2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), 1-5 October 2023, Barcelona, Spain
DOI: 10.1109/DASC58513.2023.10311160

Combination and Selection of Machine Learning Algorithms in GNSS Architecture Design for Concurrent Executions with HIL Testing

1st Zhengjia Xu
Autonomous and Cyber-Physical Sys.
Cranfield University
Cranfield, UK
billy.xu@cranfield.ac.uk

2nd Ivan Petrunin
Autonomous and Cyber-Physical Sys.
Cranfield University
Cranfield, UK
i.petrunin@cranfield.ac.uk

3rd Antonios Tsourdos
Autonomous and Cyber-Physical Sys.
Cranfield University
Cranfield, UK
a.tsourdos@cranfield.ac.uk

4th Raphael Grech
Spirent Communications Plc
Spirent
Devon, UK
raphael.grech@spirent.com

5th Pekka Peltola
Navigation
Telespazio UK
Luton, UK
pekka.peltola@telespazio.com

6th Smita Tiwari
Navigation
Telespazio UK
Luton, UK
smita.tiwari@telespazio.com

***Abstract*—As machine learning (ML) continuing to gain popularity, ML-assisted Global Navigation Satellite System (GNSS) receivers facilitate the performance of Autonomous Systems (AS) navigation solutions. However, selections of ML is often a trade-off in practice where empirical knowledge is taken to alleviate complexities. Therefore, this paper explores decision-making solutions for maximising determined hardware performance under quantitative and qualitative considerations. This work proposes Algorithm Selection and Matching with Fuzzy Analytic Hierarchy Process (ASM-FAHP) that maps multiple trade-off concerns into a Multi-Criteria Decision-Making (MCDM) problem. The ASM-FAHP firstly searches all the possible alternatives to find possible combinations with hardware resource limitations taken into account. Afterwards, ASM-FAHP prioritizes the most significant candidate by constructing a hierarchical structure with several attributes and scoring with fuzzy numbers. Hereby, the most suitable ML combinations are determined by calculating synthesised fuzzy weights per each alternative. The performance of the ML combination is evaluated by concurrently executing it on resource-constrained hardware, specifically the Jetson Nano board. The ML models are trained and tested using high-fidelity synthetic datasets produced from Spirent GSS7000 simulator and SimGen while connected to hardware-in-the-loop (HIL). It has been discovered that when approaching hardware limits, the selected combination of machine learning algorithms makes full use of memory resources but sacrifices processing speed.**

***Index Terms*—GNSS receiver design, machine learning, deep learning, algorithm selection, FAHP**

I. Introduction

Nowadays, the performance of Global Navigation Satellite System (GNSS) receivers has more requirements in terms of accuracy, availability, continuity, integrity, as well as Size, Weight, Power, and Cost (SwaP-C). Moreover, the evolution of processing hardware platforms brings more opportunities for implementing complex algorithms. Therefore, Machine Learning (ML) has attracted high potential interest due to relaxing theoretical assumptions and the possibility of improving GNSS performance.

Facilitated by self-regression implementations in ML approaches, ML-associated designs enable efficient identification of tightly-coupled dependencies by learning action variables from datasets. By integrating ML method with processing GNSS data, a few applications are anticipated in different ways. For instance, the regressor is commonly used to model error sources induced by ionospheric and troposphere effects, multipath, clock drift, receiver noise, interference, and hardware biases [1] along with error compensations through filtering to reduce degradation effects on accuracy and availability. The self-learning feature in the spatial domain enables dynamic parameter estimations and optimisation applicable in components of GNSS receivers such as acquisition component to aware RF environment [2], determination of discriminator thresholds [3], Receiver Autonomous Integrity Monitoring (RAIM) for intelligent decision-making and scheduling [4], etc.

Some ML assisted GNSS receivers utilise ML's forward propagation features for achieving efficient classification, clustering, forecasting, and anomaly detection applications. After acquiring prior knowledge from a trained neural network where hidden features are learned, layers like Softmax will assign probabilities to each category to decide the existence of features, determine the feature types, and predict feature existence. For instance, [5] applied Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) for detecting jamming types of Continuous Wave (CW) jammers, Chirp jammers, Frequency Modulated (FM) jammers, pulse jammers, and Narrow Band (NB) jamming signals.

This work is performed under the ESA-funded project VTL4AV (NAVISP-EL1-066 bis).

The ML approaches also reveal strong non-linear regression capabilities that can automatically extract inter-relationships between features from datasets that alleviate modelling parameters through theoretical equations. The regression feature also facilitates the determination of adaptive weights when fusing multi-sensor outputs. By feeding various data sources including synthetic datasets, the geographic impact on GNSS performance could be learned and used for compensating effects like multipath [6]. The utilisation of multi-data sources shall maximise the efficiency of data usage regardless of establishing and analysing relations in between.

Nevertheless, the data-driven approaches have also revealed intrinsic shortcomings in generating trustworthy outcomes due to the inability to obtain performance boundaries. Therefore, usages of ML are usually questionable and the decision of ML utilisations in GNSS receivers are subjective relying on judgments from expert knowledge. For some ML application scenarios, the nonlinearity inspired by uncertain links among neurons brings higher demands in hardware performance. As a result, the selection of ML and making decisions on how different ML algorithms will collaboratively function together to improve GNSS performance with practical considerations becomes rather important especially when tradeoffs are taken into account. Accordingly, the ML pairwise and selection problem with factors considered, such as uncertainty and doubts in assessments is demanded in practice which drives the main motivation for proposing a solution to resolve this challenge.

In order to assess a finite number of ML algorithm alternatives with a group of ambiguous criteria, Analytic Hierarchy Process (AHP) was proposed to guide the decision-making by formulating this selection problem as Multi-Criteria Decision-Making (MCDM) where characteristics are structured in a hybridized manner. The principle of AHP is offloading the primary criterion target to subcriterion objectives and its primary steps are: identifying organisation decision goals, identifying attributes in the subcriterion layers, proposition of constraints, and selecting structure alternatives [7].

Furthermore, to reduce human subjectivity including reflexes, preferences and judgments, scoring with fuzzy membership benefits the decision-making robustness by adding degrees of uncertainty rather than using crisp numbers. The exploitation of using fuzzy membership in the AHP is Fuzzy AHP (FAHP) that to be studied in this paper. [9] investigated AHP applications by reviewing publications between 2005 and 2009, it is found that the dominant research fields of FAHP applications are manufacturing, environmental management and agriculture, energy, transportation, construction and health care, whilst using FAHP for algorithm selections or for software implementations has not been sufficiently investigated. [11] attempted to apply FAHP to select the best software architecture designs. [8] studied using FAHP associated with resource allocation methods to choose optimal software system configurations.

ML-aided GNSS receivers can process multiple ML approaches concurrently using efficient memory manipulation circuit designs like GPUs. However, the SWaP-C constraints of processors limit the capability to host an unlimited number of ML algorithms, making it difficult to select appropriate ML combinations. As a result, the following gaps have been identified: (1) scarcity of approaches for selecting ML methods and combinations in the design of GNSS receiver architectures, (2) utilisation of FAHP brings challenges considering hardware resource constraints, (3) characterisation of the selection criterion has not been clear.

In this paper, a two-phase scoring and algorithm selection approach called Algorithm Selection and Matching with Fuzzy Analytic Hierarchy Process (ASM-FAHP) is proposed to find the most suitable ML algorithm combinations based on quality properties with fuzzy integer numbers. The approach addresses hardware resource constraints in the first phase by formulating a combinatorial problem through heuristic traversing all the combinations to rule out those surpassing the necessary hardware specifications. Afterwards, the approach uses AHP models that incorporate expert ratings with fuzzy memberships to score the pre-selected combinations or candidates. The main contribution of this paper is the first attempt of using AHP to ML selections for designing GNSS receivers, and the proposition of a two-phase ASM-FAHP approach where hardware constraints are considered. Attributes of hierarchical architecture including the subcritera are demonstrated in this study with broad considerations. The selected ML algorithms are evaluated in terms of processing performance.

II. ASM-FAHP Method

When dealing with a large number of ML techniques with unknown quantities, the preliminary stage of selection is important to narrow down the number of alternatives with constraints taken into account. Based on the reduced size of alternatives, rating alternatives assisted with human experts' knowledge is hence achievable and manageable.

A. Algorithm Selection and Matching with Constraints

This step aims to identify all possible algorithm combinations $\mathcal{A}$ with available individual algorithm $m \in \mathcal{M}$ where the algorithm m belongs to the space $\mathcal{M}$. Let us consider a function space $\mathcal{F}$ standing for the GNSS functions where ML can deploy. The ML algorithm for the function $f \in \mathcal{F}$ consists of an ML subset $\{f(i)\} \in \mathcal{M}$, where i is the function index.

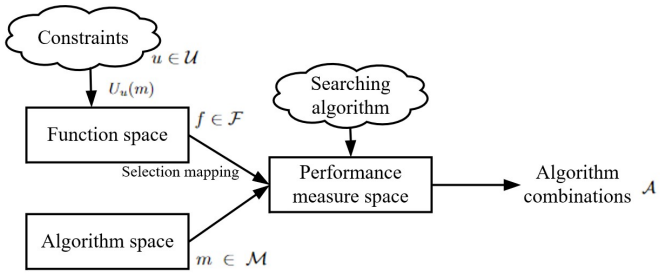

Fig. 1. Illustrative diagram for the algorithm selection and combination problem with feature spaces.

Two types of constraints are derived from assumptions during the selection:

1. The resource utilisation functions $U_u(m)$ assume can be derived from statistic estimations for the m algorithm, and the overall computational cost of the selected combination $\mathcal{A}$ shall limit to a certain value $\mathcal{C}_u$ where $u \in \mathcal{U}$ is the index of the U number of resource utilisation functions.

2. Only one ML algorithm designed for the same use case will be selected to avoid duplicate features. This assumption is converted to the constraint which is setting the maximal number of the selected algorithm among the application group $\{f(i)\}$ to be one.

Consequently, the identification of all the possible algorithm combinations is described:

$$\begin{aligned} \mathcal{A} &= \left\{ \bigcup_{i}^{f \in \mathcal{F}} f(i) \in \mathcal{M} | \emptyset \right\} \\ \text{subject to: } & \sum_{m}^{m \in \mathcal{A}} U_u(m) \leqslant \mathcal{C}_u, \forall u \in \mathcal{U} \\ & \mathbb{N}(\{f\}) \leqslant 1, \forall f \in \mathcal{F} \end{aligned} \tag{1}$$

where the function $\mathbb{N}$ is to calculate the number of a set.

B. Fuzzy Analytic Hierarchy Process Background

The typical procedure of applying FAHP to sort candidates orders follows the following steps [10] in Fig. 2:

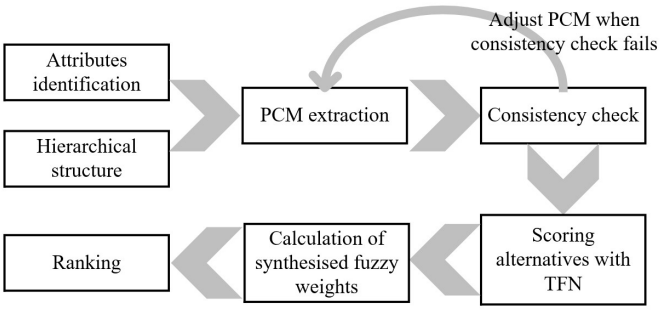

Fig. 2. Block diagram of FAHP phase process.

step (1): identification of quality attributes in the form of hierarchical structures - aims to identify subcriteria to represent key indicators during selections [11].

The hierarchical structure comprises four levels:

- Goal: ultimate purpose of establishing this hierarchical structure.
- Criterion: root attributes in fulfilling the goal.
- Subcriterion: branch attributes placed as children of their root attribute.
- Options: alternatives taken into the selection.

step (2): extraction of comparison matrix - is for specifying importance weights values among subcriteria to reflect relative importance in between.

In hierarchical models, the comparison matrix is set up by offloading importance weight through performing evaluation comparisons between two alternatives and executing the pairwise comparison to create a pairwise comparison matrix (PCM) among all the alternatives.

The equation of calculating comparison matrix value a that describes the pairwise comparison from weights W can be derived:

$$a_{ij} = \frac{W_i}{W_j} \tag{2}$$

where i and j denote index of criterion.

The comparison matrix then requires a normalisation process towards each column described as:

$$\hat{a_{ij}} = \frac{a_{ij}}{\max(a_{ik}, k \in j)}, \forall i, j \tag{3}$$

step (3): consistency check - aims to ensure consistency between PCMs by calculating its largest eigenvalue λ and its accordingly eigenvector.

The consistency measurement uses Consistency Index (CI) definition:

$$CI = \frac{\max(\lambda_i, i \in n) - n}{n - 1} \tag{4}$$

where n denotes the number of subcriteria.

A good consistency check expects CI value close to 0. The PCM is unlikely to be completely consistent over parameters, whilst a certain degree of inconsistency is tolerable. The Saaty testing assisted with a Random Consistency Index (RI) table [7] is usually employed to calculate CR for adding the acceptable errors:

$$CR = \frac{CI}{RI} \tag{5}$$

CR is compared with an empirical number of 0.1. The readjustment of PCM values is needed when $CR > 0.1$ until PCMs meet the satisfactory number of $CR <= 0.1$.

After obtaining satisfying consistent PCMs, the synthesised weights W_i for the i_{th} subcriterion weights w_i after fusion with its parent weights of criterion w_{i-1} can be derived by:

$$W_i(j) = w_i(j) \times w_{i-1}(i) \tag{6}$$

where the weights are the normalised eigenvector values of the PCM for the i_{th} subcriterion.

step (4): marking triangular fuzzy number (TFN) with fuzzy degree of membership - FAHP applies fuzzy number to measure vagueness in the degree of membership, where the fuzzy members are the standard fuzzy sets defined on the set of real number R. The TFN scale with the $M = (l, m, n)$ triple is defined by the membership function $\mu_A(x)$ for a fuzzy number A:

$$\mu_A(x) = \begin{cases} \frac{x-l}{m-l} & x \in [l, m] \\ \frac{u-x}{u-m} & x \in [m, y] \\ 0 & otherwise \end{cases} \tag{7}$$

where l is the lower bound of the field of the possible evaluation, u is the upper bound of the field, and m is the maximal

grade of $\mu_A(x)$. Satty fuzzy triangle set [7] comprising 9 TFN scales is applied for measuring member degrees, and the Satty fuzzy triangle set is illustrated in Fig. 3. It is observed that two neighbour fuzzy numbers show a common overlapping area in the fuzzy number set that aims to represent vague extents during scoring.

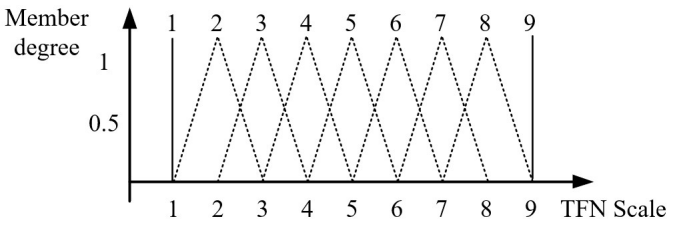

Fig. 3. Graph of Satty fuzzy triangle set.

step (5): Calculation of synthesised fuzzy weights - aims to fuse hybrid TFN to obtain synthesised values per alternatives through multiplying and summarising the fuzzy scale number $s_i(j)$ for the i_{th} candidate over the j_{th} criterion with its synthesised weights $w_i(j)$:

$$S(i) = \sum_j s_i(j) \times W_i(j) \tag{8}$$

Consequently, the ML alternatives can be ranked by ordering S values, and the highest rating number stands for the best ML candidate shall be selected.

III. Case Study for GNSS Receiver Design

After reviewing publications on ML-facilitated GNSS receivers, the following ML methods and applications are highlighted. ML is used as individual components without the need for significant architectural modifications:

• Faulty satellite selection and isolation in RAIM

The increasing number of satellites from multi-constellation contributes to the GNSS performance improvement in terms of accuracy and availability at a cost of extra computational load, processing time and hardware requirements. The selection of high-quality satellites is crucial for reducing computation complexity and narrowing down the receiving frequency of interest, especially with tightly-coupled architecture designs. The proper isolation of satellites facilitates mitigating spoofing attacks by monitoring the geometric distribution of satellites. One common implementation of RAIM is calculating Geometric Dilution of Precision (GDOP) approximation where Support-Vector Regression (SVR) is a common method applied [12].

• Detection of jamming attacks

Jamming detection can be practically built on the principle of signal-level analysis for enhancing robustness in receiving GNSS signals against anti-GNSS environments. The detection of common signal levels like Continuous Wave (CW) and chirp signals appears to be possible using absolute signal level measurements referenced to a noise level (typically based on analysis of automatic gain control voltages). Measuring interference signals at lower levels may require additional spectral analysis to detect jamming or interference, which can be noticed as an opportunity to integrate ML features in the signal spectrum. In [5], typical ways of using CNN and Support-Vector Machine (SVM) methods are investigated to classify jamming types.

• Detection of spoofing attacks

A spoofer shall generate, manipulate, and transmit false signal's waveform, power level, ranging code, and modulated data contents to match GNSS signals. An ideal GNSS spoofing signal shall reveal a corrected transmitter power at a relative distance between the spoofer and receiver to be attacked, as well as containing the synchronisation messages with phase offsets from authentic GNSS signals, and its Doppler shift effect. Regarding ML applications for spoofing detections, [13] and [14] apply unique SVM and K-Nearest Neighbors (KNN) neural networks although the network complexity is difficult to measure.

• NLOS/Multipath classification

Multipath is a major cause of measurement errors in intelligent transport applications according to the study [15]. The detection of multipath is the first stage for RAIM to isolate high multipath affected channels as well as perform multipath mitigations. [2] regards multipath detection as classifying features generated by Cross-Ambiguity Function (CAF). [16] explores SVM in separating the type of GNSS pseudorange measurement into three categories, clean, multipath and NLOS, where Received Signal Strength (RSS), change rate of RSS, and pseudorange residual are the key indicators for the multipath/NLOS classifications.

• INS error prediction

The regressor based ML approaches facilitate reducing stochastic errors in INS models and enable fusion with multiple sensors for predicting accumulative errors during the GNSS outrages. SVR [17] and GRU [6] are the popular and efficient methods in terms of high accuracy performance for the INS error prediction and fusion, thus are taken into account in this study.

A possible GNSS receiver architecture design that integrates the above ML use cases and loosely coupled Inertial Navigation System (INS) architecture is illustrated in Fig. 4.

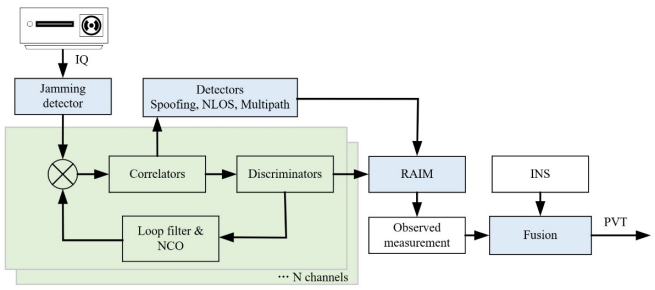

Fig. 4. ML-aided GNSS receiver architecture with integration of loosely coupled architecture.

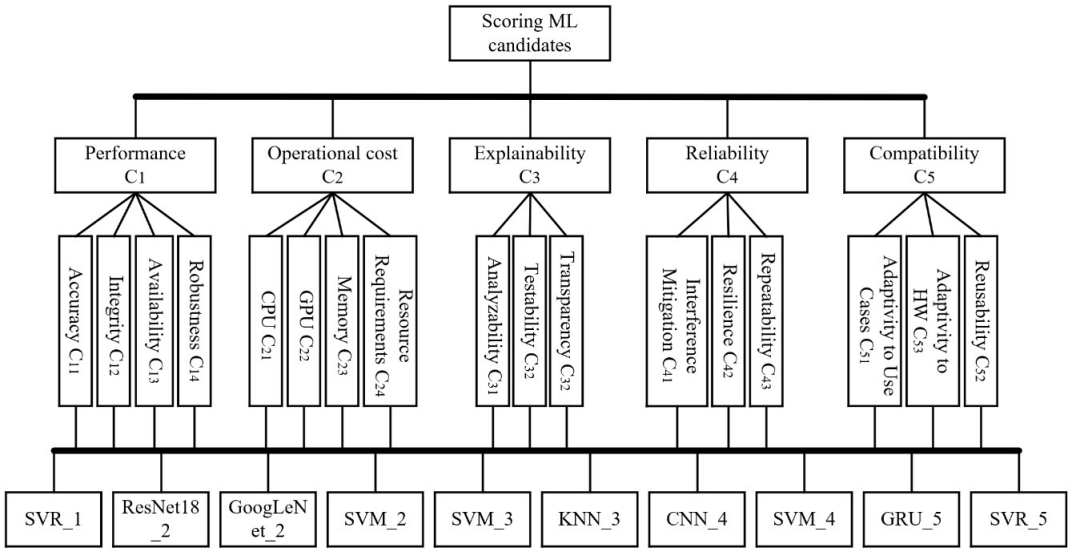

Fig. 5. Hierarchical structure for ML algorithm combination selection.

A. ASM for ML Selection and Combination

Table I summarises prevalent ML use cases where memory utilisations are presumed or estimated from literature. During the simultaneous parallel computations of multiple MLs, memory utilisation is the dominant computational resource representative accounting for out-of-memory (OOM) occurrence.

TABLE I
MEMORY UTILISATION PRESUMPTION FOR ML ALGORITHMS.

Use case	ML	Memory usage	Ref
1 - RAIM	SVR - 4	appr. 1 GB	[12]
2 - Jamming detection	ResNet18	0.69 GB [18]	[5]
	GoogLeNet	0.87 GB [18]	[5]
	SVM	appr. 1 GB	[5]
3 - Spoofing detection	SVM	appr. 1 GB	[13]
	KNN	appr. 1 GB	[14]
4 - Multipath classification	DNN- CNN	est. 0.69 GB	[2]
	SVM	appr. 1 GB	[16]
5 - INS error prediction	GRU	appr. 1.5 GB	[6]
	SVR	appr. 1 GB	[17]

Because of difficulties in sourcing details from references and uncertainties in obtaining computational requirements on a case-by-case basis, memory utilization for uncertain ML is assumed to be around 1 GB. In order to obtain a more accurate estimation of memory consumption, it is important to replicate experiments prior to selection. However, rough calculations can be made.

This study chooses a popular CPU-GPU hybrid platform NVIDIA Jetson Nano to evaluate the selected ML combinations because of its high-compact portable design and flexibility to enable a wide range of AI applications. The hardware specifications for Jetson Nano are Quad-core ARM conrtex-A57 CPU, NVIDIA Maxwell 128 CUDA cores, Tegra210 system on a chip (SoC), and 4 GB memory. The operating system is Ubuntu 18.04 with software dependencies of OpenCV 4.5.3, CUDA toolkit 10.2.300, and TensorRT 8.0.1.6.

The idle state of Jetson Nano board usually consumes around 1.5 GB of memory which leaves 2.5 GB remaining for processing ML applications. In order to bring up the parallel computation feature in the Jetson Nano, at least 2 algorithms shall be selected in $\mathcal{A}$. Hereby, the ASM problem is further simplified as:

$$\begin{aligned} \mathcal{A} &= \left\{ \bigcup_{i}^{f\in\mathcal{F}} f(i) \in \mathcal{M} | \emptyset \right\} \\ \text{subject to: } & \sum_{m}^{m\in\mathcal{A}} O_f(m) \leqslant 2.5 \\ & \mathbb{N}(\{f\}) \leqslant 1, \forall f \in \mathcal{F} \\ & \mathbb{N}(\{\mathcal{A}\}) \geqslant 2 \end{aligned} \tag{9}$$

where $O_f(m)$ denotes the memory occupation in GB unit of m algorithm for f use case.

The solver of (9) is developed based on an exhaustive searching algorithm with the principle of scanning alternatives m per each function set f sequentially, and filtering out those that do not satisfy constraints. It is easy to conclude that only 2 algorithms can run simultaneously in Jestson Nano regarding the constraint RAM size and preassumption of the memory utilisations. Therefore, the set $\mathcal{A}$ comprises of 39 pairs and rating each would take up considerable time.

B. Attributes Determination and Hierarchical Structure Construction

Given the obtained ML combinations $\mathcal{A}$, the next objective is sorting the preference of ML combinations following FAHP. The first action of FAHP is to choose attributes for determining the combination scores and create hierarchical structures based on the subcriteria.

The selection of the quality attributes applied for measuring ML algorithm performance refers to ISO/IEC model

C	C1	C2	C3	C4	C5
C1	1	2	4	2	2
C2	1/2	1	2	2	2
C3	1/4	1/2	1	1/2	1/2
C4	1/2	1/2	2	1	2
C5	1/2	1/2	2	1/2	1

C1	C11	C12	C13	C14
C11	1	1/3	2	7
C12	3	1	5	9
C13	1/2	1/5	1	3
C14	1/7	1/9	1/3	1

C2	C21	C22	C23	C24
C21	1	1/2	1/3	3
C22	2	1	1/2	4
C23	3	2	1	7
C24	1/3	1/4	1/7	1

C3	C31	C32	C33
C31	1	2	3
C32	1/2	1	2
C33	1/3	1/2	1

C4	C41	C42	C43
C41	1	1/3	1/6
C42	3	1	1/2
C43	6	2	1

C5	C51	C52	C53
C51	1	5	2
C52	1/5	1	1/3
C53	1/2	3	1

Fig. 6. PCMs of subcritera with weighted values.

[11] proposed for describing software quality evaluations. The ISO/IEC model attributes are identified as functionality, reliability, usability, efficiency, maintainability, and portability. Distinguished from software quality attributes, attributes for this ML selection scenario in designing GNSS receivers focus on balancing tradeoffs between the GNSS performance enhancement and the execution burden.

The following attributes are identified as the dominant factors during the selection phase of ML algorithms:

- The key performance indicators of measuring performance enhancement by incorporating ML into GNSS modules select accuracy, integrity, availability, and robustness. Accuracy, integrity and availability factors improve the GNSS positioning accuracy directly, whilst robustness stands for ability against interference like jamming, and spoofing.
- The operational cost attribute is extracted to evaluate the running cost in terms of CPU, GPU, memory, and other resource requirements such as data parser implementations to support data exchange between multiple ML algorithms.
- Explainability measures the degree that humans comprehend the underlying mechanisms of ML, where analyzability, testability, and transparency are identified as subcriterion attributes. Analyzability measures the difficulty of analysing the ML output. Testability evaluates the difficulty of obtaining ML outcomes. Transparency stands for difficulty in tracking ML behaviours when generating the outcomes.
- Reliability of ML indicates the capacity to produce consistent and repeatable results. The reliability is further characterised by repeatability, resilience to recover faulty states, and the ability to withstand incidents such as interference.
- Compatibility represents the adaptability of ML for additional use cases and hardware platforms. The compatibility consists of subcriteria of reusability, adaptivity to hardware, and adaptivity to use scenarios.

The $\mathcal{A}$ set contains a large number set of 39 leading to difficulties in fulfilling a PCM with 39×39 dimensions. The pairwise among those alternatives should be performed individually for 39 times and also the consistency in the matrix is difficult to maintain. Therefore, this paper focuses on scoring towards individual ML rather than ML combinations because of superposition property of FAHP.

The hierarchical structure is displayed in Fig. 5 where the ML alternatives are obtained from Table I. The hierarchical architecture for this demonstration consists of a two-layer structure defined as criterion C_i and subcriterion C_{ij} respectively.

C. Weights Distribution

The weighting values in PCMs for subcriteria are generated based on expert scores and (2), and the PCMS for this study are demonstrated in Fig. 6.

The consistency check following (4) and (5) over the PCMs proves that the rated weights meet the consistency requirements by having all the CR values less than 0.1. Table II records the meticulously calculated values of CI, CR, and maximum eigenvectors representing the assigned weights.

TABLE II
CI AND CR CALCULATION FOR EACH CRITERION OF PCM.

	CI	CR	Eigenvector
C	0.03	0.03	$[0.35, 0.24, 0.09, 0.18, 0.14]$
C1	0.02	0.02	$[0.25, 0.58, 0.12, 0.05]$
C2	0.007	0.008	$[0.16, 0.28, 0.50, 0.06]$
C3	0.005	0.008	$[0.54, 0.30, 0.16]$
C4	0	0	$[0.1, 0.3, 0.6]$
C5	0.002	0.003	$[0.58, 0.11, 0.31]$

Consequently, the synthesised weights distributed over the subcriterion W_i can be calculated from (6) by multiplying with their parent criterion W_{i-1}. The calculated distributed weights $w_i(j)$ for the subcriterion C_{ij} are summarised in Table III.

TABLE III
DISTRIBUTED WEIGHTS TABLE OVER SUBCRITERA.

	C11	C12	C13	C14
w1j	0.0875	0.203	0.042	0.0175
	C21	**C22**	**C23**	**C24**
w2j	0.0384	0.0672	0.12	0.0144
	C31	**C32**	**C33**	
w3j	0.0486	0.027	0.0144	
	C41	**C42**	**C43**	
w4j	0.018	0.054	0.108	
	C51	**C52**	**C53**	
w5j	0.0812	0.0154	0.0434	

After analysing the synthesised weighting values of the subcriterion C_{ij}, it has been determined that the most important 3 subcritera are integrity, memory utilisation and accuracy. This indicates that the primary goal of this selection is to find an ML combination that uses minimal memory while achieving maximum improvements in terms of integrity and accuracy.

D. Evaluation with Fuzzy Membership

The fuzzy membership numbers are applied as indicators to measure ML performance. The Satty fuzzy number set is applied here for demonstration purposes. Table IV has listed fuzzy numbers for ML alternatives, and the table also includes calculation results of synthesized fuzzy weights per each ML alternative following (8). The graphic diagram Fig. 7 illustrates final ranking results among ML alternatives in terms of member degree.

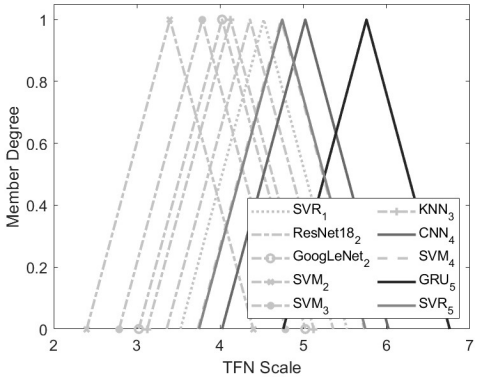

Fig. 7. Graph of FTP for ML alternatives.

From Table IV, the result of ASM-FAHP indicates the highest score of 5.7, i.e. applying GRU to predict INS error. The second highest score of 5.0 is for using CNN to classify multipath effects. The third best option for ML applications applies SVM / SVR approaches to perform the multipath detections again. Regarding the ASM conclusions derived by resolving (9), the most appropriate ML combination is hereby using the GRU and CNN combination for INS error predictions and multipath classifications with considerable factors balanced during the decision-making stage.

It is worth noting that the selected ML combination is only applicable to the demonstrated expert scores and the pairwise weights per subcriterion. During scoring, we deliberately focus on GNSS performance improvement with fewer concentrations on reliability and other factors. Obtaining knowledge from multiple experts shall benefit in reducing bias in judgment to provide decent outcomes.

IV. VERIFICATION WITH HIL DATASET

For evaluating the operation performance of the selected ML combination, this paper adopts hardware-in-the-loop (HIL) testing datasets produced from Spirent GSS7000 simulator, and SimGen for training and testing neural networks (see Fig. 8 for HIL setup details). Specifically, the GRU testing applies the collected synthetic dataset for analysing the ML operating performance. The ResNet-18 uses image datasets with input size of $700 \times 560 \times 3$ pixel dimensions.

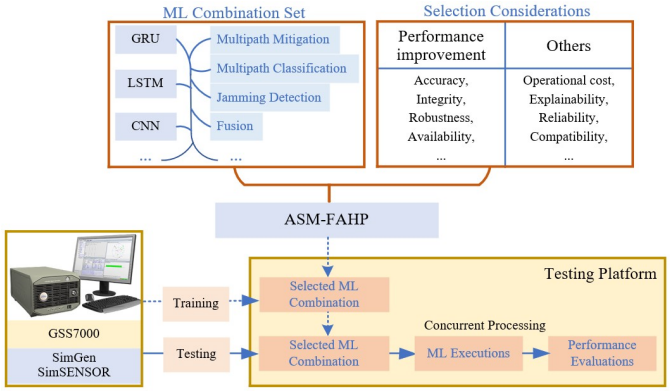

Fig. 8. HIL verification architecture.

In this work, the ML operation performance will be measured using performance indicators such as memory consumption, GPU utilization ratio, CPU temperature, and GPU temperature. The testing scenarios that need to be performed have been outlined:

• Multipath classification simulation with CNN for multiple time periods - to evaluate the CNN execution performance on the Jestson Nano hardware (see Fig. 9). The testing scenario will run multipath detection 5 times.

From Fig. 9, it is found that the initial RAM usage of Jetson Nano is close to the expected amount, consuming approximately 1500 MB. The RAM consumption of ResNet18 during classifications peaks by around 700 MB when the GPU engine operating at full capacity. The average time interval between two classifications is around 4.5 seconds. The CPU and GPU temperatures show slight changes with absolute deviations of $3.5°C$ and $1.5°C$, respectively.

• Simulation of INS error prediction with GRU implementation by feeding GNSS performance parameters as inputs and loading with pre-trained GRU networks - to evaluate GRU performance on the Jetson Nano. The multipath observables and parameters include pseudorange, ephemerides, Doppler shift, $C/N0$, and elevation from each satellite [6]. The GRU architecture comprises of 250 hidden units, ReLU, Softmax,

TABLE IV
FUZZY NUMBER VALUE OF ML ALTERNATIVES ON SUBCRITERIA.

Use case	ML	C11	C12	C13	C14	C21	C22	C23	C24	C31
RAIM	**SVR_1**	7	9	2	5	2	2	2	7	4
Jamming detection	**ResNet18_2**	3	2	7	8	4	7	2	3	7
	GoogLeNet_2	3	2	7	8	4	2	2	3	7
	SVM_2	3	2	7	8	2	2	2	3	5
Spoofing detection	**SVM_3**	3	2	7	8	2	7	2	3	5
	KNN_3	3	2	7	8	4	3	4	3	6
NLOS/multipath detection	**CNN_4**	5	6	6	6	4	2	2	3	7
	SVM_4	5	6	6	6	2	7	2	3	5
INS error prediction	**GRU_5**	6	7	6	6	5	2	7	8	3
	SVR_5	6	7	6	6	2	7	2	8	4

Use case	ML	C32	C33	C41	C42	C43	C51	C52	C53	*Val* S
RAIM	**SVR_1**	2	5	2	4	4	2	7	2	**4.524**
Jamming detection	**ResNet18_2**	6	4	7	3	5	7	3	8	**4.3579**
	GoogLeNet_2	6	4	7	3	5	7	3	8	**4.0229**
	SVM_2	5	6	7	3	6	2	7	3	**3.3973**
Spoofing detection	**SVM_3**	5	6	7	4	6	2	7	3	**3.7863**
	KNN_3	6	4	7	4	5	6	3	5	**4.1239**
NLOS/multipath detection	**CNN_4**	6	4	6	5	5	7	3	8	**5.0229**
	SVM_4	5	6	6	5	6	2	7	3	**4.7323**
INS error prediction	**GRU_5**	7	5	6	6	4	7	2	7	**5.758**
	SVR_5	2	5	6	6	4	2	7	2	**4.7454**

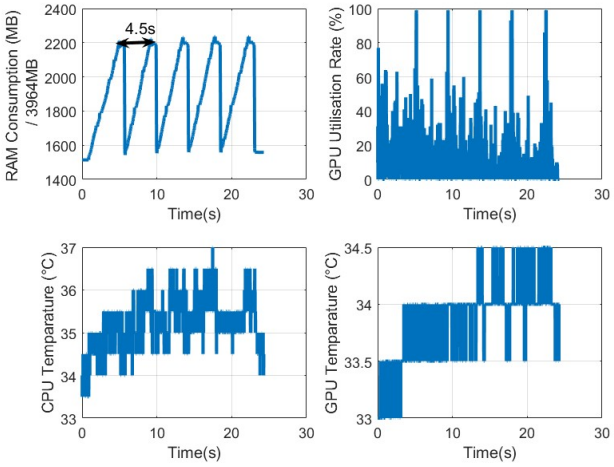

Fig. 9. CNN operation performance on Jestson Nano.

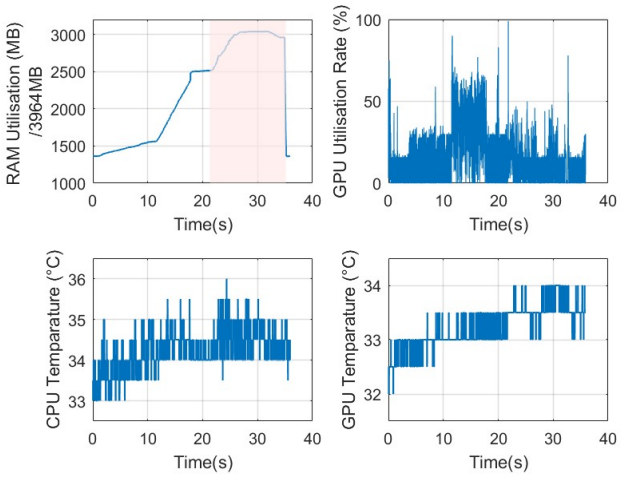

Fig. 10. GRU operation performance on Jestson Nano.

and dropout layers. The operation performance of deploying GRU on Jetson Nano is shown in Fig. 10.

From Fig. 10, it is observed that initialising the GRU and TensorFlow packages consumes the majority of memory, approximately 1100 MB. However, the prediction phase only consumes around 500 MB of memory. The peak of RAM consumption when executing the GRU-based multipath prediction is 36% higher than multipath classifications because of extra memory during initialisation. From this demonstration, it can be observed that GRU processing utilizes lighter computational resources compared to CNN executions. This is evident from the GPU utilization ratio figure, where GRU processing utilizes almost half of what CNN executions use.

• Simulation of concurrent processing for multipath classification and INS error predictions - to evaluate the simultaneous parallel processing capability of deploying CNN and GRU on the same Jetson Nano. The testing scenario will run multiple multipath classifications and run INS error predictions at the same time. The operation performance is illustrated in Fig. 11

Fig. 11 shows that when the RAM reaches saturation, the

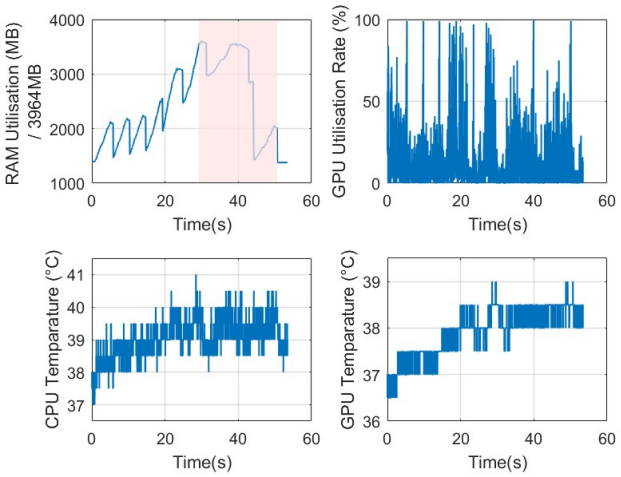

Fig. 11. GRU and CNN concurrent operation performance on Jestson Nano.

speed of processing both CNN and GRU slows down even if the GPU engine is still vacant. The maximum time taken for a single multipath classification is almost 11 seconds, which is 2.4 times longer than the time taken without INS error predictions. The INS error predictions take a total of 24 seconds, which have additional 11 seconds longer than processing without multipath detections. During the period of $29-53$ seconds, the operation performance degrades when the RAM uses exceeds 3000 MB. The highest RAM consumption observed is around 3600 MB, which is close to the maximum hardware RAM capacity of 3964 MB. The CPU and GPU temperatures display an upward trend, i.e. around $3°C$ and $1.5°C$ temperature increments over 50 seconds period.

V. Conclusion

This paper presents a method, i.e. ASM-FAHP that aims to resolve the algorithm selection and matching challenges and select the most efficient ML combinations to optimise GNSS performance with distinguished considerations. The proposed solution involves two stages. First, a heuristic search is conducted to preliminarily identify the combination set based on MCMD objective functions. A hierarchical structure is constructed to evaluate multiple attributes. Associated with expert knowledge, the weight values of subcritera are scored along with the scores per individual ML alternative with fuzzy logic numbers. The most appropriate ML combinations are determined by calculating synthesised values after ensuring the consistency of PCMs. To confirm the effectiveness of the selected ML combination, HIL datasets are generated and fed to neural networks for training. Jetson Nano is used as the hardware platform to verify ML execution performance. The results demonstrated that the selected ML combination can be processed in parallel within the same processing board despite limited hardware resources. This method can be applied to various algorithm selection applications and has the potential to improve functionality and performance of GNSS receivers. The study will investigate solutions to improve performance, including adding RAM, optimizing network complexity, static unrolling, batch processing, and using GPU-accelerated layers.

References

[1] Siemuri, Akpojoto, Kannan Selvan, Heidi Kuusniemi, Petri Valisuo, and Mohammed E. Elmusrati. "A Systematic Review of Machine Learning Techniques for GNSS Use Cases." IEEE Transactions on Aerospace and Electronic Systems (2022).

[2] Borhani-Darian, Parisa, Haoqing Li, Peng Wu, and Pau Closas. "Deep Learning of GNSS Acquisition." Sensors 23, no. 3 (2023): 1566.

[3] Xia, Yan, Shuguo Pan, Xiaolin Meng, Wang Gao, Fei Ye, Qing Zhao, and Xingwang Zhao. "Anomaly detection for urban vehicle GNSS observation with a hybrid machine learning system." Remote Sensing 12, no. 6 (2020): 971.

[4] Singh, Prateek, Janamejay Joshi, Abhijit Dey, and Nitin Sharma. "GNSS Satellite Selection-based on Per-satellite Parameters Using Deep Learning." IETE Journal of Research (2022): 1-12.

[5] Morales Ferre, Ruben, Alberto de la Fuente, and Elena Simona Lohan. "Jammer classification in GNSS bands via machine learning algorithms." Sensors 19, no. 22 (2019): 4841.

[6] Geragersian, Patrick, et al. "An INS/GNSS fusion architecture in GNSS denied environment using gated recurrent unit." AIAA SCITECH 2022 Forum. 2022.

[7] Saaty, Thomas L. What is the analytic hierarchy process?. Springer Berlin Heidelberg, 1988.

[8] Averchenkov, Vladimir Ivanovich, et al. "Fuzzy and hierarchical models for decision support in software systems implementations." Knowledge-Based Software Engineering: 11th Joint Conference, JCKBSE 2014, Volgograd, Russia, September 17-20, 2014. Proceedings 11. Springer International Publishing, 2014.

[9] Sipahi, Seyhan, and Mehpare Timor. "The analytic hierarchy process and analytic network process: an overview of applications." Management decision 48, no. 5 (2010): 775-808.

[10] Enea, Mario, and Tommaso Piazza. "Project selection by constrained fuzzy AHP." Fuzzy optimization and decision making 3 (2004): 39-62.

[11] Moaven, Shahrouz, and Jafar Habibi. "A fuzzy-AHP-based approach to select software architecture based on quality attributes (FASSA)." Knowledge and Information Systems 62 (2020): 4569-4597.

[12] Wu, Chih-Hung, Wei-Han Su, and Ya-Wei Ho. "A study on GPS GDOP approximation using support-vector machines." IEEE Transactions on Instrumentation and Measurement 60.1 (2010): 137-145.

[13] Semanjski, Silvio, Ivana Semanjski, Wim De Wilde, and Alain Muls. "Use of supervised machine learning for gnss signal spoofing detection with validation on real-world meaconing and spoofing data—part i." Sensors 20, no. 4 (2020): 1171.

[14] Qin, Wenjian, and Fabio Dovis. "Situational Awareness of Chirp Jamming Threats to GNSS Based on Supervised Machine Learning." IEEE Transactions on Aerospace and Electronic Systems 58, no. 3 (2021): 1707-1720.

[15] Imparato, Davide, Ahmed El-Mowafy, and Chris Rizos. "Integrity monitoring: From airborne to land applications." Multifunctional operation and application of GPS (2018): 23-43.

[16] Hsu, Li-Ta. "GNSS multipath detection using a machine learning approach." 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017.

[17] Cong, Li, Song Yue, Honglei Qin, Bin Li, and Jintao Yao. "Implementation of a MEMS-based GNSS/INS integrated scheme using supported vector machine for land vehicle navigation." IEEE Sensors Journal 20, no. 23 (2020): 14423-14435.

[18] Bianco, Simone, Remi Cadene, Luigi Celona, and Paolo Napoletano. "Benchmark analysis of representative deep neural network architectures." IEEE access 6 (2018): 64270-64277.

2023-11-10

Combination and selection of machine learning algorithms in GNSS architecture design for concurrent executions with HIL testing

Xu, Zhengjia

IEEE

Zhengjia X, Petrunin I, Tsourdos A, et al., (2023) Combination and selection of machine learning algorithms in GNSS architecture design for concurrent executions with HIL testing. 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), 1-5 October 2023, Barcelona, Spain

https://doi.org/10.1109/DASC58513.2023.10311160