



HAL
open science

Navigation Context Adaptive Fault Detection and Exclusion Strategy based On Deep Learning & Information Theory: Application To a GNSS/IMU integration

Nesrine Harbaoui, Nourdine Aït Tmazirte, Khoder Makkawi, Maan El Badaoui El Najjar

► To cite this version:

Nesrine Harbaoui, Nourdine Aït Tmazirte, Khoder Makkawi, Maan El Badaoui El Najjar. Navigation Context Adaptive Fault Detection and Exclusion Strategy based On Deep Learning & Information Theory: Application To a GNSS/IMU integration. ION GNSS+ 2021, Sep 2021, Missouri, United States. hal-03399136

HAL Id: hal-03399136

<https://hal.science/hal-03399136>

Submitted on 23 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Navigation Context Adaptive Fault Detection and Exclusion Strategy based On Deep Learning & Information Theory: Application To a GNSS/IMU integration

Nesrine HARBAOUI, *CRISAL laboratory, University of Lille*
Nourdine Aït Tmazirte, *COSYS-LEOST, University of Gustave Eiffel*
Khoder Makkawi, *CRISAL laboratory, University of Lille*
Maan El Badaoui El Najjar, *CRISAL laboratory, University of Lille*

BIOGRAPHY

Nesrine Harbaoui PhD student at Lille University and CRISAL laboratory. Received her Master's in Control for Green Mechatronics from Bourgogne Franche Comté University in Dec 2019. She is a member of TOSyMA (Tolérance aux fautes des Systèmes Mobiles Autonomes) team of the CRISAL laboratory UMR CNRS 9189.

Nourdine Aït Tmazirte Nourdine Aït Tmazirte works at French Institute of Technology Railenium since 2018. He got his engineering and M.Sc. degree in automation engineering from Ecole Centrale de Lille, France, both in 2010. His research interests include multi-sensor fault tolerant fusion for localization and integrity assessment.

Khoder Makkawi Lecturer at Polytech Lille and Researcher at CRISAL laboratory. Received the Ph.D. degree in 2020, in collaboration between Lille University and Lebanese University. He is a member of the TOSyMA (Tolérance aux fautes des Systèmes Mobiles Autonomes) team of the CRISAL laboratory UMR CNRS 9189.

Pr. Maan El Badaoui El Najjar Head of ToSyMa team in CRISAL laboratory. The research activities focused on on-line system Diagnosis (FDE, FDI...) and fault tolerant multi-sensor data fusion which is able to deal with faulty sensors of dynamic systems and their desired capacities for robotics and autonomous vehicles applications.

ABSTRACT

The growing demand for a reliable localization function in safety-relevant autonomous applications has prompted the navigation community to take a close interest in Fault Detection algorithms. Particularly powerful, these methods have as common limitation, the need to define a priori probabilities of fault(s) presence or absence in order to set a detection threshold. In the case of systems insensitive to external phenomena, these probabilities can be obtained although with difficulty. However, in a multi-sensor localization system using, among other, Global Navigation Satellites System (GNSS), this task can prove to be delicate. Indeed, these global navigation systems suffer from perturbations induced by the local environment (buildings in urban canyons, foliage in forests, intentional or nonintentional interferences...) causing local feared events such as multipath, Non-Line-Of-Sight (NLOS) or GNSS outage. The a priori probabilities to face this type of event, and therefore to observe one or more inconsistent measures, are very unpredictable. This makes the task of setting a threshold difficult. In this study, we investigate the combined use of 1) GNSS expertise: allowing to identify the different parameters, making it possible to monitor the state of health of the measurements, and therefore of the global system, and 2) the implementation of a deep learning scheme, as a decision-making support, considering these inputs, and providing the adequate a priori probability of fault presence.

I. INTRODUCTION

In order to be certified by the competent authorities, all the elements constituting a localization function integrating a safety-critical application, must demonstrate that their behavior is predictable and that it obeys a predetermined bounded logic [1]. This involves, for example, establishing measures against the divergence of a sequential filter (like the Kalman Filter) or even demonstrating the merits of a measurements weighting strategy [2]. But the safety requirement is not the only objective to be achieved. To be massively adopted, the developed function must also, among others, be precise, available, continuous, robust to navigation context changes and finally low cost. Regarding accuracy, a multi-sensor strategy (e.g. GNSS/INS/odometry) can achieve the set objective [3], [4]. But the greater the number of input measurements, the greater the probability of encountering an inconsistent measurement. A diagnostic layer is therefore necessary [5]. The most conservative diagnostic policy, which

would consist in making the localization function unavailable at the slightest suspicion of a fault, does not constitute a problem in terms of certification. However, it negatively impacts availability and continuity.

To overcome this problem, mitigation techniques also known as fault management or fault tolerance make it possible to reduce unavailability by isolating the suspicious measurement(s). But it is essential to prove that the taken measure(s), for instance a measurement exclusion, or a reweighting procedure, is(are) the right one(s) [6]. To cite an example, it is necessary to prove that an excluded measure should indeed be excluded and that it is the only one that should be excluded. In other words, the assessment of the number of false alarms and missed detections is necessary [7]. However, the proof is complicated to establish when dealing with on field acquired data (in opposition to simulated data provided by a GNSS simulator). Indeed, labeling faults such as multipath or NLOS in the pseudorange level is extremely complex and requires accurate 3d map and a powerful raytracing engine [8], [9]. For this reason, a validation at the position level is generally carried out. At this level, the behavior predictability is ensured by the provision of a protection level capable of limiting the unknown position error [10]. One has to avoid the case where the protection level does not limit the position error but exceeds the alarm limit defined by the application. This is called the integrity risk. An infinitesimal failure rate, which is called THR for Tolerable Hazardous Rate, is generally accepted as integrity risk and is of the order of 10^{-7} to 10^{-9} per hour of use [11].

In summary in order to achieve all the objectives, it is necessary to implement actions at different levels of the localization function. To achieve precision and availability, a multi-sensor fusion is strongly recommended [12], [13], [14]. However, the multitude of inputs increases the probability of facing measurement faults. Therefore, it is required to implement a diagnostic layer which can itself impact the availability of the system. It is then necessary to develop a fault management layer for which decisions must be validated. In addition, a complexity of interest to the community over the last decade: the change of environment. Indeed, GNSS has the particularity of being very dependent on the environment surrounding the receiver [15]. Since the principle of position determination is based on the reception of signals emitted by satellites over 20 000km, obstacles encountered, for land navigation applications, can reflect or obstruct signals in urban canyons or forests. However, although the probability of facing a fault varies according to the crossed environment, most fault detection techniques are based on the setting of a single heuristic threshold, and focus only on one criterion (i.e. either the probability of false alarm P_{fa} or the probability of missed detection P_{md}) [16].

In this study we propose a complete framework allowing a localization function to target all the requirements. For this, we propose to merge the data from a GNSS (raw data), and IMU with a dual filter to the Kalman filter, the nonlinear information filter. A discussion on the interest of the informational form is proposed in section II. Then we integrate a diagnostic layer based on informational divergence, the α -Rényi divergence. Next, we present an optimization function for thresholding taking in account both Pfa and Pmd, this part is presented in section III.1. We develop a deep learning-based approach to assess the crossed environment and to estimate as closely as possible the probability of facing a fault detailed in section IV. Finally, the proposed framework is tested on a set of experimental data acquired in different environments in section V. Discussions on the performance obtained are conducted and possible perspectives are highlighted in section VI.

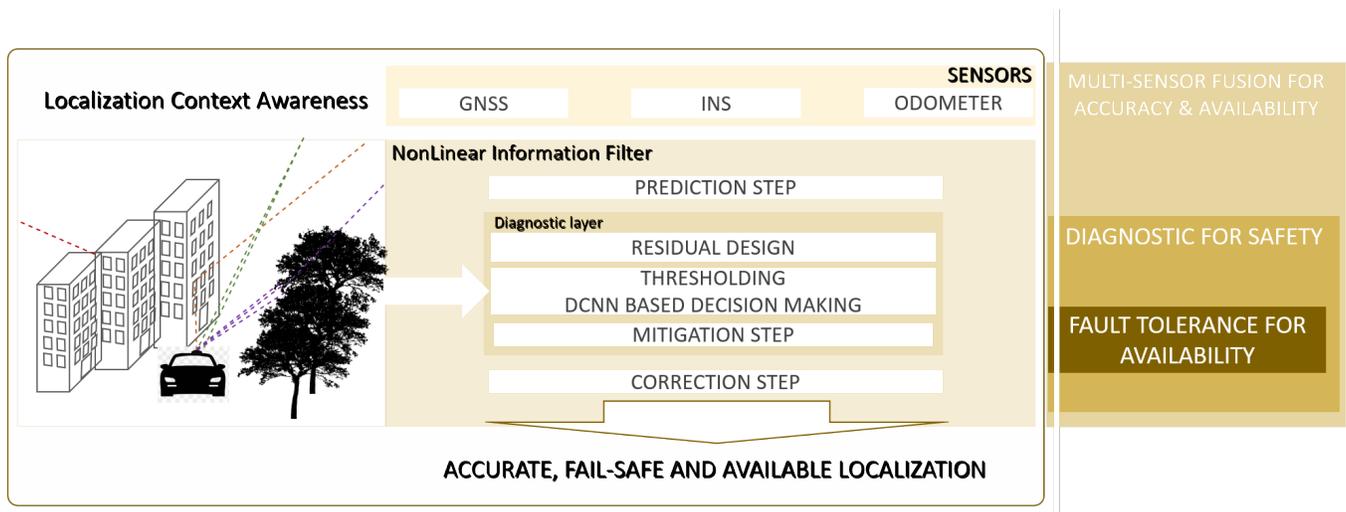


Figure 1. Global framework for an adaptive fault tolerant multi-sensor fusion

II. GNSS/IMU AND ODOMETRY TIGHT INTEGRATION THROUGH UIF FILTER

1. Multi-sensor data fusion

To reach an accurate and continuous position, This section presents a tightly coupling of multi-sensor GNSS/INS/odometer based on UIF (Unscented Information Filter) algorithm. Considering an NED (North East Down) navigation frame, the state vector for the integration consists of 17 states and can be written as:

$$X_k = [p_x \ p_y \ p_z \ V_x \ V_y \ V_z \ a_x \ a_y \ a_z \ b_{xacc} \ b_{yacc} \ b_{zacc} \ b_{xgyro} \ b_{ygyro} \ b_{zgyro} \ c\delta t \ c\dot{\delta} t]^T$$

Where, $p = [p_x, p_y, p_z]^T$ is the position given at instant k , $V = [V_x, V_y, V_z]^T$ is the velocity, $a = [\phi, \theta, \varphi]^T$ is the attitude (roll, pitch, yaw), b_{acc} is acceleration bias, b_{gyro} is gyroscope bias, c is the speed of light, and δt and $\dot{\delta} t$ denote the receiver clock bias and the corresponding clock drift respectively.

The stochastic merge filter is formed by discrete-time linear system model and measurement model described by:

$$X_{k|k-1} = F_k X_{k-1|k-1} + \nu_k \quad (1)$$

$$z_k = H_k x_k + \epsilon_k \quad (2)$$

Where:

F_k is the transition matrix, $\nu_k \sim \mathcal{N}(0, Q_k)$ is the process noise with Q_k the covariance matrix of the evolution model. The observations vector is z_k (the pseudo-range of GNSS), H_k is the observation matrix and $\epsilon_k \sim \mathcal{N}(0, R_k)$ is the observation noise with its covariance matrix R_k .

For the prediction step of the filtering algorithm, the discrete dynamic model of INS (Inertial Navigation system) is used. The position, velocity and attitude are derived from IMU measurements through the simplified mechanization equations presented as following:

$$\begin{aligned} p_{k+1} &= p_{n,k} + V_k \times T_s \\ V_{k+1} &= V_k + [R_{b2n,k} \times f_{b,k} + g_n] \times T_s \\ a_{k+1} &= a_k + E_{b2n,k} \times \omega_{b,k} \times T_s \\ b_{acc,k+1} &= b_{acc,k} + \nu_k \\ b_{gyro,k+1} &= b_{gyro,k} + \nu_k \end{aligned} \quad (3)$$

Where T_s is system propagation time interval, f_b is measurement vector of specific force, g_n represents gravity indicated in the navigation frame, which is assumed to be constant. ω_b represents the angular rate vector, R_{b2n} is the rotation matrix from body frame to NED navigation frame, and E_{b2n} is the rotation rate transformation matrix between the body and navigation frame as shown in the following equations:

$$R_{b2n} = \begin{pmatrix} c(\varphi)c(\theta) & c(\varphi)s(\theta)s(\phi) - s(\varphi)c(\phi) & c(\varphi)s(\theta)c(\phi) + s(\varphi)s(\phi) \\ s(\varphi)c(\theta) & s(\varphi)s(\theta)s(\phi) + c(\varphi)c(\phi) & s(\varphi)s(\theta)c(\phi) - c(\varphi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{pmatrix}, \quad E_{b2n} = \begin{pmatrix} 1 & s(\phi)t(\theta) & s(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)/c(\theta) & c(\phi)/c(\theta) \end{pmatrix} \quad (4)$$

where $c(x)$, $s(x)$ and $t(x)$ represent the mathematical operations of $\cos(x)$, $\sin(x)$ and $\tan(x)$ respectively. According to [17], [12] and based on equation (3) the transition matrix F_k is expressed as:

$$F_k = \left[\begin{array}{ccccc|cc} I_{3 \times 3} & I_{3 \times 3} T_s & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 2} \\ O_{3 \times 3} & I_{3 \times 3} & T_s [f_n \times] & T_s R_{b2n} & O_{3 \times 3} & O_{3 \times 2} \\ O_{3 \times 3} & O_{3 \times 3} & I_{3 \times 3} & O_{3 \times 3} & -T_s R_{b2n} & O_{3 \times 2} \\ O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & I_{3 \times 3} & O_{3 \times 3} & O_{3 \times 2} \\ O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & I_{3 \times 3} & O_{3 \times 2} \\ \hline O_{2 \times 3} & \begin{array}{c} 1 \quad T_s \\ 0 \quad 1 \end{array} \end{array} \right] \quad (6)$$

where $[f_n \times]$ is the skew-symmetric matrix of $f_n = R_{b2n} f_b$.

And the observation matrix H is composed of two sub matrices h_1 and h_2 . Where h_1 including GNSS pseudo-range and h_2

position update from additional sensor odometer. Since, the observations of the GNSS and the odometer can be considered independent (there is no spatial-temporal correlation between their covariance matrix). Hence, h_1 can be written as:

$$h_1 = \left[\begin{array}{cccccccc|cc} \nabla H_{xpr}^{x_1^s} & \nabla H_{ypr}^{y_1^s} & \nabla H_{zpr}^{z_1^s} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & 1 & 0 \\ \vdots & \\ \nabla H_{xpr}^{x_i^s} & \nabla H_{ypr}^{y_i^s} & \nabla H_{zpr}^{z_i^s} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & 1 & 0 \\ \vdots & \\ \nabla H_{xpr}^{x_m^s} & \nabla H_{ypr}^{y_m^s} & \nabla H_{zpr}^{z_m^s} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & 1 & 0 \end{array} \right] \quad (7)$$

Where

$$\begin{aligned} \nabla H_{xpr}^{x_i^s} &= \frac{\partial \rho^{x_i^s}}{\partial x} = \frac{x_i^s - \hat{x}_{pr}}{\rho^{s_i}} \\ \nabla H_{ypr}^{y_i^s} &= \frac{\partial \rho^{y_i^s}}{\partial x} = \frac{y_i^s - \hat{y}_{pr}}{\rho^{s_i}} \\ \nabla H_{zpr}^{z_i^s} &= \frac{\partial \rho^{z_i^s}}{\partial x} = \frac{z_i^s - \hat{x}_{pr}}{\rho^{s_i}} \end{aligned}$$

ρ is the GNSS pseudo-range.

In addition to GNSS pseudorange observations, measurements from odometer sensor are used considering their performance to enhance the position and mitigate IMU error. However, the expression of observation matrix h_2 of odometer can be written as:

$$h_2 = \left[\Delta_{k-1} \times \cos(\theta_{k-1} + \frac{\omega_{k-1}}{2}) \quad \Delta_{k-1} \times \sin(\theta_{k-1} + \frac{\omega_{k-1}}{2}) \quad 0 \quad O_{1 \times 3} \quad 0 \quad 0 \right] \quad (8)$$

Where Δ_{k-1} and ω_{k-1} are the elementary translation and rotation of the vehicle respectively, and θ_{k-1} is the vehicle's heading,

2. UIF pose estimation

Recently, a growing number of researchers in vehicle localization field are interested in the use of the information form of Kalman Filter in multi-sensor fusion approach since it has some distinctive characteristics. It can significantly reduce the computational load, the initialization and update step are simple and can be easily decentralized [18], [19]. Given these arguments, Unscented Information Filter (UIF) is proposed for the GNSS/INS/odometer integration. The method is based on Unscented Transformation (UT) using the sigma points in order to estimate weighted means and variances of the system state.

The set of the sigma points χ generated are seen in equations (9) and (10):

$$\begin{aligned} \chi_0 &= X, & W_0^{(m)} &= \lambda / (L + \lambda), \\ \chi_i &= X + \left(\sqrt{(L + \lambda)P} \right)_i, \quad i = 1, \dots, L & W_0^{(c)} &= \lambda / (L + \lambda) + (1 - \gamma^2 + \vartheta), \\ \chi_i &= X - \left(\sqrt{(L + \lambda)P} \right)_i, \quad i = L + 1, \dots, 2L & W_i^{(m)} &= W_i^{(c)} = 1 / (2(L + \lambda)), \quad i = 1, \dots, 2L \end{aligned} \quad (9) \quad (10)$$

Where P is the augmented state covariance matrix, $\lambda = \gamma^2(L + k) - L$, is a scaling factor to define the distance between the points and the mean, with γ and k tuning parameters. The parameter γ , $0 \leq \gamma \leq 1$ control the size of the sigma-point distribution, L is defined as the augmented state dimension, and ϑ is the third parameter for incorporating extra higher order effects [19].

- Prediction step

Based on the unscented transformation [20], the predicted information vector and matrix equations are defined as:

$$Y_{k|k-1} = [P_{k|k-1}]^{-1}, \quad (11)$$

$$y_{k|k-1} = Y_{k|k-1} \sum_{i=0}^{2n} W_i^{(m)} \chi_{i,k}. \quad (12)$$

Where the predicted state variances co-variances matrix $P_{k|k-1}$ is defined as:

$$P_{k|k-1} = \sum_{i=0}^{2n} W_i^{(c)} [\chi_{i,k} - X_{k|k-1}] [\chi_{i,k} - X_{k|k-1}]^T. \quad (13)$$

- Update step

The updated information matrix $Y_{k|k}$ and vector $y_{k|k}$ are respectively represented as:

$$Y_{k|k} = Y_{k|k-1} + \sum_{i=1}^N gI_i(k), \quad (14)$$

$$y_{k|k} = y_{k|k-1} + \sum_{i=1}^N pI_i(k). \quad (15)$$

where the informational contributions $gI_i(k)$ and $pI_i(k)$ are defined as:

$$gI_i(k) = H_{i,k}^T R_i^{-1}(k) H_{i,k}, \quad (16)$$

$$pI_i(k) = H_{i,k}^T R_i^{-1}(k) [(Z_{i,k} - \hat{Z}_{i,k}) + H_{i,k} X_{k|k-1}]. \quad (17)$$

N is the number of observations at instant k .

III. DIAGNOSIS LAYER BASED ON α -RD RESIDUAL GENERATION

1. Residual generation

In order to generate sensitive residuals able to detect most of faults, several methods exist in the literature. Among these methods, we found the information divergence. However, the choice of the divergence can have a significant impact on the performance of the diagnostic layer mainly in the type of fault to be detected. In this work, Rényi Divergence the generalization of special cases of well-known divergences including Kullback Leibler Divergence, Bhattacharyya distance is used. According to [21] and [22] the α -RD between two probability distributions P and Q is defined as:

$$RD_\alpha(P||Q) = \frac{1}{\alpha - 1} \ln \int P^\alpha(x) Q^{1-\alpha}(x) dx \quad (18)$$

Where $\alpha \in R - \{1\}$.

For a fixed value of α ($\alpha=0.8$), the α -RD residual is defined through two multivariate Gaussian densities, $g(k|k-1)$ for the prediction step and $g(k|k)$ for the correction step and can be defined as:

$$RD_\alpha(g(k|k)||g(k|k-1)) = \frac{\alpha}{2} (X_{k|k} - X_{k|k-1})^T \left(\sum_{\alpha} \right)^{-1} (X_{k|k} - X_{k|k-1}) - \frac{1}{2(\alpha - 1)} \log \frac{|\sum_{\alpha}|}{|\sum_{k|k}|^{1-\alpha} |\sum_{k|k-1}|^{\alpha}} \quad (19)$$

Where $\sum_{\alpha} = \alpha \sum_{k|k-1} + (1 - \alpha) \sum_{k|k}$, $\sum_{k|k-1} = \frac{1}{Y_{k|k-1}}$ and $\sum_{k|k} = \frac{1}{Y_{k|k}}$

2. Residual statistical characterization and threshold setting

After performing several vehicle trajectories for the sensor database collection in different types of environments and conditions. After using the database in the generation of residuals, a threshold able to make the right decision for faulty and non-faulty cases is needed. Thus, two probability density functions (*pdfs*) of α -RD residual represent the faulty and the non faulty cases are created. For this goal, the classification of *pdfs* is proposed through an unsupervised learning algorithm called the Expectation Maximization (EM) as a solution [23]. This method switches between two main steps, the first step called the E-step consists of approximating a probability distribution over the missing data according to the current model, and the second step the M-step is based on the re-estimation of the model parameters. The result of the clustering algorithm is illustrated in figure 2 which shows the two classes of non-faulty (green) and faulty (purple) distributions. Regarding the faulty *pdf*, the most challenging faults to detect and analyze are those who belong to *pdf* sharing an overlap with the fault-free case, and especially when it comes to setting a threshold which represents the following step. In order to evaluate the residual, an appropriate threshold is defined in the aim to decide efficiently about fault(s) in case of their existence(s).

Assuming that we dispose of two hypotheses, in the case of no fault in the sensor(s) measurement(s) thus null hypothesis H_0 , otherwise the alternative hypothesis H_1 , where the probabilities of making the decision u associated to these assumptions are defined as following:

- the probability of detection: the probability of deciding u_1 when H_1 is true: $P_D = p(u_1/H_1)$
- the probability of non detection: the probability of deciding u_0 when H_0 is true: $P_{ND} = p(u_0/H_0)$
- the probability of false alarm: the probability of deciding u_1 when H_0 is true: $P_{fa} = p(u_1/H_0)$
- the probability of missed detection: the probability of deciding u_0 when H_1 is true: $P_{md} = p(u_0/H_1)$

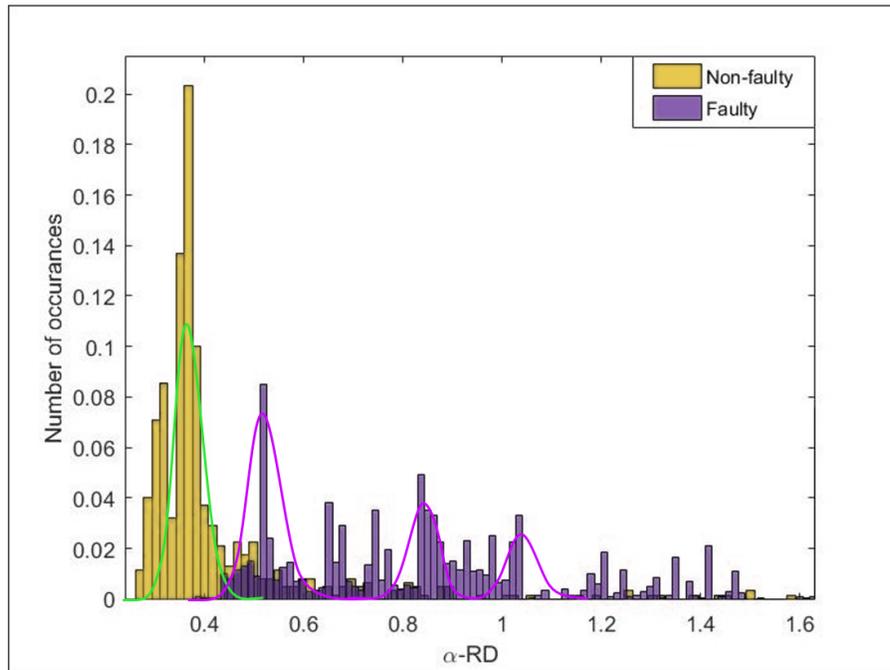


Figure 2. EM (Expectation Maximization) algorithm solution for *pdfs* (Probability density functions) separation of faulty (purple) and non faulty measurements (green)

The policy for the threshold optimization function is to ensure an optimal compromise between the operational requirements mentioned earlier in the introduction, which may be antagonistic and related to different parameters, improving one of these requirements may have a negative impact on others. Generally, the calculation of threshold is based on two the probabilities of

false alarm P_{fa} and missed detection P_{md} [24]. By minimizing the probability P_{fa} , the availability of the system is improved but this can affect the operational safety since the probability of missed detection P_{md} is not taken into account. Therefore, an optimal thresholding strategy is expected to take into account the two probabilities to ensure the best trade-off between operational requirements. To reach this aim, a risk function that minimizes the last two probabilities is defined. Since the cost attributed to the decisions which is contrary to the hypothesis is higher than the cost attributed to the decision in agreement with the hypothesis. To facilitate the optimization step, a log scale is applied using the divergence used to design the residual:

$$Rc_\alpha = [RD_\alpha(p(H_i/u_j)||p(H_i)) + RD_\alpha(p(H_j/u_i)||p(H_j))] \quad (20)$$

With $i \neq j$

Which leads to :

$$Rc_\alpha = \frac{1}{\alpha - 1} \left[\log \left(\frac{p(u_1/H_0)}{p(u_1)} \right)^\alpha p(H_0) + \log \left(\frac{p(u_0/H_1)}{p(u_0)} \right)^\alpha p(H_1) \right] \quad (21)$$

Thus, the α -Rényi criterion is written as:

$$\begin{aligned} Rc_\alpha &= \frac{1}{\alpha - 1} \log [P_0(1 - P_0)] \\ &+ \frac{1}{\alpha - 1} \log [P_0\gamma_1\gamma_0 + (1 - P_0)\gamma_1\beta_0]^\alpha + \frac{1}{\alpha - 1} \log [P_0\gamma_1\beta_0 + (1 - P_0)\beta_1\beta_0]^\alpha \\ &- \frac{\alpha}{\alpha - 1} \log [P_0\gamma_0 + (1 - P_0)\beta_0] - \frac{\alpha}{\alpha - 1} \log [P_0\gamma_1 + (1 - P_0)\beta_1] \end{aligned} \quad (22)$$

where: $\gamma_0 = 1 - P_{fa}$, $\beta_0 = P_{md} = 1 - P_D$, $\gamma_1 = P_{fa}$, $\beta_1 = P_D$ and $P_0 = p(H_0)$.

It was also proved through the derivation made in the study [5] that the maximum of the risk function corresponds to the optimal threshold. However, one of the key elements for the determination of the threshold is the probability of not facing fault(s) P_0 as mentioned in the introduction, generally this value is calculated based on MLE algorithm [16]. It is important to note that to make a decision this method relies on a time window of the last decisions made. This represents one of the limitations that makes it unsuitable for dealing with all types of errors, especially the most threatening ones such as sudden GNSS errors. Therefore, in the next section, we introduce a deep learning based approach for the computation of P_0 that can address this type of error(s).

IV. ADAPTIVE DECISION AND FAULT MANAGEMENT BASED ON DCNN

Over the last years, Deep learning techniques have shown remarkable performance in various fields such as computer vision, natural language processing, speech recognition and localization [25]. In localization field, Deep Convolution Neural Network is one of the most employed models for environment classification [26], GNSS jamming detection [27], traffic prediction [28], etc. What makes this network more attractive than other types of networks such as DNNs, is its potential to exploit spatial or temporal correlation by extracting useful information from input data (2D images, voice signal, sensors measurements...) and learn distinctive features in order to match as precisely as possible the inputs with the outputs. For these reasons, figure 3 propose the integration of DCNN in thresholding step for regression task. The goal of the DCNN is the estimation of the probability of not facing to fault(s) called P_0 at each instant to mitigate sudden GNSS errors in crossed environment. In a first step, an offline procedure is launched which consists in training the input data corresponding to the data derived from GNSS signals and fusion process with several CNN models. Then, a validation step comes to evaluate the training model and at last a testing step is proposed for the evaluation of the final selected model. After choosing the suitable model that best matches the inputs with the output corresponding to labeled P_0 , the online process in detection step can start, the final result of the estimation of the probability P_0 is integrated in the last phase of the detection step in the thresholding Risk function in the purpose to adapt the decision to the navigation context. Thereafter, the decision in case of presence of fault(s) or not is performed and fault isolation step is directly activated. However, in order to establish an effective data representation and to establish a link between the inputs and output of a neural network, the choice of inputs and the way in which the output is labeled is as important as the topology of the network. For this reason, in the following sections the identification of the input data, the manner of labeling the output and the topology of the network are discussed.

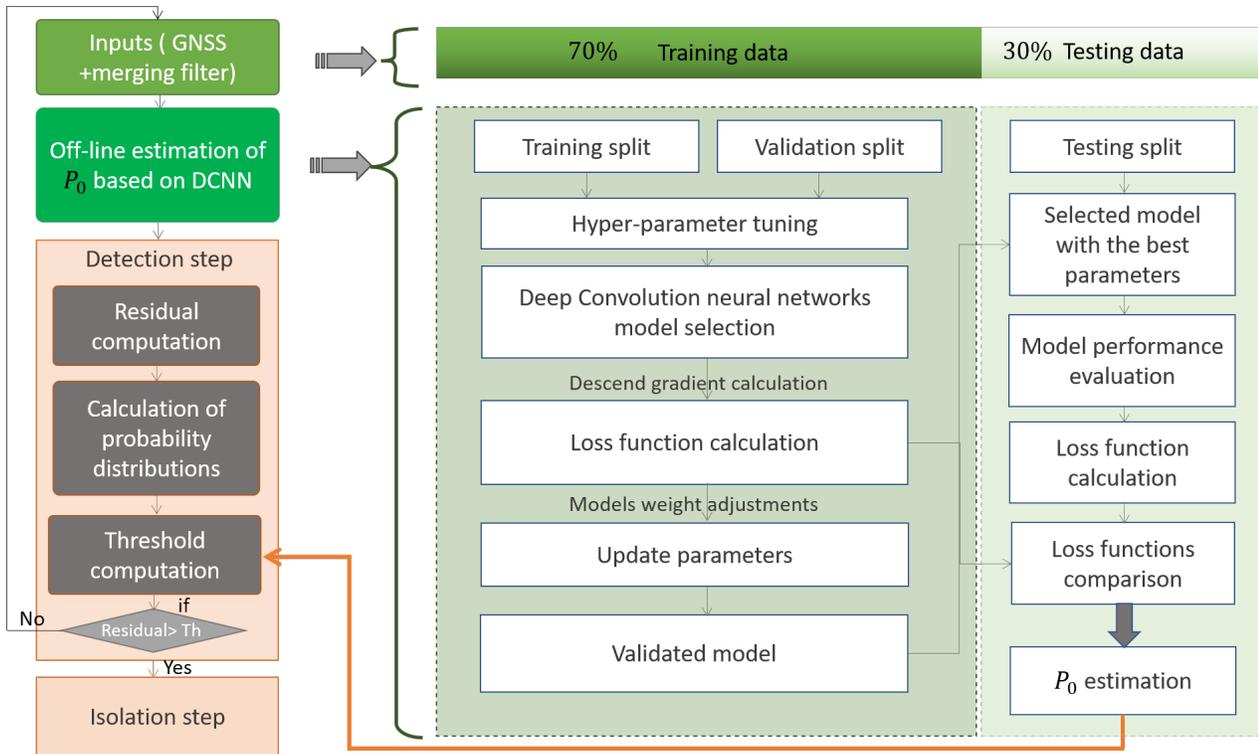


Figure 3. Overall scheme of the integration of deep learning process used in the FDI approach

1. Inputs identification

The emergence of deep learning in the field of GNSS has changed the way of interpreting and solving problems for localization and navigation. One of the main challenges of using deep learning is the selection of relevant features. Neural networks algorithm requires a variety of inputs hard to conceive when it concerns the analysis of a GNSS signal and the effect of environmental changing on it. In this context, various researches have been investigated to improve the performance of GNSS systems based on this tool for different problems as NLOS/multipath signal detection, detection of GNSS ionospheric, spoofing attacks and jammer classification. Most of the solutions for features extraction are based on one or even two of GNSS signal observations such as elevation angle [29], SNR (Signal to Noise Ratio) [30] or a combination of both [31], CN/0 (Carrier-to-Noise Ratio) [32], DOP (Dilution Of Precision) [33], as they can be considered as indicators of the performance of a measurement. Nevertheless, if the combination of these different features allows to provide reliable measurements and a prior knowledge of the current situation, the logic of combining more than three features becomes complex and difficult to apprehend.

In this study, relevant features are defined from the characteristic information of the abnormal GNSS signals that allow to have a set of a priori knowledge on the signal state and on the localization context. These features mainly contain satellite information such as satellite elevations, signal propagation information (CN/0) which showed high efficiency for classification of LOS and NLOS signals from in dense urban environment. In addition, other features that contain valuable information about the quality of GNSS observations such as CMC (code-minus-carrier), DOP. Therefore, the input layer exhibits the seven following features: number of satellites as they affect the accuracy of the position, the geometry configuration analysis of visible satellites with HDOP (Horizontal DOP), CMC, satellite elevation angle, CN/0, and historical information of the merging filter contained in variance-covariance matrices of the prediction step P_{Pred} , and correction step P_{Corr} .

2. P_0 labeling

In this section, the output labeling representing the main commonality between supervised deep learning algorithms is discussed. During training, to ensure proper optimization of deep learning models, a certain amount of labeled data is required. However, as mentioned earlier, generating large sensor dataset with annotations of a probability of not facing a fault P_0 each instant considering cross environment conditions is a complex task. To effectively define this probability, expertise and experience in GNSS signaling is required. The main challenge is to guarantee at each instant the presence or absence of fault(s) in order to fix this probability with a high precision. Generally, fault(s) labellisation is done through a static experiment in each environment (buildings, trees, open sky) where a GNSS receiver is deployed to collect raw data for a minimum 24 hours. This method can be efficient but it has certain limitations as time consumption and implicit signal transmission which requires a 3d map [34]. In this

study, to define the value of the probability to not facing fault P_0 , FDI coupled with PPP (precise point positioning) algorithms are used after they proved their efficiency giving high detectability and low probability of false alarm. Thus, the labeling of the output is performed according to the rate of detection of faults in the following manner. At the beginning the value of P_0 is maximized at 0.8, after each detection of a faulty measurement it begins to lower the value of P_0 by 0.2 until it reaches 0.2.

3. Deep Convolution Neural Networks topology

The proposed architecture is one dimension DCNN for the estimation of the probability of not facing a fault P_0 is illustrated in figure 4. It includes 9 layers including input layer, 2 convolutional layers, one max pooling layer, one flattening layer, 3 fully connected dense layers and an output layer for the estimation of P_0 . The input layer is composed of 114×4112 input size representing information on the status of the GNSS signal at each instant with various variables such as the elevation of satellites, DOP, CMC, etc. The following layers represent the main block of construction of a convolutional neural network (CNN) the "2 convolutional layers". Inspired biologically by the processing of the visual cortex of mammals, each CNN layer is composed of 128 kernels or filters with a window size of 6. Particularly different from classical neural networks which are fully connected layers to each other, the CNN architecture is based on the so-called convolution operation, each neuron of the feature map receives connections only from a subset of neurons from the previous layer which will allow first to extract a local correlation of the input data and then to extract the inherent features of the previous CNN layer. During this process, a new feature map is generated at each time by the convolution between the input feature maps and the kernels then a function is applied to these convolutional operations. Thus, the output feature map of each layer is calculated as [35]:

$$y_j^i = f\left(\sum_{l \in M_j} y_l^{i-1} * k_j^i + b_j^i\right) \quad (23)$$

Where y_j^i represents the feature map output by the j th convolutional kernel at the i th layer; M_j represents all the feature maps at the $i-1$ layer; k_j^i represents a convolutional kernel at the i th layer; b_j^i represents the bias corresponding to the y_j^i features at the i th layer; $f()$ represents the operation function; and $*$ represents the convolution operation. Then to reduce the features dimension a max pooling layer is implemented to extract to highest values (252 most active features) from the last convolutional layer [36], the extracted features learned from are merged into one single vector consisting of 252×128 numbers, followed by 3 fully-connected hidden layers consisting of with 128 nodes each one. The last layer is connected to the output layer which contain 4112 nodes with ReLU (Rectified Linear) as an activation function as the topology developed is for regression approach [37]. For the optimization of the neural networks a backpropagation algorithm is used, by computing the Stochastic Gradient Descent (SGD) which updates the neurons weights using a feedback connections to deliver error.

To summarize, the process of learning is performed in 4 steps. Step1: collecting data from the sensors, inputs selection and output labellisation. Step2: splitting the data into two groups of data "train set" composed of approximately 70% of the data set, among them there are 10 % for the "validation set" and 30 % of the data for the "test set". Step3: Start the training process which consists of two groups of data "training set" and "validation set". The role of the first group is to establish a relationship (or a complex function) between the input parameters of the sensors and the output, which is the probability not facing a fault P_0 , in order to synthesize a functional relationship between them to develop models for estimating P_0 . The second group represents the part of the training data called the "validation set" which evaluate the performance of the trained models and validate the most appropriate one. Finally, the selected trained model will be evaluated in the last step "the estimation step" by the testing set which consists of data that the trained model has never seen before.

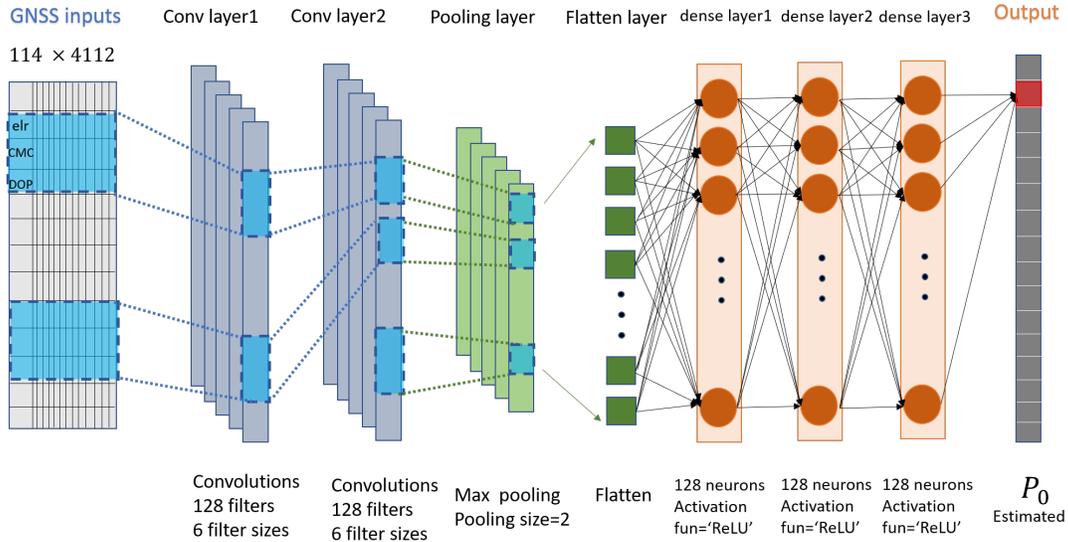


Figure 4. A graphical illustration of deep convolution neural network topology

V. EXPERIMENTAL RESULTS

This section presents the experimental results to evaluate the effectiveness of the proposed method and to validate the whole concept. It consists of two parts. the first part is dedicated to the validation of the choice of the DCNN by comparing it with the MLE method and the second part contains an impact analysis of this tool on the fault detection and isolation method and on the GNSS-based position solution.

1. Data collection

Six real trajectories are used for the experimental concept validation. Figure 5 show the real 2D view, using Google Earth software, and table 1 details the information for all the used trajectories. The data are collected using a robotized Renault ZOE, the experimental vehicle of CRISAL laboratory. The vehicle is equipped with Novatel PwrPak7D merging GNSS, INS and RTCM messages in order to create ground truth references. Tests have been carried out using real GNSS raw measurements with Ublox M8P coupled with Inertial Labs OS3D-FG IMU raw data and vehicle internal valuable information such as steering angles and odometer data via the CAN bus. All these data were acquired via a ROS platform ensuring a software based GPS synchronization. The collected data are split into two sets: train and test data sets. Where C1, C2, C4, C5 and C6 are used for the learning phase equal to 16,196 km. While C3 is used for testing phase phase equal to 9,844km in order to evaluate the training model performance.

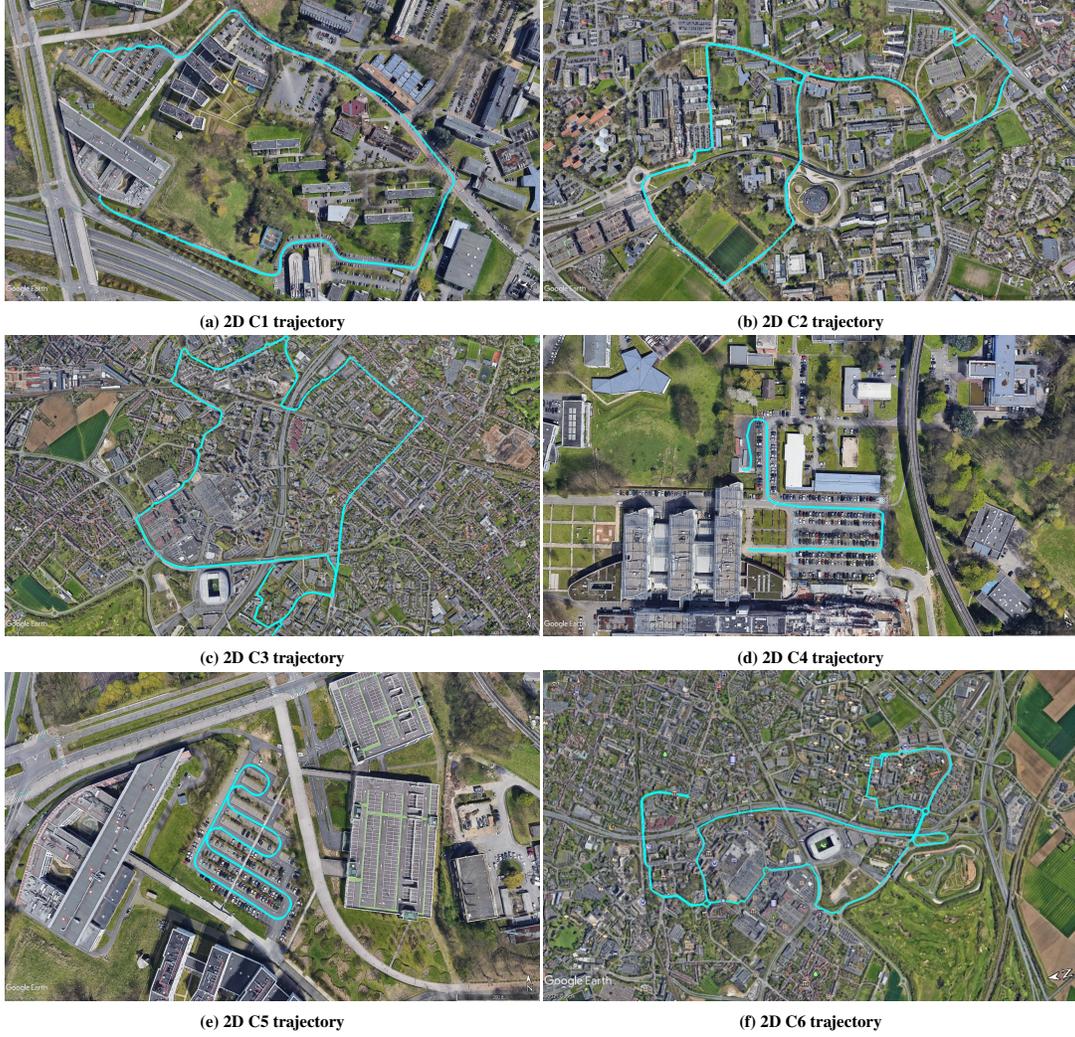


Figure 5. Trajectories references

Table 1: Data acquisition information for trajectories

Trajectory name	Trajectory length
C1	1134.2 meters
C2	3444.6 meters
C3	9844.71 meters
C4	348.85 meters
C5	467.45 meters
C6	10803 meters

2. Results for FDI validation approach

Having defined the Fault Detection and Isolation process previously. The first step is to analyse the global residual behavior before the implementation of FDI process. Figure 6 shows the behavior of α -Rényi residual before the implementation of the FDI for the testing trajectory *C3*, given the peaks and the high variation of this curve it confirms the existence of faults but it gives no indication on the type of fault(s) and which of the fault(s) should be isolated and which not.

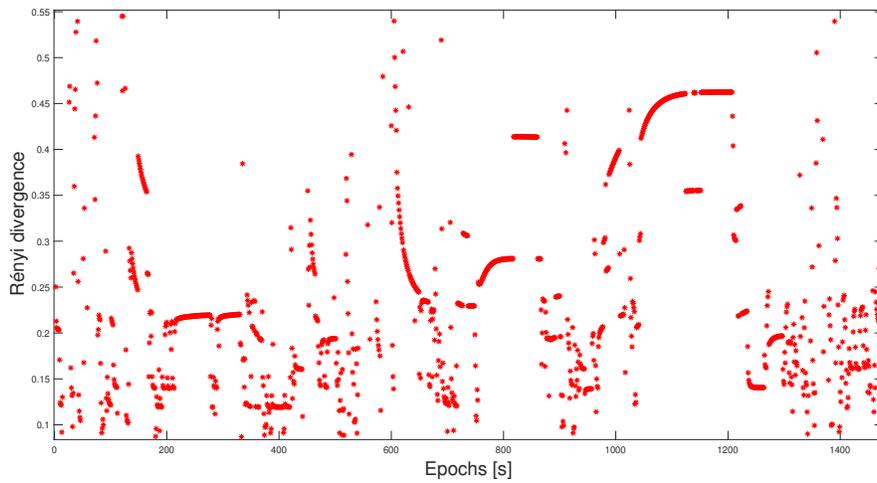


Figure 6. Rényi without FDI

To explore more deeply, there is a need to analyze the variation of the parameters related to the GNSS signal state such as the residual of each available satellite at the time of the experimental tests. Figure 7 presents the individual residual for each satellite. The residual of the satellite 7, 8, 21 and 26 most of the time show a larger variation than the others which can be considered as a fault.

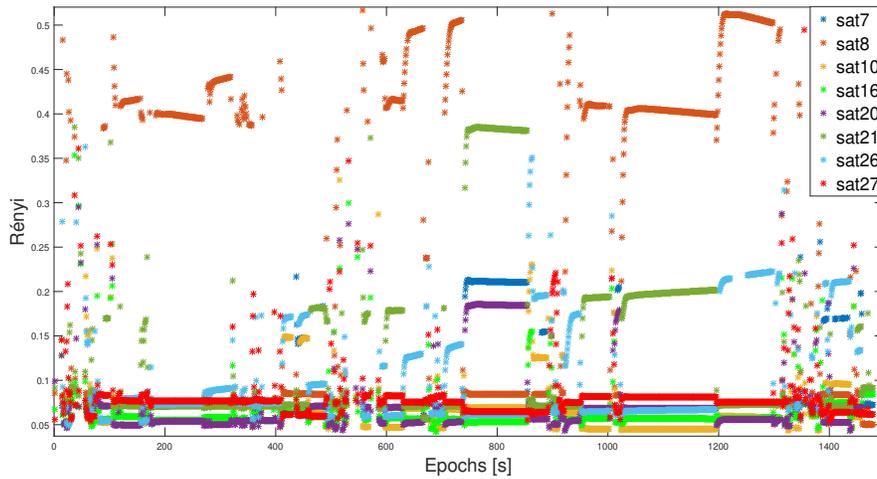


Figure 7. Rényi satellites gain

The abrupt variation of the residuals of the satellites involved can be caused by the decrease of the elevation of these satellites. From the figure 8, we can see that most of these satellites have an elevation below 30° which explains in a way the behavior of their residual.

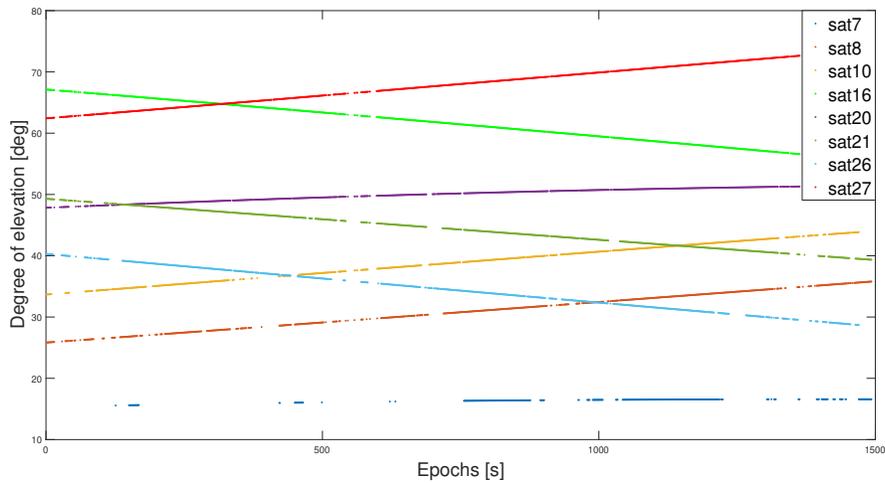


Figure 8. Satellites elevation

After applying the FDI method, it is clear that the residual has decreased in magnitude and has a regular behavior. Figure 9 show the effect of the integration of FDI method on the α -Rényi residual.

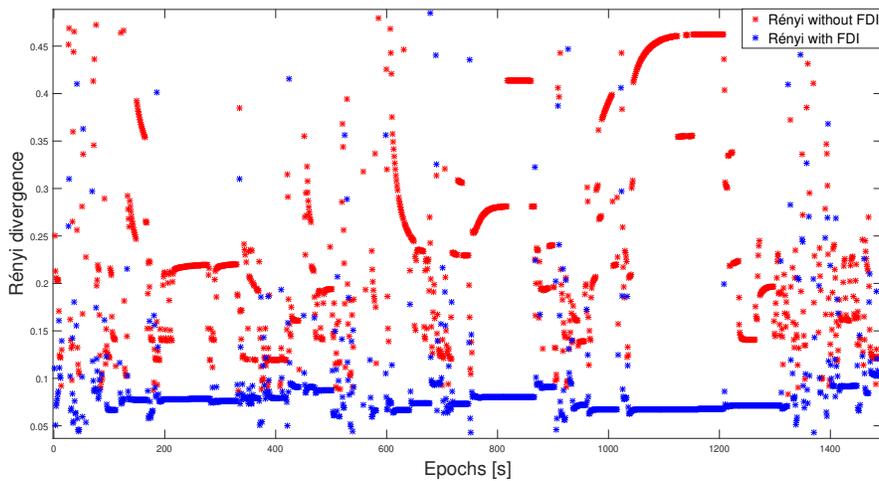


Figure 9. Rényi without and with FDI

3. P_0 estimation

In order to validate the performance of DCNN model, it needs to be compared to a standard regression deep learning method in term of Loss function (error calculation). For the comparison, classic feed-forward neural network called Multilayer Perceptron (MLP) is used. It's composed of 3 dense fully connected layers with 128 nodes each layer with back propagation algorithm. To compare the Losses of these topologies fairly, the same batches size 250 and iterations 500 are used for the learning process of the two networks. In order to reduce the error between the estimated value and the labeled one, two main metrics are applied in regression problems Mean Square Error (MSE) and Mean Absolute Error (MAE) the results of their estimation are presented in table, the lowest value of the loss is attributed to DCNN with MAE loss function. Figure 10 show the results of loss function of the two typologies DCNN and MLP. We notice easily that the two losses of train and test follow each other, which means there is roughly no overfitting or underfitting. However, the Loss of the DCNN topology has a rather smooth pace and tends to converge quickly than the MLP until 0.01 of error.

Table 2: DNN Models performances Results

Topology	Metric	Loss result
MLP	MSE	0.0524
DCNN	MAE	0.0183

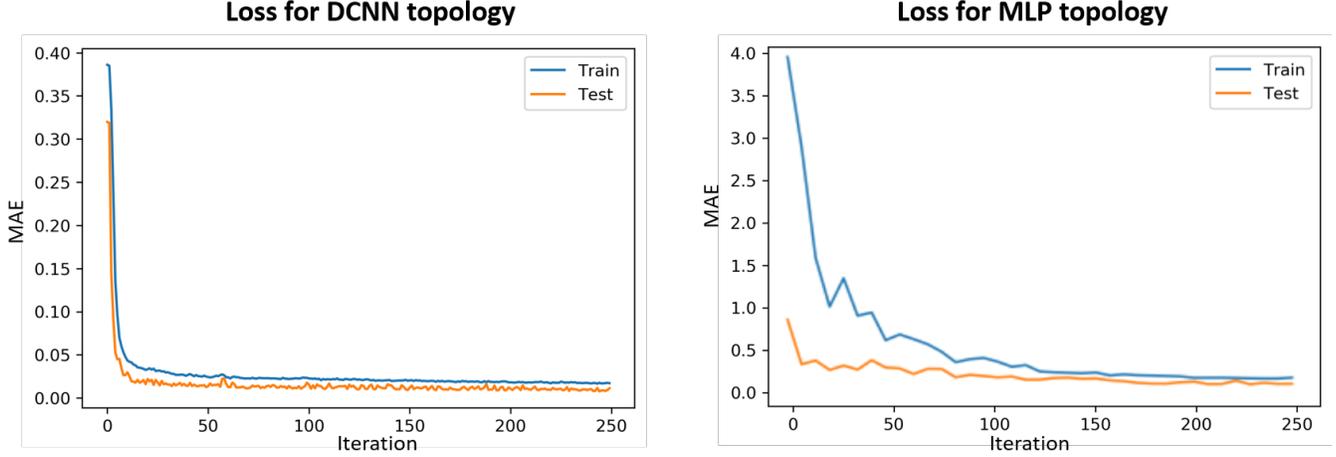
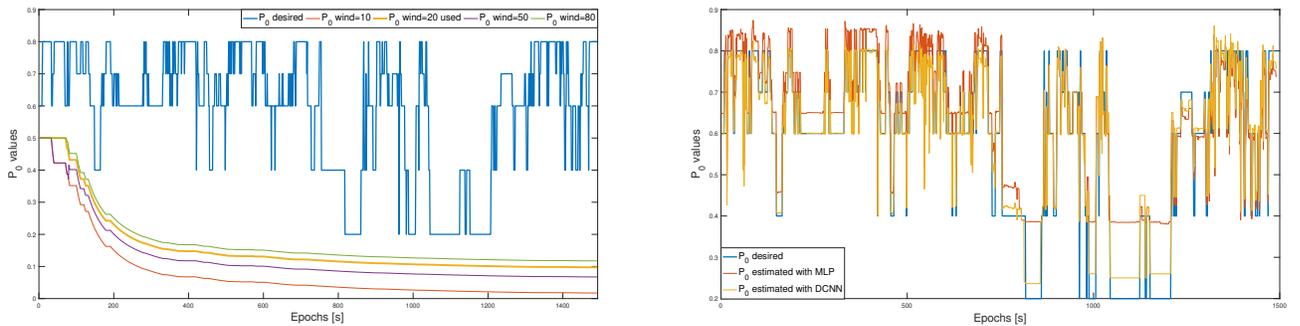


Figure 10. Loss for training and testing data set

After the validation of the selected model, the evaluation of the P_0 estimation proposed by the two approaches DCNN and MLE is launched. Since the MLE method is based on a history of previous decisions, several sizes of time window were performed, in the following comparative results we have selected the size 20. Figure 11a present the comparison between the labeled P_0 (desired P_0) estimation and the estimated one with different time window size. We note that the variation of P_0 using MLE method is very low compared to that of the labeled P_0 and the more we increase the size of the window the more the curve tends to have a filtering aspect. In figure 11 the result of comparison between labeled P_0 and the estimation of P_0 using and DCNN and MLP algorithm is shown. It is clear that the estimation of P_0 based on DCNN network coincides with labeled P_0 and outperformed the MLP estimation.

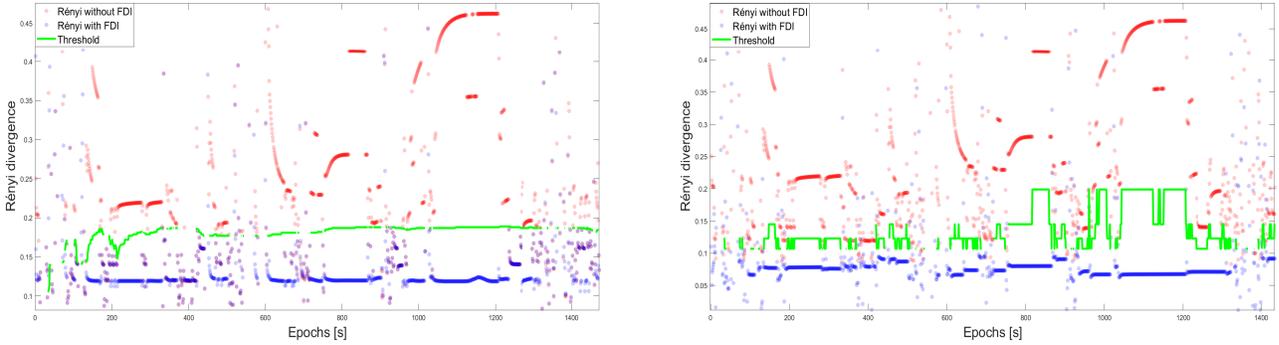


(a) Comparison of P_0 estimation using MLE method with different time window with the desired P_0 values, in blue: the desired P_0 values, in orange: P_0 estimation based on MLE for time window size= 10s, in yellow: P_0 estimation based on MLE for time window size= 20s and the used one for the rest of the study, in purple: P_0 estimation based on MLE for time window size= 50s, purple: P_0 estimation based on MLE for time window size= 80s

(b) Comparison of P_0 estimation based on different neural networks topology with the desired P_0 values, in blue: the desired P_0 values, in orange: P_0 estimation based on Multilayer perceptron (MLP) and in yellow: P_0 estimation based on Deep Convolution neural networks (DCNN)

Figure 11. Testing trajectories P_0 estimation performance

In figure 12, the impact of P_0 estimation on threshold calculation is presented. In figure 12a, it can be seen that the low variation of P_0 estimated by MLE has well affected the threshold variation. On the other hand on the figure 12b the threshold based on the estimation of P_0 with DCNN shows more dynamics with a much larger variation which is mainly due to different decision making during the experimental tests.



(a) Rényi divergence without and with FDI with threshold based on MLE (b) Rényi divergence without and with FDI with adaptive threshold based on DNN

Figure 12. Impact of P_0 estimation on threshold selection

In order to justify the behavior of the threshold using the two methods. A deep decision analysis is established through the isolated satellites using the two methods. Figure 13 show the isolated satellites in different colors for each instant based on MLE in figure 13a and DCNN in figure 13b. For a first time seen there are some satellites that have been isolated by one method and not in the other. However, to highlight the difference more effectively figure 14 show only the difference between the isolated satellites in the two approaches. Both methods represent several isolated satellites in commune but in some range differ, a first area of difference is highlighted in pink from 598 to 660 seconds where the decision based DCNN approach represented by the black dot isolate the satellite 21 while the threshold based on MLE method has decided to isolate the 26 during this time frame. In addition, a second outlined zone in blue represents another kind of difference from 1398 to 1423 seconds during all this period no satellites were isolated by the MLE method while the DCNN method decision was totally different by isolating 4 satellites.

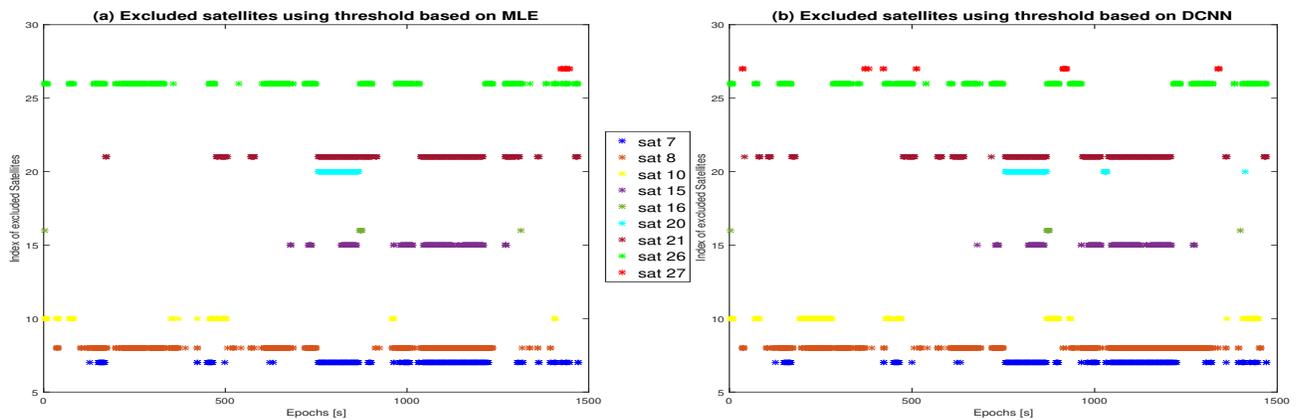


Figure 13. Impact of estimated P_0 on decision making

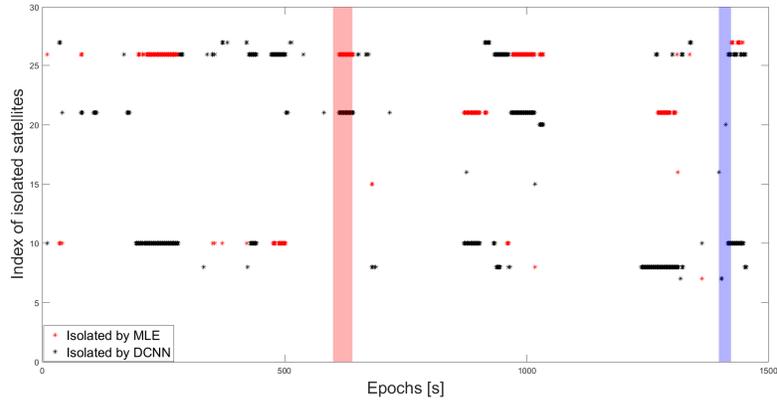


Figure 14. Difference in decision making using P_0 estimated with MLE and DNN

Having defined the sources of fault detection and isolation. We now focus on providing a measure of the system performance to the users that allows to validate the proposed approach, a Relative Performance Indicator (RPI) is performed. In order to compare the improvement in term of position of the two approaches, the difference between position errors of each method according to the reference is calculated based on the following equation:

$$RPI = RMSE(X_{MLE}, X_{GroundTruth}) - RMSE(X_{DCNN}, X_{GroundTruth}) \quad (24)$$

Based on the previous equation 24 and figure 15, when the value is equal, and thus the relative performance indicator is equal to zero, both approaches make the same decision(s) and provide the same position error. However, the positive side of the curve (blue color) represent the good decisions taken by DCNN method, where in the opposite direction the red curve represent the appropriate decisions taken by MLE algorithm to improve the the position solution. We can notice clearly that the proposed approach achieves better performance in positioning than MLE algorithm by removing larger errors on the majority of the trajectory.

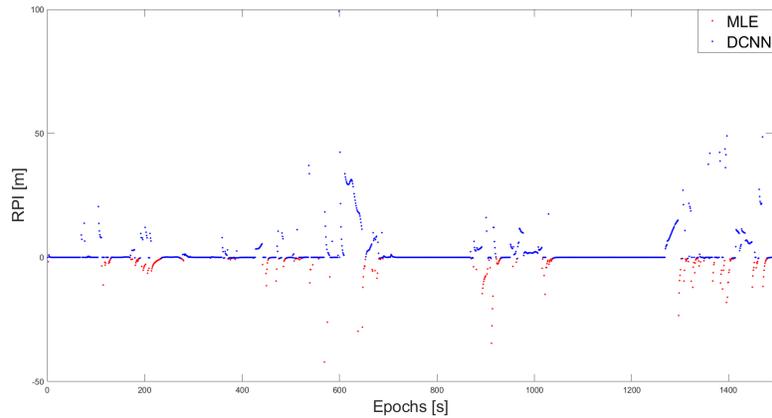


Figure 15. RPI

For numerical evaluation of the performance of the two methods, we calculate the mean and the max error detected by each approach. The results are presented in Table 3.

Table 3: Evaluation methods performances using errors metrics

Error metric [m]	Removed only by DCNN	Removed only by MLE
Mean error	5.581	2.315
Max error	100	48

VI. CONCLUSION

In this paper, an adaptive fault tolerant fusion framework based on DCNN decision making was developed. We aimed to deal with sudden GNSS errors caused by changing environment. For fusion process, the complementarity of the sensors has been exploited to increase the availability of the system. However, for the diagnostic part, a divergence called α -RD that generalized known informational divergences has been applied for the generation of the residual. For threshold calculation, we investigated the integration of Deep Convolution Neural Networks (DCNN) to calculate the probability of non faulty cases called P_0 . The proposed approach was compared with an existing method based on time window the Maximum Likelihood estimation (MLE). However, for the selection of the appropriate neural network, we developed and compared different learning models and topology with different metrics of evaluation (MAE, MSE). The proposed method solved the problem of time windowing of the MLE algorithm. The results of DCNN have shown their feasibility on the position estimation with high accuracy (until 100m of max removed error). Future work includes investigating the efficiency of deep learning in the selection of α parameter of α -RD and the application of the proposed method on multi-vehicles in real time.

ACKNOWLEDGEMENTS

The present work benefits from the financial support of the ANR French Research Agency within the project LOCSP No 2019-CE22-0011.

REFERENCES

- [1] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.
- [2] S. Tay and J. Marais, "Weighting models for gps pseudorange observations for land transportation in urban canyons," in *6th European workshop on GNSS signals and signal processing*, 2013, p. 4p.
- [3] A. Angrisano *et al.*, "Gnss/ins integration methods," *Dottorato di ricerca (PhD) in Scienze Geodetiche e Topografiche Thesis, Universita' degli Studi di Napoli PARTHENOPE, Naples*, vol. 21, 2010.
- [4] A. Broumandan and G. Lachapelle, "Spoofing detection using gnss/ins/odometer coupling for vehicular navigation," *Sensors*, vol. 18, no. 5, p. 1305, 2018.
- [5] K. Makkawi, N. Ait-Tmazirte, M. El Badaoui El Najjar, and N. Moubayed, "Adaptive diagnosis for fault tolerant data fusion based on α -rényi divergence strategy for vehicle localization," *Entropy*, vol. 23, no. 4, p. 463, 2021.
- [6] X. Han, S. A. Kazim, N. A. Tmazirte, J. Marais, and D. Lu, "Gnss/imu tightly coupled scheme with weighting and fde for rail applications," in *Proceedings of the 2020 International Technical Meeting of The Institute of Navigation*, 2020, pp. 570–583.
- [7] Y. Gao, Y. Gao, B. Liu, and Y. Jiang, "Enhanced fault detection and exclusion based on kalman filter with colored measurement noise and application to rtk," *GPS Solutions*, vol. 25, no. 3, pp. 1–13, 2021.
- [8] P. D. Groves, Z. Jiang, L. Wang, and M. K. Ziebart, "Intelligent urban positioning using multi-constellation gnss with 3d mapping and nlos signal detection," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, 2012, pp. 458–472.
- [9] D. Betaille, S. Miquel, F. Godan, and F. Peyret, "3d-city-model-aided gnss accurate positioning with integrity provision using simplified geometry of buildings," in *European Navigation Conference 2015*, 2015, p. 6p.
- [10] M. Tossaint, J. Samson, F. Toran, J. Ventura-Traveset, M. Hernández-Pajares, J. Juan, J. Sanz, and P. Ramos-Bosch, "The stanford–esa integrity diagram: A new tool for the user domain sbas integrity assessment," *Navigation*, vol. 54, no. 2, pp. 153–162, 2007.
- [11] C. Stallo, A. Neri, P. Salvatori, A. Coluccia, R. Capua, G. Olivieri, L. Gattuso, L. Bonenberg, T. Moore, and F. Rispoli, "Gnss-based location determination system architecture for railway performance assessment in presence of local effects," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2018, pp. 374–381.
- [12] P. D. Groves, "Principles of gnss, inertial, and multisensor integrated navigation systems, [book review]," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 2, pp. 26–27, 2015.
- [13] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4670–4677.

- [14] Q. Li, J. P. Queralta, T. N. Gia, Z. Zou, and T. Westerlund, "Multi-sensor fusion for navigation and mapping in autonomous vehicles: Accurate localization in urban environments," *Unmanned Systems*, vol. 8, no. 03, pp. 229–237, 2020.
- [15] H. Gao and P. D. Groves, "Environmental context detection for adaptive navigation using gnss measurements from a smartphone," *NAVIGATION, Journal of the Institute of Navigation*, vol. 65, no. 1, pp. 99–116, 2018.
- [16] J. Al Hage, M. E. El Najjar, and D. Pomorski, "Multi-sensor fusion approach with fault detection and exclusion based on the kullback–leibler divergence: Application on collaborative multi-robot system," *Information Fusion*, vol. 37, pp. 61–76, 2017.
- [17] M. G. Petovello, *Real-time integration of a tactical-grade IMU and GPS for high-accuracy positioning and navigation*. Citeseer, 2003.
- [18] D.-J. Lee, "Nonlinear estimation and multiple sensor fusion using unscented information filtering," *IEEE Signal Processing Letters*, vol. 15, pp. 861–864, 2008.
- [19] L. Wei, C. Cappelle, and Y. Ruichek, "Camera/laser/gps fusion method for vehicle positioning under extended nis-based sensor validation," *IEEE transactions on Instrumentation and Measurement*, vol. 62, no. 11, pp. 3110–3122, 2013.
- [20] R. Van Der Merwe, E. Wan, and S. Julier, "Sigma-point kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 5120.
- [21] T. Hobza, D. Morales, and L. Pardo, "Rényi statistics for testing equality of autocorrelation coefficients," *Statistical Methodology*, vol. 6, no. 4, pp. 424–436, 2009.
- [22] M. Khoder, A.-T. Nouridine, M. Nazih *et al.*, "Fault tolerant multi-sensor data fusion for vehicle localisation using maximum correntropy unscented information filter and α -rényi divergence," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. IEEE, 2020, pp. 1–8.
- [23] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal processing magazine*, vol. 13, no. 6, pp. 47–60, 1996.
- [24] L. Li, Z. Li, H. Yuan, L. Wang, and Y. Hou, "Integrity monitoring-based ratio test for gnss integer ambiguity validation," *GPS solutions*, vol. 20, no. 3, pp. 573–585, 2016.
- [25] J. Ahmad, H. Farman, and Z. Jan, "Deep learning methods and applications," in *Deep Learning: Convergence to Big Data Analytics*. Springer, 2019, pp. 31–42.
- [26] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018.
- [27] Z. Wu, Y. Zhao, Z. Yin, and H. Luo, "Jamming signals classification using convolutional neural network," in *2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2017, pp. 062–067.
- [28] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.
- [29] R. Yozevitch, B. B. Moshe, and A. Weissman, "A robust gnss los/nlos signal classifier," *NAVIGATION, Journal of the Institute of Navigation*, vol. 63, no. 4, pp. 427–440, 2016.
- [30] P. R. Strode and P. D. Groves, "Gnss multipath detection using three-frequency signal-to-noise measurements," *GPS solutions*, vol. 20, no. 3, pp. 399–412, 2016.
- [31] B. GUERMAH, H. EL GHAZI, T. SADIKI, and H. GUERMAH, "A robust gnss los/multipath signal classifier based on the fusion of information and machine learning for intelligent transportation systems," in *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD)*. IEEE, 2018, pp. 94–100.
- [32] R. Sun, L.-T. Hsu, D. Xue, G. Zhang, and W. Y. Ochieng, "Gps signal reception classification using adaptive neuro-fuzzy inference system," *The Journal of Navigation*, vol. 72, no. 3, pp. 685–701, 2019.
- [33] N. M. Drawil, H. M. Amar, and O. A. Basir, "Gps localization accuracy classification: A context-based approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 262–273, 2012.
- [34] L.-T. Hsu, "Gnss multipath detection using a machine learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

- [35] L. Sun, J. Chen, K. Xie, and T. Gu, "Deep and shallow features fusion based on deep convolutional neural network for speech emotion recognition," *International Journal of Speech Technology*, vol. 21, no. 4, pp. 931–940, 2018.
- [36] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.
- [37] K. Hara, D. Saito, and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–8.