ineffective when used as a stand-alone navigation system due to the accumulation of IMU errors [2]. The majority of errors suffered by commercial MEMS IMU used in navigation are due to bias, scalefactor, and random noises [3]. Conversely, GNSS measurements offer reliable navigation for more extended periods but are noisy. This noisy nature of GNSS is considered as a drawback in applications that require precise navigation solutions [4]. This raised the necessity for a fusion mechanism capable of integrating IMU and GNSS values to deliver an accurate and enduring navigation solution.

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken.

Numerous researchers have devised sensor fusion methods, with the EKF emerging as one of the most widely utilized, dominating alternative approaches. The EKF leverages IMU and GNSS data to mitigate drift by eliminating IMU noises and correcting PVA values. However, a drawback of the EKF is the drift in navigation during GNSS outages. When the GNSS outage occurs, the fusion system relies solely on INS to obtain the navigation information. The INS depends on the quality of the IMU signals to provide accurate navigation for longer periods and this raises the need for denoising of IMU signals [5]. Most of the researchers focused on using traditional signal processing denoising methods to remove the IMU errors and with the recent advancements in ML, there have been several works focusing on using Deep Learning (DL) techniques like CNN, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU) for denoising IMU signals. These methods have shown significant improvements in reducing IMU errors. The main drawback of these works is the usage of high-precision navigation grade IMU sensors to gather the training data. Navigation grade IMU sensors have low error specifications and provide stable and accurate navigation even during GNSS outages. This approach of using navigation grade IMU sensors is expensive and not ideal for low-cost systems. Also, most of the published works have not discussed the computation power required for the ML models and their feasibility in real-time scenarios. So in this paper, we propose an efficient method to remove the IMU errors and improve the navigation solution during the GNSS outages mainly in urban areas where the outage duration is less than 1 minute. The main contribution of this work is in the utilization of Inverse Kinematics to facilitate the ML model training.

The rest of the paper is organized into the following sections: Section II covers the related works. Section III provides the necessary background details implemented in this paper. Section IV illustrates the proposed methodology. Section V covers the experimentation and results. Finally, Section VI concludes this paper.

II. RELATED WORKS

During the GNSS outages, the output of the fusion mechanism starts to drift rapidly. This problem is tackled by different approaches. Few researchers have focused on predicting the Kalman Filter estimates to correct the system states, few works focused on estimating the position errors, few approaches focused on estimating the GNSS position increments to reduce the error drift in outage scenarios. Wen Ye et al. proposed Noisy Input Gaussian Process Regression (NIGPR) to learn the INS error models when GPS is available and bridge the GPS outages by predicting the INS observation measurement to feed to the Kalman Filter [7]. Zhao et al. and Liu and Guo proposed deep learning algorithms CNN-GRU and LSTM to estimate the GNSS position increments during the GNSS outages and provide continuous error updates in the fusion mechanism [8], [9]. Li et al. have proposed a LightGBM algorithm to predict the position difference during the GNSS outages [10]. In this paper, the focus is on removal of the noises from IMU and using INS to provide reliable navigation during GNSS outages.

Conventional signal processing filter methods are the fundamental approaches for denoising signals. Gonzalez and Catania proposed a Moving Average (MA) filter to denoise the IMU signals [11]. The input noisy signal is smoothened using MA filter by averaging the signal values over a specific time window. The MA filter is one of the simplest methods and is often used as an initial approach for removing white noises in a signal. The output of a MA filter for a signal input x[n] is

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \tag{1}$$

where N is the window size and the only parameter to be determined. The main disadvantage of MA filter is the lack of methods to estimate the N value. In this work, the authors tackled this issue by varying the N value with constant step size and the resultant IMU signal is compared with high-end noise-free IMU signal and the optimal N value is obtained. The MA filter is modified by adapting weights, known as Auto-Regressive Moving Average (ARMA). Waegli et al., Tu and Peng, and Yuan et al. used ARMA for denoising of IMU signals [12], [13], [14]. The ARMA model assumes that a current value of a signal at time t can be represented as the combination of weighted sum of past values and error terms [15]. Gan et al. and Liu et al. applied the Empirical Mode Decomposition (EMD) method [16], [17]. EMD decomposes the signal into multiple simple functions called Intrinsic Mode Functions (IMF), effectively separating the noise from the original signal. Karaim et al. proposed Savitzky Golay (SG) filter to denoise IMUs [18], [19]. This filter is a signal smoothing and noise reduction technique that focuses on a local subset of data points within a moving window to fit a polynomial curve. The polynomial p(x) fitted on the signal y[x] over the window of length 2m+1 where m is the length of the window, k is the degree of polynomial and is usually an odd number can be represented as

$$p(x) = \sum_{i=0}^{k} a_i x^i \tag{2}$$

where a_i are the polynomial coefficients. The optimal polynomial is obtained by using the least squares method on the cost function given as

$$e = \sum_{-m}^{m} (p(x) - y[x])^2 \tag{3}$$

Kang et al. and Kang et al. performed wavelet transform method on IMU signals which splits the signals into multiple wavelet functions with different frequencies and amplitudes [20], [21]. After the decomposition, the wavelet functions at higher frequencies are mostly noise components, and a threshold value is selected to remove the noise, and the resultant signal is reconstructed.

In recent years there have been several advancements in the field of ML and its applications in real time. This has prompted increased research into employing data models and a variety of ML techniques to enhance the quality of IMU data by reducing unwanted noise. The data-driven models outperformed the conventional signal processing techniques, despite the fact that the former are known to help reduce noise. Gonzalez and Catania proposed Multi Linear Regression (MLR) algorithm and used navigation grade IMU to obtain the training data [22]. The Root Mean Square Error (RMSE) values of the results obtained are compared with those obtained through MA filter and MLR algorithm outperformed. Brossard et al. introduced a Convolutional Neural Network (CNN) with dilation layers for denoising IMU gyroscope signals [23]. The efficiency of CNN is proved by comparing the estimated orientation with that of the ground truth values. However, a limitation of this work is the necessity for ground truth values during training. Han et al. also proposed DUET, an online deep IMU calibration approach using dilated CNN network to improve the inertial-based odometry [24]. Liu et al. proposed a learning method called Gyro-Net to estimate the random errors in gyroscopes and improve the orientation [25]. The drawback of this work is the emphasis is primarily on noise removal in gyroscopes and enhancing orientation. Gao et al. proposed CNN architecture to reduce the IMU errors and improve inertial navigation [26]. Abolfazli Esfahani et al. proposed OriNet and AbolDeepIO, LSTM-based learning architectures to improve the orientation and odometry of the vehicle respectively [27], [28]. Abolfazli Esfahani et al. proposed OdoNet, a CNN-based-pseudo-odometer learning model to improve the accuracy of IMU/GNSS fusion during GNSS outages [29]. Tang et al. proposed DeepOdo, combining CNN and GRU networks to estimate velocity of the vehicle using IMU and barometer values in smartphones during GNSS outages [30]. Wang et al. proposed IDOL, an LSTM-based learning model to estimate orientation and position using smartphones [31]. The IMU signals are treated as time-series data and various types of RNN algorithms are utilized. LSTM networks, GRU and all the combinations of LSTM and GRU are proposed by Sun et al., Zhu et al., and Jiang at al. to remove noise in gyroscopes [32], [33], [34]. The orientation values are calculated after denoising, and the RMSE values are reported. The drawback of these works is that the focus is only on gyroscopes and the experiments were carried out in a stationary environment where the IMU is not moving. K-Nearest Neighbours (KNN), a basic ML algorithm is proposed by Engelsman,D to denoise the accelerometer values [35]. This work uses high-end IMU data for training the KNN algorithm and the results obtained are compared to that of RNN, LSTM, GRU, MA, SG methods and KNN outperformed all the methods which proves the effectiveness of classical ML algorithms which require less training resources and less hyperparameter tuning over DL algorithms. A Machine Learning based Adaptive Neuro-Fuzzy Inference System (ML-ANFIS) is introduced by Mahdi et al. [36]. This system is designed to denoise IMUs, enabling their use as a standalone navigation system, with subsequent comparisons of Position, Velocity, and Attitude (PVA) values. Noteworthy drawbacks in these approaches include a predominant focus on denoising gyroscopes alone and the reliance on navigation-grade sensor values for training the model, which is not only expensive but also suboptimal for the denoising of IMUs. This paper aims to fill existing research gaps, and the contributions are outlined as follows: (i). In instances where GNSS is accessible, an EKF is employed to compute dependable PVA values. By utilizing Inverse Kinematics (IK), approximate pseudo noise-free IMU values are obtained, which prove valuable for training purposes. (ii). The study introduces LightGBM and CatBoost, both being types of boosting Machine Learning Algorithms, specifically designed to address and mitigate noise in IMU data.

This paper is an extension of the work presented at IEEE Sensors Application Symposium 2023 conference proceedings [6].

III. BACKGROUND

A. KALMAN FILTER AND EXTENDED KALMAN FILTER

Kalman Filter (KF) can be applied to any linear system to predict the future states and improve the noisy measurements. Kalman Gain is a correction gain calculated by KF to minimize the error in the state estimation. Kalman Filter assumes that the system is linear and all the noise signals have zero mean [37]. The two important steps involved in KF are the prediction step in which the filter estimates the system states and state covariance matrix followed by the update step or correction step, which uses the measurement data to update the states and covariance matrix. The main drawback of KF is its applicability to non-linear systems. Most of the systems in real-time are nonlinear and an extended version of KF is required to apply on non-linear systems. This resulted in the development of Extended Kalman Filter.

Extended Kalman Filter is utilized when the system model is non-linear. The initial step is to linearize the system model using Taylor series approximation. This leads to a linear system model and KF can be applied. The performance of the KF depends on the initial values of the parameters. The following equations explain the steps involved in applying EKF to a non-linear system. Any non-linear system can be represented using a system model and measurement model as shown below

$$\dot{x}(t) = f(x(t), u(t)) + w(t) \tag{4}$$

$$y(t) = h(x(t)) + v(t) \tag{5}$$

On applying Taylor series expansion, the linearized system model and measurement model are represented as

$$\delta\dot{x}(t) \approx F(t)\delta x(t) + G(t)w(t) \tag{6}$$

$$\delta y(t) \approx H(t)\delta x(t) + v(t) \tag{7}$$

where F,G,H are Jacobian matrices. The future state and the error covariance matrix can be obtained through KF using the following equations

i) Prediction Step:

$$x^-_{k+1} = f(x_k, u_k) \tag{8}$$

$$P^-_{k+1} = (I + F_{k+1}T)P_k(I + F_{k+1}T)^T + Q_{k+1} \tag{9}$$

ii) Update Step:

$$K_{k+1} = P_{k+1}H^T_{k+1}(H_{k+1}P_{k+1}H^T_{k+1} + R_{k+1})^{-1} \tag{10}$$

$$x^+_{k+1} = x^-_{k+1} + K_{k+1}[y_{k+1} - h(x^-_{k+1})] \tag{11}$$

$$P^+_{k+1} = (I - K_{k+1}H_{k+1})P^-_{k+1} \tag{12}$$

where T is sampling period, K is the Kalman gain, P is the state error covariance matrix, Q is process noise covariance matrix and R is measurement covariance matrix.

B. IMU MECHANIZATION EQUATIONS

IMU measurements are transformed into system states using IMU mechanization equations. Each of the vehicle states can be obtained from respective system models. The orientation system model can be obtained using Direction Cosine Matrix (DCM) or quaternion. In this paper, quaternion based implementation is considered due to its stability [38]. All the system models are defined in local frame. The quaternion system model is defined as

$$\dot{q}^L_B = 0.5 \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} \begin{bmatrix} 0 \\ w^B_{LB_x} \\ w^B_{LB_y} \\ w^B_{LB_z} \end{bmatrix} \tag{13}$$

where a,b,c,d are quaternion components, w^B_{LB} is the corrected gyroscope measurements in body frame B. The corrected gyroscope measurements are obtained by removing gyrosocpe bias (b_g), scalefactor (s_g) errors and also subtracting earth rate (w^L_{IE}) and transport rate (w^L_{EL}). The following are the equations

$$w^B_{LB} = \frac{\hat{w}^B_{IB} - b_g}{1 + s_g} + C^B_L w^L_{IL} \tag{14}$$

$$w^L_{IL} = w^L_{IE} + w^L_{EL} \tag{15}$$

$$w^L_{IE} = \begin{bmatrix} w^e Cos(\phi) & 0 & -w^e Sin(\phi) \end{bmatrix} \tag{16}$$

$$w^L_{EL} = \begin{bmatrix} \frac{v_e}{R_N + h} & \frac{-v_n}{R_M + h} & \frac{-v_e}{R_N + h} tan(\phi) \end{bmatrix} \tag{17}$$

$$v^L = \begin{bmatrix} v_n & v_e & v_d \end{bmatrix} \tag{18}$$

where w^e is earth angular rotation, ϕ is latitude value, h is the height of the vehicle, R_M and R_N are radii of curvature and the velocity in L frame is given as v^L. The velocity is estimated by projecting the accelerometer measurements from B frame to L frame and integrating them after compensating for gravity. Since the L frame is rotating the coriolis effect is also compensated. The velocity state model is represented by the equation shown below

$$\dot{v}^L = C^L_B a^B_{SF} + g^L - (w^L_{EL} + 2w^L_{IE}) \times v^L \tag{19}$$

where a^B_{SF} are accelerometer values in frame B and g^L is gravity vector in L frame. The geodetic position values can be obtained by using the position system model given by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{v_n}{R_M + h} \\ \frac{v_e}{(R_N + h)Cos(\phi)} \\ -v_d \end{bmatrix} \tag{20}$$

C. IMU ERRORS

MEMS IMUs encounter various types of errors, broadly categorized into two groups: deterministic errors and stochastic errors. Deterministic errors, including bias, scale factor, and axis misalignment errors, can be mitigated through calibration techniques [39]. On the other hand, stochastic errors encompass bias instability, scale factor instability, and random noises, all of which exhibit variations over time. In contrast, high-end navigation-grade IMU sensors exhibit minimal stochastic errors, with significantly lower levels of bias instability, scale factor instability, and random noise. The notable factors contributing to the majority of navigation drifts are bias (b_g and b_a), scale factor (s_g and s_a) and random noise (η) [3]. These errors need to be mitigated to reduce the navigation drifts and hence these errors are modeled along with the system states using the following error model equations

$$\tilde{a} = (1 + S_a + \delta S_a)a + B_a + \delta B_a + \eta \tag{21}$$

$$\tilde{w} = (1 + S_g + \delta S_g)w + B_g + \delta B_g + \eta \tag{22}$$

where $\tilde{a}$ is output of accelerometer, a is actual accelerometer value, $\tilde{w}$ is output of gyroscope, w is actual gyroscope measurement, S_a, S_g are accelerometer and gyroscope scale factors, δS_a, δS_b are accelerometer and gyroscope scale factor instability, B_a, B_g is static bias, δB_a, δB_g is bias instability in accelerometer and gyroscope respectively and η is random noise. The ML algorithms treat all the errors and random noises as combined errors and try to predict clean IMU. This paper aims to remove these noises and errors using ML algorithms.

D. IMU GNSS LOOSELY COUPLED FUSION

The external measurements in loosely coupled integration of IMU and GNSS are the position and velocity values estimated directly by GNSS [40]. A simple block diagram explaining IMU GNSS loosely coupled fusion architecture is shown in Fig. 1.

The state space models for each of the navigation states can be found in (13),(19) and (20). The input state model also includes the error model and is given as follows

$$x(t) = \begin{bmatrix} P_{1\times3} & V_{1\times3} & (q^L_B)_{1\times4} & (b_g)_{1\times3} \\ (b_a)_{1\times3} & (s_g)_{1\times3} & (s_a)_{1\times3} & \end{bmatrix}^T \tag{23}$$

where P is the position and V is the velocity in L frame, q^L_B is the attitude, b_g is gyroscope bias, b_a is accelerometer bias, s_g is gyroscope scalefactor and s_a is accelerometer

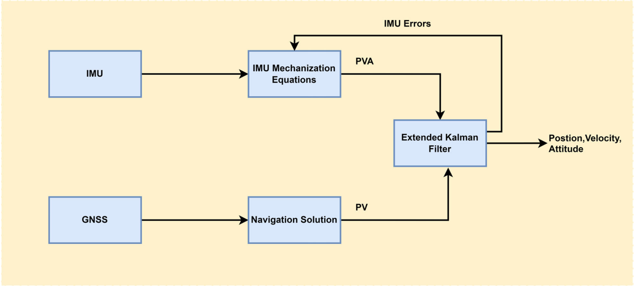

FIGURE 1. Loosely coupled IMU/GNSS sensor fusion architecture.

scalefactor. The measurement model is given by

$$\begin{bmatrix} E \\ N \\ U \\ v_e \\ v_n \\ v_u \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & . & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & . & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & . & 0 \end{bmatrix}_{6\times\ 22} x(t) + \eta \tag{24}$$

where η is measurement noise.

E. IMU GNSS TIGHTLY COUPLED FUSION

The psedorange (ρ) and pseudorangerate ($\dot{\rho}$) are used as external observations to fuse with IMU [41]. The pseudorange and pseudorangerate is the measured distance and the rate of change of distance between GNSS receiver and multiple satellites, respectively. The error parameters in the GNSS measurements include receiver bias and receiver drift. The receiver bias (b_r), known as clock bias, is the time offset in the receiver's internal clock. The receiver drift (d_r) is the rate of change of receiver bias over time. The pseudorange and pseudorangerate measurement model is given by

$$\begin{aligned} \rho^m &= \sqrt{(x - x^m)^2 + (y - y^m)^2 + (z - z^m)^2} \\ &\quad + b_r + \tilde{\epsilon}^m_\rho \qquad (25) \\ \dot{\rho}^m &= 1^m_x.(v_x - v^m_x) + 1^m_y.(v_y - v^m_y) + 1^m_z.(v_z - v^m_z) \\ &\quad + d_r + \epsilon^m_{\dot{\rho}} \qquad (26) \end{aligned}$$

where (x, y, z) is receiver position in ECEF frame, (x^m, y^m, z^m) is satellite position in ECEF frame, (x_x, v_y, v_z) is receiver velocity in ECEF frame, (v^m_x, v^m_y, v^m_z) is satellite velocity in ECEF frame, 1^m is the line of sight unit vector between receiver and satellite. The receiver bias and receiver drift need to be included in the input states and the F, G matrix are modified accordingly for this fusion scheme. So the input states can be modified as

$$x(t) = \begin{bmatrix} P_{3\times1} \\ V_{3\times1} \\ (q^L_B)_{4\times1} \\ (b_g)_{3\times1} \\ (b_a)_{3\times1} \\ (s_g)_{3\times1} \\ (s_a)_{3\times1} \\ b_r \\ d_r \end{bmatrix} \tag{27}$$

The measurement model for tightly coupled fusion is given by

$$\begin{bmatrix} \delta\rho^1 \\ \cdot \\ \cdot \\ \delta\rho^m \\ \delta\dot{\rho}^1 \\ \cdot \\ \cdot \\ \delta\dot{\rho}^m \end{bmatrix} = \begin{bmatrix} (1^1_{IMU})^T & 0_{3\times1} & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ (1^m_{IMU})^T & 0_{3\times1} & 1 & 0 \\ 0_{3\times1} & (1^1_{IMU})^T & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0_{3\times1} & (1^m_{IMU})^T & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta v_x \\ \delta v_y \\ \delta v_z \\ \delta b_r \\ \delta d_r \end{bmatrix} + \begin{bmatrix} \tilde{\epsilon}^1_\rho \\ \cdot \\ \cdot \\ \tilde{\epsilon}^m_\rho \\ \epsilon^1_{\dot{\rho}} \\ \cdot \\ \cdot \\ \epsilon^m_{\dot{\rho}} \end{bmatrix} \tag{28}$$

where $\delta\rho^m$ is the pseudorange error of m^{th} satellite, $\delta\dot{\rho}^m$ is pseudorange rate error of m^{th} satellite, 1^m_{IMU} is the line of sight unit vector between satellite m and user position given by IMU and $\tilde{\epsilon}^m_\rho$ and $\epsilon^m_{\dot{\rho}}$ is zero mean random noise. The position and velocity error states δx, δy, δz, δv_x,δv_y and δv_z. The block diagram explaining the architecture of tightly coupled fusion scheme is shown in Fig. 2.

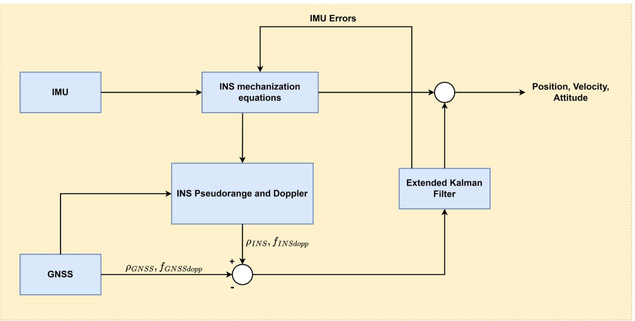
IMU Errors
IMU
INS mechanization equations
Position, Velocity, Attitude
INS Pseudorange and Doppler
Extended Kalman Filter
$\rho_{INS}, f_{INSdopp}$
GNSS
$\rho_{GNSS}, f_{GNSSdopp}$
+
-

FIGURE 2. Tightly coupled IMU/GNSS sensor fusion architecture.

IV. METHODOLOGY

A. INVERSE KINEMATICS

Inverse Kinematics (IK) [38] involves the reverse engineering of the equations, wherein the PVA values are supplied as input to derive gyroscope and accelerometer values. The fundamental principle underlying IK is that when precise PVA values are input into the system, pseudo error-free IMU measurements can be obtained. This approach is particularly useful when there is a need to determine accelerometer and gyroscope values based on a known PVA value. In general, IMU values are typically measured in the IMU body frame. The acceleration in the local frame can be derived by taking the derivative of velocity. Subsequently, the acceleration values from the L frame can be converted into the B frame using DCM and adjusted for the Earth rate and transport rate, as demonstrated in the following equations

$$a^L(t) = \frac{dV^L}{dt} \tag{29}$$

$$a^B_{SF}(t) = C^B_L(t)(a^L(t) - g^L + (w^L_{EL} + 2w^L_{IE}) \times V^L \tag{30}$$

where V^L is the velocity in L frame, C^B_L is the DCM from L to B, g^L is the gravity vector. The rate at which the DCM changes can be used to obtain gyroscope measurements and mitigated for earth and transport rate using the below equation

$$S_g \approx \frac{C^B_L(k-1)C^L_B(k) - I}{T} \tag{31}$$

$$w^B_{LB} = \begin{bmatrix} S_g(3,2) & S_g(1,3) & S_g(2,1) \end{bmatrix} \tag{32}$$

where S_g is the skew matrix and T is the sampling period. This approach is used in this paper to obtain the target values for training ML algorithms.

B. GRADIENT BOOSTING

Boosting is a type of ensemble Machine Learning approach where multiple weak models are combined together by focusing on the mistakes made by the previous models. The final decision of the model is the combined weighted average of each of the models. Gradient Boosting methods use gradient descent method to minimize the loss function and it is used for both classification and regression tasks. Frequently decision trees are employed as weak learners called Gradient Boosting Decision Trees (GBDT) [42]. GBDT employs a series of decision trees connected in sequence, with each subsequent tree trained on the residuals of the preceding one to enhance the overall model performance. Residuals are computed through the Mean Squared Error (MSE) loss function in the context of regression. The most computationally demanding aspect of GBDT revolves around determining optimal split points. Depending on the specific algorithm employed for this purpose, various iterations or versions of GBDT have been developed. CatBoost and LightGBM are types of Gradient Boosting Machines (GBM) where the weak learners are decision trees. The benefits of GBM are the competitive accuracy and training times compared to complex deep learning approaches.

Light Gradient Boosting Machine (LightGBM) represents a variation of GBDT known for its speed and efficiency relative to other Gradient Boosting Machine (GBM) algorithms [43]. Specifically designed to effectively manage substantial volumes of high-dimensional data, LightGBM boasts increased efficiency and reduced training time. Two key techniques contribute to its performance: Gradient-based One-Side Sampling (GOSS), which disregards data with low gradients during information gain computation, and the Exclusive Feature Bundling (EFB) process, bundling mutually exclusive features to decrease the overall number of features.

CatBoost is another variant of GBDT and it is known for its efficiency in handling categorical features [44]. Even though CatBoost algorithm is well known for handling use cases that contain both numerical and categorical data, research showed the efficiency of the CatBoost algorithm in regression tasks. It is optimized for both training and inference speed. It uses techniques such as ordered boosting, a technique used to reduce prediction shifts and overfitting of data and symmetric trees [44], to achieve low inference times while maintaining competitive predictive accuracy. The symmetric trees are the trees where the same condition is applied to split the leaves from the previous tree. The feature and split choice that result in the minimum loss are identified and subsequently employed for all nodes at that level. This algorithm is well-suited for tasks where the training and prediction times are efficient without affecting the accuracy of the model.

C. TRAINING PHASE

In this paper, the primary focus is on removing IMU noises and providing accurate navigation during GNSS outages in urban areas and tunnels where the duration of outages is less than 1 minute.

The sensor fusion of IMU and GNSS data is carried out either loosely coupled scheme or tightly coupled based on the type of GNSS measurements. The resultant reliable PVA values are obtained. In order to obtain pseudo error-free IMU values, the calculated PVA values undergo IK module. The IMU measurements obtained through IK are stored as target values along with the respective noisy input values in a buffer. Once a sufficiently large dataset is accumulated, in this paper depending on the availability of data a minimum of 14000 data points is considered for training. It is important to address that the performance of the ML model depends on the collected training dataset. This work assumes that the GNSS is present and provides data at the start to obtain sufficient training data. To enhance the quality of the training data, emphasis is placed on collecting IMU values during vehicle motion, thus exclusively considering dynamic data portions for training in our experiments.

The proposed ML algorithms take raw accelerometer and gyroscope data along the three axes as input, producing error-free IMU values as output. In the case of CatBoost and LightGBM, six models are trained for each of the accelerometer and gyroscope as output. The training phase's workflow is depicted in Figure 3. One of the major advantages of the proposed methodology is the availability of latest and updated training data. This results in updating the model frequently, which is ideal in IMU denoising as the IMU noises vary with time.

D. TESTING PHASE

In the absence of GNSS, INS acts as a stand-alone system to provide the navigation information to the vehicle. As explained the noisy IMU data when passed through the mechanization equations results in poor PVA values. The latest ML model trained in the training phase is used to clean the IMU data and improve the navigation results. When the vehicle is navigating through the GNSS challenging

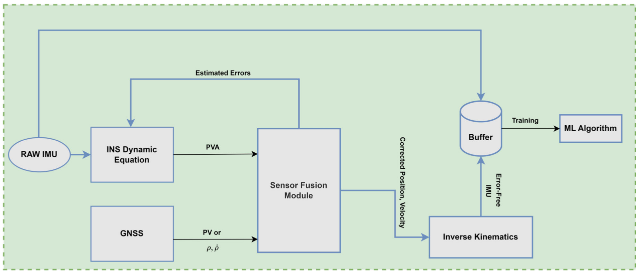

FIGURE 3. Training phase demonstrating fusion scheme, Inverse Kinematics and ML algorithm.

environments the ML model predicts the pseudo error-free IMU data. This process helps in maintaining error drift within acceptable limits, ensuring reliable navigation. Upon the restoration of GNSS signals, the training steps are reinstated. The testing phase is illustrated in Fig. 4.

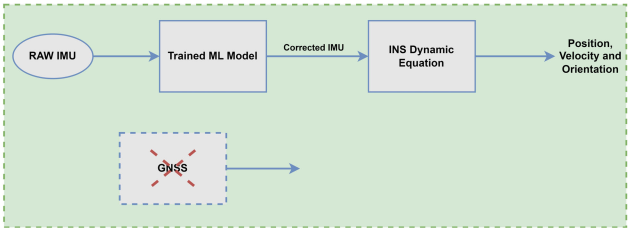

FIGURE 4. Testing phase showing ML prediction during GNSS outages.

There are several sensor fusion techniques available and our proposed approach can be expanded to all the approaches. In this paper, the experiments are carried out in loosely and tightly coupled fusion scenarios. The performance of the proposed LightGBM and CatBoost algorithms are compared with that of CNN. Also, the ML-predicted IMU data is compared with MA and SG filters.

V. RESULTS

The proposed methodology is tested in a loosely coupled fusion scheme over a simulation dataset and a real dataset with different outage durations [45]. The real dataset is used to test the methodology in tightly coupled fusion with a different outage period. The ground truth data in the real dataset is obtained using NovAtel ProPak6 and the specifications of the IMU involved are provided in Table 1 and further details about the data collection can be found in [45]. The IMU data in the simulated dataset is contaminated with noise parameters to resemble the data as that of a MEMS sensor. Since the primary focus of this paper is to compensate IMU and reduce drift during outages in urban areas the maximum outage duration considered in the experiment is 70s. It is worth mentioning that the observability analysis and convergence of the EKF in the case of IMU/GNSS fusion has been studied in the literature [46]. In our experiments, we apply the inverse kinematics when the filter converges, and the residuals become sufficiently small. The device used for carrying out the experiments has 16 GB RAM and 4 GB RTX 2060 graphics card. The sensor fusion equations are run in MATLAB and the ML algorithms are implemented in Python. The MATLAB code for loosely and tightly coupled fusion can be found in [47]. The ML algorithms applied in this paper are CNN, LightGBM and CatBoost. For a fair comparison of training times between these algorithms, the number of iterations parameter is limited to 200. The CNN architecture for the experiment is chosen based on the availability of data and to avoid overfitting. A simple 2-layered CNN architecture is chosen for the comparison of ML algorithms. The CNN architecture used in this paper consists of 2 one-dimensional convolution layers with 100 and 50 nodes respectively, separated by a dropout layer and followed by a flattened layer and the output layer. ReLu activation function is utilized in the convolution layers. The CatBoost and LightGBM algorithms have a tendency to overfit on small training data. To overcome this, both models are tuned with optimal hyperparameters using Optuna hyperparameter tuning framework [48].

TABLE 1. Specifications of IMU in novatel ProPak6.

Specification	Value
Gyroscope Bias stability($/hr$)	0.05
Angular Random Walk($/\sqrt{hr}$)	0.012
Accelerometer Bias stability(mg)	7.5
Velocity Random Walk ($ft/sec/\sqrt{hr}$)	0.23

A. LOOSELY COUPLED FUSION

The simulation trajectory duration is 480s, of which the initial 310s data is assumed to have GNSS. In the rest of the trajectory, the GNSS outages are considered for a period of 50s and 70s. The trajectory of the simulation dataset along with the outage durations are illustrated in Fig. 5. The portions where GNSS is available are shown in blue and the outages are shown in red. The proposed ML algorithms learn the combined error and random noise of the raw IMU measurements. Since it is difficult to distinguish between true IMU measurements corresponding to true dynamics and erroneous IMU measurements, we evaluate the performance of the proposed ML algorithms by applying the mechanization equations on the resulting IMU and compare the RMSE of the output PVA against the ground truth PVA. A 50s duration of the trajectory is considered and the ground truth PVA values are passed through the IK to obtain pseudo error-free IMU data. The calculated IMU data is passed through the mechanization equations and PVA values are obtained. The RMSE value of the ground truth PVA and the PVA value obtained using IK calculated IMU is tabulated in Table 2 and the low RMSE values prove the efficiency of IK.

The initial 310s trajectory is assumed to have the optimal conditions where GNSS is accessible and reliable, the sensor fusion of IMU and GNSS is carried out by EKF. Simultaneously, the obtained PVA values are passed through the IK module to gather pseudo error-free IMU data which acts as ground truth labels in training ML algorithms. The IMU data is stored in the buffer along with the corresponding

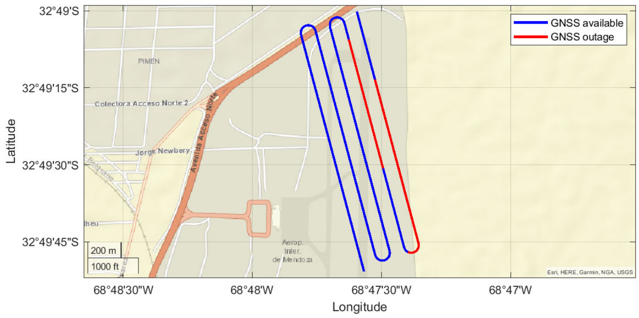

FIGURE 5. Simulated data trajectory highlighting training (blue) and testing (red) portions.

TABLE 2. RMSE analysis on PVA obtained using pseudo error-free IMU.

PVA	Error
East(m)	0.692
North(m)	0.88
Up(m)	4.13
Velocity East (m/s)	0.036
Velocity North (m/s)	0.045
Velocity Up (m/s)	0.209
Roll (deg)	0.0002
Pitch (deg)	0.00005
Heading (deg)	0.0001

raw IMU data. The data collected in the buffer until the GNSS outage occurs. The collected buffer data is passed to the proposed ML models for training. The dataset is divided with a split ratio of 80:20 into training and validation data. The hyperparameters used for this trajectory are shown in Table 3 and Table 4. A comparison of training time for all the algorithms is tabulated in Table 5.

TABLE 3. List of optimal hyperparameters for LightGBM.

HyperParameter	**Value**
num_leaves	544
max_depth	20
learning_rate	0.068
n_estimators	200
min_data_in_leaf	22
colsample_bytree	0.95

TABLE 4. List of optimal hyperparameters for CatBoost.

HyperParameter	**Value**
L2	1
max_depth	10
learning_rate	0.076
iterations	200
min_data_in_leaf	94
colsample_bylevel	0.995

TABLE 5. Comparision of training time for various algorithms.

ML Algorithm	**Time(s)**
CNN	83.175
CatBoost	33.73
LightGBM	7.15

Once the GNSS outage occurs, the raw IMU data is passed through the latest trained ML models and the denoised IMU data are predicted. The predicted IMU data is then passed through IMU mechanization equations and the PVA values are calculated for both of the outage durations. The RMSE values of all ML-predicted IMU are compared to IMU data obtained post-application of MA and SG filters are shown in Table 6 and Table 7. Notably, the ML algorithms exhibit better performance over traditional signal processing methods in acquiring cleaner IMU data.

TABLE 6. RMSE analysis of IMU during the 50s GNSS outage scenario.

	MA	SG	CNN	CB	LGBM
Gyro_x(rad/s)	0.0035	0.0034	0.0006	0.00018	**0.00016**
Gyro_y(rad/s)	0.0071	0.007	0.0009	0.0002	**0.0001**
Gyro_z(rad/s)	0.0105	0.0104	0.0003	0.0009	**0.0004**
Accel_x(m/s^2)	0.1562	0.0012	0.0002	0.0012	**0.0002**
Accel_y(m/s^2)	0.3139	0.3137	0.0029	0.0028	**0.0027**
Accel_z(m/s^2)	0.634	0.633	0.1218	0.0031	**0.0023**

TABLE 7. RMSE analysis of IMU during the 70s GNSS outage scenario.

	MA	SG	CNN	CB	LGBM
Gyro_x(rad/s)	0.0299	0.0045	0.0017	**0.0013**	0.0015
Gyro_y(rad/s)	0.0193	0.0082	0.0014	**0.0003**	0.0008
Gyro_z(rad/s)	0.0299	0.0128	0.0038	**0.0014**	0.0021
Accel_x(m/s^2)	0.1563	0.1562	0.0065	0.0025	**0.0011**
Accel_y(m/s^2)	0.5699	0.3346	0.0547	**0.0198**	0.0282
Accel_z(m/s^2)	0.6635	0.6384	0.1398	0.0078	**0.0074**

In addition, all PVA values obtained using ML algorithms are compared with the values obtained using raw IMU are shown in Table 8 and Table 9. The 2-dimensional position plot for the 50s outage duration is shown in Fig. 6. The ML-predicted position plots are overlapped and a zoomed position plot containing only ML algorithms is shown in Fig. 7. Similarly, the position plot for 70s outage scenario is shown in Fig. 8 and plot containing only ML algorithms is shown in Fig. 9. The results obtained show the satisfactory performance of ML algorithms to reduce the error drift during GNSS outages. Among the ML algorithms, though the performance of CatBoost and LightGBM algorithms are better than CNN, the training time of LightGBM provides an edge in real-time applications.

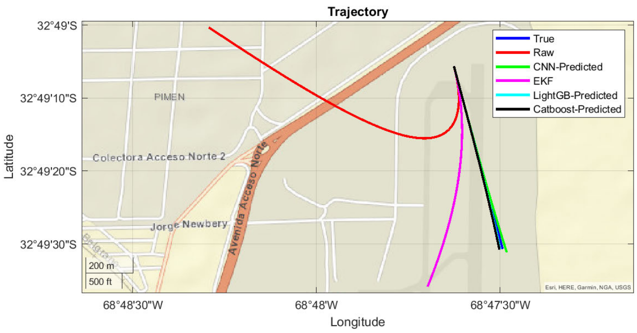

FIGURE 6. Position plot of reference, raw, EKF-calculated IMU, CNN, CatBoost and LightGBM denoised IMU for 50s GNSS outage in simulated data. The true and ML predicted position plots seem overlapped from certain altitude.

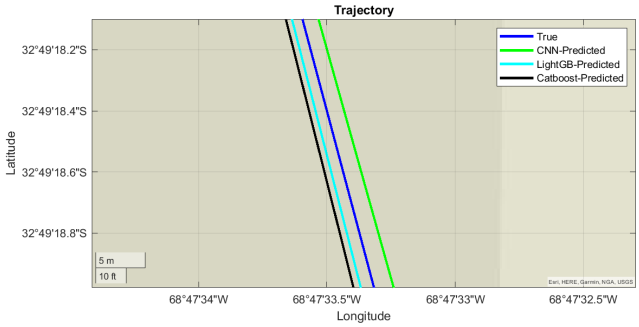

FIGURE 7. Zoomed position plot of reference, CNN, CatBoost and LightGBM denoised IMU for 50s GNSS outage in simulated data.

TABLE 8. RMSE analysis of PVA during the GNSS for 50s in simulated data.

	Raw	CNN	CatBoost	LightGBM
East(m)	492.14	6.561	5.160	**3.858**
North(m)	341.63	5.566	1.096	**0.695**
Up(m)	399.16	63.883	4.124	**4.123**
Velocity East (m/s)	32.06	0.462	0.379	**0.303**
Velocity North (m/s)	26.15	0.381	0.058	**0.034**
Velocity Up (m/s)	22.35	3.306	0.209	**0.208**
Roll (deg)	6.79	0.104	0.121	**0.104**
Pitch (deg)	10.97	0.135	0.03	**0.029**
Heading (deg)	63.80	0.492	0.02	**0.017**

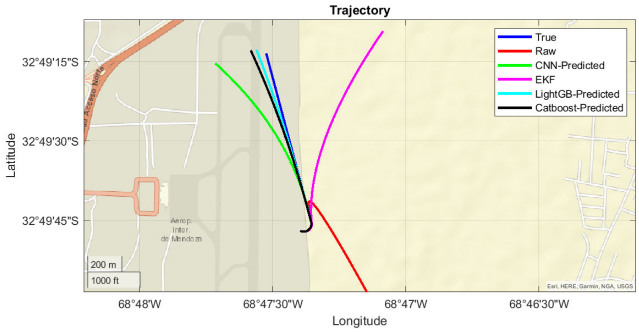

FIGURE 8. Position plot of reference, raw, EKF-calculated IMU, CNN, CatBoost and LightGBM denoised IMU for 70s GNSS outage in simulated data.

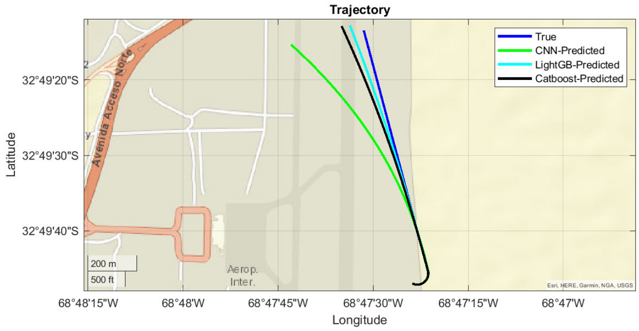

FIGURE 9. Position plot of reference, CNN, CatBoost and LightGBM denoised IMU for 70s GNSS outage in simulated data.

The loosely coupled scheme is also tested using a dataset collected in real time. The complete trajectory duration is 200s. The GNSS is not available for a duration of 30s. The trajectory with the outage portion is shown in Fig. 10.

TABLE 9. RMSE analysis of PVA during the GNSS for 70s in simulated data.

	Raw	CNN	CatBoost	LightGBM
East(m)	635.91	117.866	37.117	**22.583**
North(m)	1336.65	25.572	**8.936**	9.679
Up(m)	785.31	151.325	**4.766**	5.602
Velocity East (m/s)	31.75	10.543	1.634	**1.059**
Velocity North (m/s)	68.09	0.994	**0.272**	0.321
Velocity Up (m/s)	34.49	5.326	**0.199**	0.216
Roll (deg)	9.79	1.061	0.266	**0.179**
Pitch (deg)	15.01	0.120	0.084	**0.07**
Heading (deg)	23.48	2.952	**0.072**	0.119

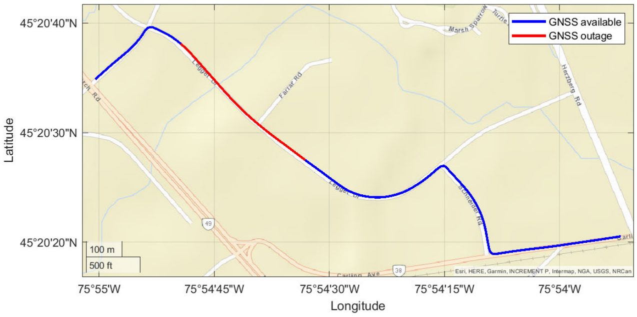

FIGURE 10. Real trajectory with training (blue) and testing (red).

As explained earlier, EKF is used to fuse the IMU GNSS data when the GNSS is available. The IK is applied to the obtained PVA values and the resultant pseudo error-free IMU measurements along with noisy IMU data is stored in the buffer except for the portion of trajectory with GNSS outages. The buffer data is then split into training and validation datasets and all the ML models are trained. The architecture of CNN is considered to be the same as explained above and the hyperparameters used for training LightGBM and CatBoost are provided in Table 10 and Table 11 respectively.

TABLE 10. List of optimal hyperparameters for LightGBM.

HyperParameter	Value
num_leaves	374
max_depth	20
learning_rate	0.021
n_estimators	200
min_data_in_leaf	35
colsample_bytree	0.95

TABLE 11. List of optimal hyperparameters for CatBoost.

HyperParameter	Value
L2	1
max_depth	6
learning_rate	0.065
iterations	200
min_data_in_leaf	44
colsample_bylevel	0.75

During the GNSS challenging conditions, the trained ML models are used to predict the denoised IMU. The PVA values are obtained by passing the ML-predicted IMU through IMU mechanization equations. The comparison of PVA values between different predicted IMUs against raw IMU is shown in Table 12. The 2-dimensional position plot of all the predicted IMU during the GNSS outage is shown in Fig. 11, followed by the plot showing only ML-predicted IMU in Fig. 12.

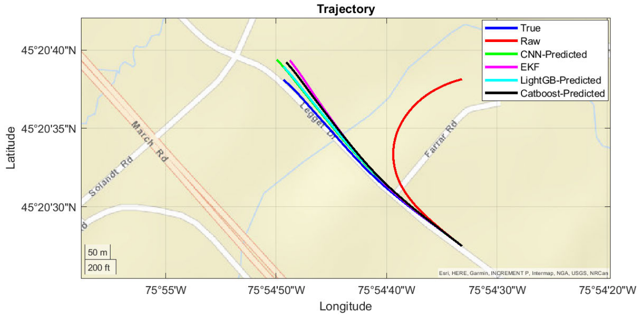

FIGURE 11. Position plot of reference, raw, EKF-calculated, CNN, CatBoost and LightGBM denoised IMU during GNSS outage in real-time data in loosely coupled fusion.

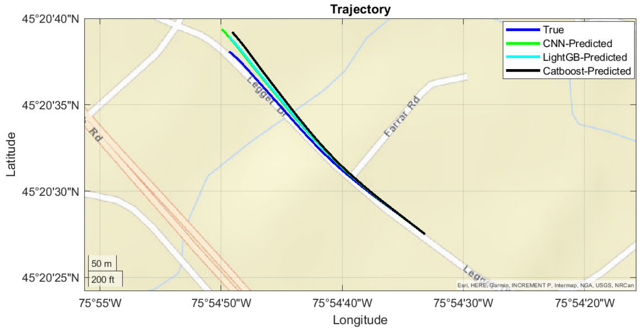

FIGURE 12. Zoomed position plot of reference, CNN, CatBoost and LightGBM denoised IMU during GNSS outage in real-time data in loosely coupled fusion.

TABLE 12. RMSE analysis of PVA during the GNSS outage in real-time data in loosely coupled fusion.

	Raw	CNN	CatBoost	LightGBM
East(m)	135.42	5.729	2.11	**0.57**
North(m)	12.81	16.67	15.18	**9.85**
Up(m)	72.71	8.07	**0.71**	2.27
Velocity East (m/s)	15.55	0.55	0.24	**0.07**
Velocity North (m/s)	1.60	1.61	1.34	**0.87**
Velocity Up (m/s)	6.63	0.71	**0.19**	0.27
Roll (deg)	4.81	0.102	0.05	**0.02**
Pitch (deg)	5.88	0.35	**0.05**	0.06
Heading (deg)	10.81	2.27	0.78	**0.47**

The position plots and RMSE table shows the effectiveness of the proposed methodology to remove the IMU noise and maintaining the drifts within limits. The LightGBM algorithm outperformed all the other algorithms and proved beneficial. It is important to understand that the performance of ML algorithms improves with the improvement in the training data. The training data in our work can be improved by improving the fusion output to obtain better pseudo error-free IMU data for training purposes. The same real dataset is used in tightly coupled fusion with longer outage duration and the results are evaluated.

B. TIGHTLY COUPLED FUSION

The tightly coupled fusion combines IMU GNSS data using the pseudorange and pseudorangerate as the measurement model. The real time collected dataset explained previously is now utilized to test the proposed methodology in tightly coupled scheme. In the entire trajectory, it is assumed that the number of visible satellites is zero for a duration of 40s. The training and testing portions are illustrated in Fig. 13.

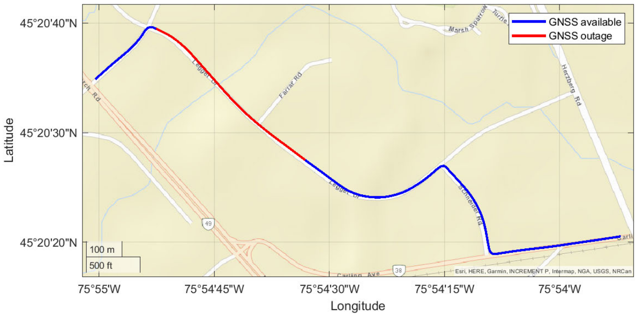

FIGURE 13. Real trajectory with training (blue) and testing (red) for tightly coupled fusion.

During favorable conditions, the number of visible satellites is more than 4 which provides better fusion output in the tightly coupled fusion. In the dataset used, at least 5 satellites are available at any given time when GNSS is present. EKF is used to fuse IMU GNSS data and reliable PVA is obtained. IK is employed on the obtained PVA values and pseudo error-free IMU are calculated. These IMU values along with the noisy raw input are stored in the buffer. The ML models are trained with the data collected in the buffer excluding the outage portions. This data is split into training and validation sets with a ratio of 80:20. The CNN architecture is considered to be the same as explained for loosely coupled fusion and the hyperparameters for Catboost and LightGBM are provided in Table 13 and Table 14.

TABLE 13. List of optimal hyperparameters for LightGBM.

HyperParameter	Value
num_leaves	685
max_depth	20
learning_rate	0.045
n_estimators	200
min_data_in_leaf	49
colsample_bytree	0.995

TABLE 14. List of optimal hyperparameters for CatBoost.

HyperParameter	Value
L2	1
max_depth	10
learning_rate	0.06
iterations	200
min_data_in_leaf	85
colsample_bylevel	0.787

When the GNSS outage occurs, since it is assumed that the number of visible satellites is zero, the fusion mechanism fails to provide even partial correction updates which is one of the advantages of tightly coupled fusion mechanism. So the system relies on INS during the outage. The raw IMU data is passed through the ML models and the denoised IMU measurements are predicted. These ML-predicted IMU are passed through mechanization equations and the PVA values are estimated. The RMSE values of PVA calculated using ML-predicted IMU are compared with those obtained using raw IMU values in Table. 15. The 2-dimensional position plot comparing all positions obtained using predicted IMU is shown in Fig. 14. The ML predicted position plots are overlapped and zoomed position plot with only ML algorithms is shown in Fig. 15.

The position plots and the RMSE values prove the effectiveness of tree-based boosting algorithms to denoise IMU and reduce navigation drift in tightly coupled fusion scheme. Though the performance of CatBoost and LightGBM is satisfactory, LightGBM is efficient in real-time implementation due to its lower training time.

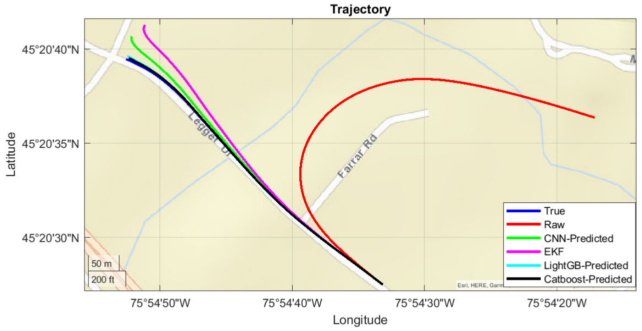

FIGURE 14. Position plot of reference, raw, EKF-calculated, CNN, CatBoost and LightGBM denoised IMU during GNSS outage in real-time data in tightly coupled fusion. The ML-predicted plots are overlapped with the reference.

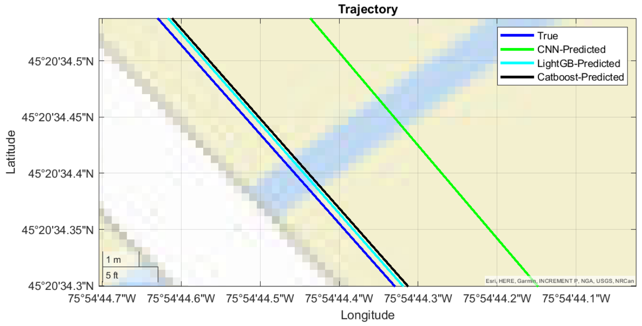

FIGURE 15. Zoomed position plot of reference, CNN, CatBoost and LightGBM denoised IMU during GNSS outage in real-time data in Tightly coupled fusion.

TABLE 15. RMSE analysis of PVA during the GNSS outage in real-time data in Tightly coupled fusion.

	Raw	CNN	CatBoost	LightGBM
East(m)	303.53	2.56	2.04	**0.97**
North(m)	25.74	13.09	**0.53**	2.2
Up(m)	139.53	104.64	**8.64**	9.8
Velocity East (m/s)	25.07	0.42	0.24	**0.21**
Velocity North (m/s)	5.39	1.32	**0.06**	0.15
Velocity Up (m/s)	10.35	6.74	**0.54**	0.6
Roll (deg)	5.29	0.42	**0.06**	0.08
Pitch (deg)	8.62	0.21	0.23	**0.05**
Heading (deg)	14.43	3.24	0.90	**0.43**

VI. CONCLUSION AND FUTURE WORK

The paper introduced an Inverse Kinematics (IK) approach to acquire ground truth data for training Machine Learning (ML) algorithms, aiming to address navigation drifts during GNSS outages. The trained ML models exhibited superior performance compared to traditional signal processing methods. Notably, the proposed LightGBM and CatBoost algorithms effectively handled IMU noise, yielding satisfactory estimations of Position, Velocity, and Attitude (PVA) values. The key advantage of utilizing Inverse Kinematics for obtaining training data lies in the real-time availability of data. This facilitates timely model updates with the latest information to adapt to new patterns. The future work is to apply this methodology on larger dataset in real-time to estimate the optimal buffer size. This approach could also be complemented with other fusion algorithms like Particle Filter (PF), Unscented Kalman Filter (UKF). More ML algorithms could be implemented to provide a comparative study of efficiency of various algorithms.

REFERENCES

[1] G. Chatterjee, L. Latorre, F. Mailly, P. Nouet, N. Hachelef, and C. Oudea, "MEMS based inertial measurement units," in *Proc. Symp. Design, Test, Integr. Packag. MEMS/MOEMS (DTIP)*, Apr. 2015, pp. 1–5.

[2] Q. Zhang, X. Niu, and C. Shi, "Impact assessment of various IMU error sources on the relative accuracy of the GNSS/INS systems," *IEEE Sensors J.*, vol. 20, no. 9, pp. 5026–5038, May 2020.

[3] D. Unsal and K. Demirbas, "Estimation of deterministic and stochastic IMU error parameters," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, Apr. 2012, pp. 862–868.

[4] H. Chen, W. Wu, S. Zhang, C. Wu, and R. Zhong, "A GNSS/LiDAR/IMU pose estimation system based on collaborative fusion of factor map and filtering," *Remote Sens.*, vol. 15, no. 3, p. 790, Jan. 2023.

[5] A. Abosekeen, U. Iqbal, A. Noureldin, and M. J. Korenberg, "A novel multi-level integrated navigation system for challenging GNSS environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4838–4852, Aug. 2021.

[6] R. Kumar Reddy Damagatla and M. Atia, "A novel approach for IMU denoising using machine learning," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Jul. 2023, pp. 1–6.

[7] W. Ye, Z. Liu, C. Li, and J. Fang, "Enhanced Kalman filter using noisy input Gaussian process regression for bridging GPS outages in a POS," *J. Navigat.*, vol. 71, no. 3, pp. 565–584, May 2018.

[8] S. Zhao, Y. Zhou, and T. Huang, "A novel method for AI-assisted INS/GNSS navigation system based on CNN-GRU and CKF during GNSS outage," *Remote Sens.*, vol. 14, no. 18, p. 4494, Sep. 2022.

[9] J. Liu and G. Guo, "Vehicle localization during GPS outages with extended Kalman filter and deep learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[10] B. Li, G. Chen, Y. Si, X. Zhou, P. Li, P. Li, and T. Fadiji, "GNSS/INS integration based on machine learning LightGBM model for vehicle navigation," *Appl. Sci.*, vol. 12, no. 11, p. 5565, May 2022.

[11] R. Gonzalez and C. A. Catania, "A statistical approach for optimal order adjustment of a moving average filter," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2018, pp. 1542–1546.

[12] A. Waegli, J. Skaloud, S. Guerrier, M. E. Parés, and I. Colomina, "Noise reduction and estimation in multiple micro-electro-mechanical inertial systems," *Meas. Sci. Technol.*, vol. 21, no. 6, Jun. 2010, Art. no. 065201.

[13] Y.-H. Tu and C.-C. Peng, "An ARMA-based digital twin for MEMS gyroscope drift dynamics modeling and real-time compensation," *IEEE Sensors J.*, vol. 21, no. 3, pp. 2712–2724, Feb. 2021.

[14] G. Yuan, H. Liang, K. He, and Y. Xie, "Research on signal de-noising technique for MEMS gyro," in *Proc. 3rd Int. Symp. Syst. Control Aeronaut. Astronaut.*, Jun. 2010, pp. 1288–1291.

[15] S. Nassar, K.-P. Schwarz, N. EL-Sheimy, and A. Noureldin, "Modeling inertial sensor errors using autoregressive (AR) models," *Navigation*, vol. 51, no. 4, pp. 259–268, Dec. 2004.

[16] Y. Gan, L. Sui, J. Wu, B. Wang, Q. Zhang, and G. Xiao, "An EMD threshold de-noising method for inertial sensors," *Measurement*, vol. 49, pp. 34–41, Mar. 2014.

[17] C. Liu, Z. Yang, Z. Shi, J. Ma, and J. Cao, "A gyroscope signal denoising method based on empirical mode decomposition and signal reconstruction," *Sensors*, vol. 19, no. 23, p. 5064, Nov. 2019.

[18] M. Karaim, A. Noureldin, and T. B. Karamat, "Low-cost IMU data denoising using savitzky-golay filters," in *Proc. Int. Conf. Commun., Signal Process., their Appl. (ICCSPA)*, Mar. 2019, pp. 1–5.

[19] J. He, C. Sun, and P. Wang, "Noise reduction for MEMS gyroscope signal: A novel method combining ACMP with adaptive multiscale SG filter based on AMA," *Sensors*, vol. 19, no. 20, p. 4382, Oct. 2019.

[20] C. W. Kang, C. G. Park, and N. I. Cho, "Improvement of INS-GPS integrated navigation system using wavelet thresholding," *J. Korean Soc. Aeronaut. Space Sci.*, vol. 37, no. 8, pp. 767–773, 2009.

[21] C.-H. Kang, S.-Y. Kim, and C.-G. Park, "Improvement of a low cost MEMS inertial-GPS integrated system using wavelet denoising techniques," *Int. J. Aeronaut. Space Sci.*, vol. 12, no. 4, pp. 371–378, Dec. 2011.

[22] R. Gonzalez and C. A. Catania, "Time-delayed multiple linear regression for de-noising MEMS inertial sensors," *Comput. Electr. Eng.*, vol. 76, pp. 1–12, Jun. 2019.

[23] M. Brossard, S. Bonnabel, and A. Barrau, "Denoising IMU gyroscopes with deep learning for open-loop attitude estimation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4796–4803, Jul. 2020.

[24] S. Han, Z. Meng, X. Zhang, and Y. Yan, "Hybrid deep recurrent neural networks for noise reduction of MEMS-IMU with static and dynamic conditions," *Micromachines*, vol. 12, no. 2, p. 214, Feb. 2021.

[25] H. Liu, X. Wei, M. Perusquía-Hernández, N. Isoyama, H. Uchiyama, and K. Kiyokawa, "DUET: Improving inertial-based odometry via deep IMU online calibration," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, Aug. 2023, doi: 10.1109/TIM.2023.3306833.

[26] Y. Gao, D. Shi, R. Li, Z. Liu, and W. Sun, "Gyro-Net: IMU gyroscopes random errors compensation method based on deep learning," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1471–1478, Mar. 2023, doi: 10.1109/LRA.2022.3230594.

[27] H. Chen, T. M. Taha, and V. P. Chodavarapu, "Towards improved inertial navigation by reducing errors using deep learning methodology," *Appl. Sci.*, vol. 12, no. 7, p. 3645, Apr. 2022, doi: 10.3390/app12073645.

[28] M. A. Esfahani, H. Wang, K. Wu, and S. Yuan, "OriNet: Robust 3-D orientation estimation with a single particular IMU," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 399–406, Apr. 2020, doi: 10.1109/LRA.2019.2959507.

[29] M. Abolfazli Esfahani, H. Wang, K. Wu, and S. Yuan, "AbolDeepIO: A novel deep inertial odometry network for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1941–1950, May 2020, doi: 10.1109/TITS.2019.2909064.

[30] H. Tang, X. Niu, T. Zhang, Y. Li, and J. Liu, "OdoNet: Untethered speed aiding for vehicle navigation without hardware wheeled odometer," *IEEE Sensors J.*, vol. 22, no. 12, pp. 12197–12208, Jun. 2022, doi: 10.1109/JSEN.2022.3169549.

[31] J. Wang, D. Weng, X. Qu, W. Ding, and W. Chen, "A novel deep odometry network for vehicle positioning based on smartphone," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023, doi: 10.1109/TIM.2023.3240227.

[32] S. Sun, D. Melamed, and K. Kitani, "IDOL: Inertial deep orientation-estimation and localization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 6128–6137.

[33] Z. Zhu, Y. Bo, and C. Jiang, "A MEMS gyroscope noise suppressing method using neural architecture search neural network," *Math. Problems Eng.*, vol. 2019, pp. 1–9, Nov. 2019.

[34] C. Jiang, Y. Chen, S. Chen, Y. Bo, W. Li, W. Tian, and J. Guo, "A mixed deep recurrent neural network for MEMS gyroscope noise suppressing," *Electronics*, vol. 8, no. 2, p. 181, Feb. 2019.

[35] D. Engelsman, "Data-driven denoising of accelerometer signals," Ph.D. dissertation, Univ. Haifa, Haifa, Israel, 2022.

[36] A. E. Mahdi, A. Azouz, A. E. Abdalla, and A. Abosekeen, "A machine learning approach for an improved inertial navigation system solution," *Sensors*, vol. 22, no. 4, p. 1687, Feb. 2022.

[37] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *Proc. 8th Int. Conf. Intell. Netw. Intell. Syst. (ICINIS)*, Nov. 2015, pp. 74–77.

[38] S. S. Kourabbaslou, A. Zhang, and M. M. Atia, "A novel design framework for tightly coupled IMU/GNSS sensor fusion using inverse-kinematics, symbolic engines, and genetic algorithms," *IEEE Sensors J.*, vol. 19, no. 23, pp. 11424–11436, Dec. 2019.

[39] S. Du, W. Sun, and Y. Gao, "MEMS IMU error mitigation using rotation modulation technique," *Sensors*, vol. 16, no. 12, p. 2017, Nov. 2016.

[40] M. Kim, C. Park, and J. Yoon, "The design of GNSS/IMU loosely-coupled integration filter for wearable EPTS of football players," *Sensors*, vol. 23, no. 4, p. 1749, Feb. 2023.

[41] Q. Zhou, H. Zhang, Y. Li, and Z. Li, "An adaptive low-cost GNSS/MEMS-IMU tightly-coupled integration system with aiding measurement in a GNSS signal-challenged environment," *Sensors*, vol. 15, no. 9, pp. 23953–23982, Sep. 2015.

[42] Z. Wen, B. He, R. Kotagiri, S. Lu, and J. Shi, "Efficient gradient boosted decision tree training on GPUs," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2018, pp. 234–243.

[43] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–24.

[44] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–27.

[45] M. Atia, "Design and simulation of sensor fusion using symbolic engines," *Math. Comput. Model. Dyn. Syst.*, vol. 25, no. 1, pp. 40–62, Jan. 2019.

[46] J. Farrell, *Aided Navigation: GPS With High Rate Sensors*. New York, NY, USA: McGraw-Hill, 2008.

[47] M. Atia, "EmbeddedMultisensorSystems/EMS-SymINSGNSS v1.0.0," Zenodo, Feb. 2018, doi: 10.5281/zenodo.1183569.

[48] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.

ROHAN KUMAR REDDY DAMAGATLA received the B.Tech. degree in electrical and electronics engineering from Vellore Institute of Technology, India, in 2017. He is currently pursuing the M.A.Sc. degree in systems and computer engineering with Carleton University. He works as a Research Assistant with the Embedded and Multi-Sensory Systems Laboratory (EMSLab), Carleton University. His research interests include machine learning and sensor fusion systems.

MOHAMED ATIA (Senior Member, IEEE) received the B.S. and M.Sc. degrees in computer systems from Ain Shams University, Cairo, Egypt, in 2000 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from Queen's University, Kingston, ON, Canada, in 2013. He is currently an Associate Professor with the Department of Systems and Computer Engineering, Carleton University. He is also the Founder and the Director of the Embedded and Multi-Sensory Systems Laboratory (EMSLab), Carleton University. He has more than 90 publications and several granted patents in the area of sensor fusion and multi-sensor navigation systems. His research interests include sensor fusion, navigation systems, artificial intelligence, and robotics.