

CURRENT TREND AND STUDIES IN ELECTRICAL, ELECTRONIC AND COMMUNICATION ENGINEERING



All Sciences Academy

***CURRENT TREND AND
STUDIES IN ELECTRICAL,
ELECTRONIC AND
COMMUNICATION
ENGINEERING***

Editor

Prof. Dr. Enes YİĞİT





Current Trends and Studies in Electrical, Electronic and Communication Engineering

Editor: Prof. Dr. Enes YİĞİT

Design: All Sciences Academy Design

Published Date: September 2026

Publisher's Certification Number: 72273

ISBN: 978-625-5472-04-5

© All Sciences Academy

www.allsciencesacademy.com

allsciencesacademy@gmail.com

CONTENT

1. Chapter	5
2-Degree-of-Freedom PID Control Application for a DC Motor Using the Jetson Nano Platform	
<i>Erdem ILTEN</i>	
2. Chapter	16
Encoding Inertial Reference Unit Data Using ARINC 429 and Software-Based Verification	
<i>Murat EKİCİ</i>	
3. Chapter	38
Radar Vital Signs Detection by Using Optimized CNN+ GRU Model	
<i>Umut OZKAYA</i>	

2-Degree-of-Freedom PID Control Application for a DC Motor Using the Jetson Nano Platform

Erdem ILTEN¹

1- Assoc. Prof. Dr.; Balıkesir University Faculty of Engineering Department of Electrical-Electronics Engineering. erdemilten@balikesir.edu.tr ORCID No: 0000-0002-9608-2148

ABSTRACT

This study presents a 2-Degree-of-Freedom (2DOF) PID position control application for a Direct Current (DC) motor using the NVIDIA Jetson Nano platform. To overcome the performance trade-off between setpoint tracking and disturbance rejection commonly encountered in conventional PID controllers, the 2DOF PID control architecture, which mathematically decouples these two objectives, was utilized. The control algorithm was developed in the Simulink environment using a Model-Based Design (MBD) approach and was deployed directly to the Jetson Nano platform through automatic code generation via the Simulink *Embedded Coder* tool. The experimental setup consists of a brushed DC motor, an optical encoder, a motor driver circuit, and a Jetson Nano. Furthermore, a user panel was designed in Simulink for the remote control and real-time data monitoring of the system. In the experimental tests, the system's responses to varying reference position values (0, 25, 60, 85, 50 and 30) were analyzed. It was observed that the controller operated stably even with unoptimized gain parameters ($K_p = 0.08$, $K_i = 0.06$, $K_d = 0.02$, $b = 0.8$ and $c = 0.6$). It is stated that the system would perform much more effectively with optimized gains.

Keywords – DC motor, 2DOF PID, Jetson Nano, Simulink, Embedded Coder.

INTRODUCTION

A Direct Current motor, commonly known as a DC motor, is an electrical machine designed to convert direct current electrical energy into mechanical rotation. It operates on the fundamental principle that a current-carrying conductor placed within a magnetic field experiences a mechanical force. This transformation relies on several core components working together seamlessly to generate continuous motion. The stationary outer housing of the motor, called the stator, contains either permanent magnets or electromagnetic field windings that create a continuous and fixed magnetic field. Inside this magnetic field sits the rotor, or armature, which is the rotating internal assembly made up of coils of wire wound around an iron core. To deliver power from the external direct current source to this moving armature, the motor uses brushes, which are carbon or metal contacts that maintain a sliding electrical connection. These brushes connect to the commutator, a mechanical switch typically designed as a split copper ring attached to the rotor shaft. The commutator periodically reverses the direction of the electrical current in the rotor coils to ensure that the rotational force, or torque, constantly pushes in the same direction. When voltage is applied, current travels through the brushes, across the commutator, and into the armature coil. According to the Lorentz Force Law, this current interacts with the stator's magnetic field, causing one side of the coil to be pushed upward while the opposite side is

pushed downward, which creates the torque that rotates the shaft. As the rotor spins through the magnetic field, it simultaneously acts as a generator, producing a voltage that opposes the supplied voltage. This phenomenon is known as Back Electromotive Force, and because it is directly proportional to the motor's speed, it naturally regulates the current drawn from the supply and stabilizes the motor's speed under varying physical loads (Ilten, 2025c, 2025b; Küçükdermenci, 2024, 2025b, 2025a; Küçükdermenci & Ilten, 2025).

Proportional-Integral-Derivative controllers have long served as the fundamental backbone of industrial control systems due to their structural simplicity, reliability, and robust performance across a wide range of operating conditions. The traditional, or one-degree-of-freedom, PID architecture relies on a single error signal (the straightforward difference between the desired setpoint and the measured process variable) to compute the corrective control action. While highly effective for many standard applications, this unified approach inherently couples the controller's response to command tracking with its response to external load disturbances. Consequently, control engineers are frequently forced into a performance compromise during the tuning process. Optimizing the controller gains strictly for rapid and aggressive disturbance rejection often leads to a hyper-responsive setpoint tracking behavior, characterized by excessive overshoot, transient instability, and the sudden control spikes commonly referred to as derivative kick. To overcome the inherent structural limitations of the traditional one-degree-of-freedom architecture, the two-degree-of-freedom PID controller emerged as a more advanced and flexible control strategy. The primary distinction between the two topologies lies in the mathematical decoupling of the setpoint tracking and disturbance rejection objectives. Instead of restricting all control actions to a single error calculation, the two-degree-of-freedom framework introduces setpoint weighting parameters. These structural weights independently scale the influence of the reference signal on the proportional and derivative control efforts, while ensuring the integral action continues to operate on the true error to eliminate steady-state offsets. By processing the reference command and the system feedback as mathematically distinct entities, the controller effectively behaves as two independent control algorithms working cooperatively. When comparing the operational outcomes of the two architectures, the advantages of the two-degree-of-freedom PID controller become particularly evident in complex dynamic systems where both robust disturbance attenuation and smooth setpoint transitions are critical. In a traditional setup, mitigating unacceptable setpoint overshoot typically necessitates detuning the primary controller gains, which invariably degrades the system's ability to recover from sudden external load variations. Conversely, the decoupled configuration allows for an aggressive baseline tuning targeted entirely at optimal disturbance rejection, followed by a secondary adjustment of the setpoint weights to

systematically shape and soften the command response. This independent tuning capability not only eliminates the adverse effects of abrupt reference changes but also maximizes the overall closed-loop bandwidth, providing a comprehensive solution for advanced processes that cannot afford to compromise between operational stability and dynamic responsiveness (Alfaro & Vilanova, 2013; Bingi, Ibrahim, Karsiti, Hassan, & Harindran, 2018, 2019; Guras, Strambersky, & Mahdal, 2022).

The increasing complexity of modern control and signal processing systems demands highly efficient development workflows. Model-Based Design (MBD) has emerged as a cornerstone paradigm in engineering, allowing researchers to visually simulate, verify, and validate complex algorithms long before physical hardware deployment. Within this ecosystem, Simulink *Embedded Coder* serves as a critical bridge, automating the translation of high-level block diagrams into optimized, production-ready C and C++ code. This automated code generation minimizes human translation errors and significantly accelerates the iterative design cycle, enabling developers to focus entirely on algorithmic refinement rather than the intricacies of low-level software implementation (Bilgen & Ilten, 2026; Ilten, 2025d, 2026c; Ilten & Bilgen, 2026b, 2026a; Ilten, Duzgun, & Gullu, 2026).

Concurrently, the rapid evolution of edge computing has shifted the computational burden away from centralized cloud servers directly to the data source. The NVIDIA Jetson Nano platform stands at the forefront of this shift, offering a highly efficient, compact, and energy-conscious architecture tailored specifically for embedded systems. Equipped with an integrated GPU, the Jetson Nano provides the immense parallel processing capabilities required for executing computationally intensive tasks in real-time. It has become an essential hardware target for deploying advanced computer vision pipelines and intelligent algorithms directly at the edge, where low latency and immediate, localized decision-making are paramount (Valladares, Toscano, Tufiño, Morillo, & Vallejo-Huanga, 2021; Zhang et al., 2022).

The integration of Simulink Embedded Coder with the Jetson Nano platform creates a powerful, unified framework for developing sophisticated edge intelligence systems. By leveraging dedicated hardware support packages, developers can seamlessly deploy complex, computationally heavy models (such as convolutional neural networks like SqueezeNet and ResNet-50) directly onto the Jetson's GPU architecture straight from a Simulink environment. This streamlined workflow is particularly advantageous for real-time applications requiring continuous environmental perception, such as vision-based robotics, automated physical inspection, and smart agricultural monitoring. Ultimately, combining high-level model-based design with robust edge computing hardware empowers researchers to rapidly prototype,

validate, and deploy cutting-edge deep learning and computer vision architectures without the traditional bottlenecks of manual, hardware-level programming (Ilten, 2026b, 2026d, 2026a).

The paper is organized as follows: The Materials and Methods section describe the 2-Degree-of-Freedom PID control structure and Simulink block diagram prepared for Jetson Nano platform. The Results and Discussion section presents the results of the experimental studies.

MATERIALS AND METHOD

A 2-Degree-of-Freedom (2DOF) PID controller is an advanced variation of the standard Proportional-Integral-Derivative (PID) control algorithm. While a standard PID controller is highly effective for many industrial processes, it fundamentally links the controller's response to setpoint changes with its response to system disturbances. A 2DOF PID controller decouples these two objectives, allowing engineers to independently tune the system for optimal setpoint tracking and optimal disturbance rejection.

In a conventional PID controller (1-Degree-of-Freedom), the control action is calculated based on a single error signal, which is the difference between the reference setpoint and the measured process output:

$$e(t) = r(t) - y(t) \quad (1)$$

Because the proportional (P) and derivative (D) terms act directly on this single error signal, tuning the controller for rapid disturbance rejection often results in a highly aggressive response to setpoint changes. This can cause significant overshoot or "derivative kick" when the setpoint is adjusted abruptly. Conversely, tuning the system to prevent overshoot during setpoint changes often makes the system too sluggish to effectively handle sudden external disturbances.

A 2DOF PID controller solves this compromise by introducing setpoint weighting. This allows the proportional and derivative actions to act differently on the reference signal $r(t)$ than they do on the measured output $y(t)$.

In the time domain, the control law for a 2DOF PID controller is typically expressed as (Ilten, 2025a):

$$u(t) = K_p (b \cdot r(t) - y(t)) + K_i \int_0^t (r(\tau) - y(\tau)) d\tau + K_d \left(c \cdot \frac{dr(t)}{dt} - \frac{dy(t)}{dt} \right) \quad (2)$$

Where K_p , K_i , and K_d are the standard proportional, integral, and derivative gains. b is the proportional setpoint weight (typically $0 \leq b \leq 1$). c is the derivative setpoint weight (typically $0 \leq c \leq 1$).

In the Laplace domain, this decoupling becomes even clearer, showing how the controller acts as two separate transfer functions:

$$U(s) = \left(b \cdot K_p + \frac{K_i}{s} + c \cdot K_d s \right) R(s) - \left(K_p + \frac{K_i}{s} + K_d s \right) Y(s) \quad (3)$$

In this study, a 2DOF PID controller is used for the position control of a DC motor using a Jetson Nano. The block diagram of the system is presented in Figure 1. The experimental setup prepared for the system is shown in Figure 2. The experimental setup consists of a brushed DC motor, an optical encoder, a DC motor driver circuit, a DC power supply and a Jetson Nano.

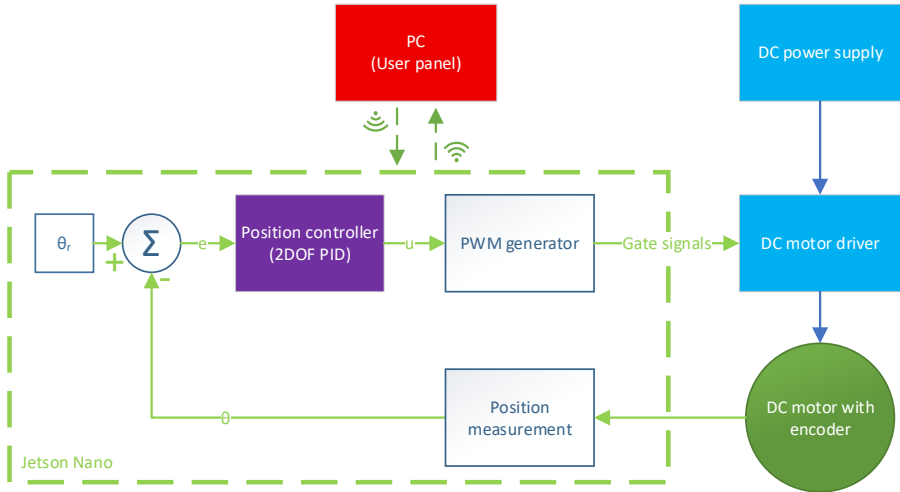


Figure 1: Block diagram of the 2DOF PID position control system.

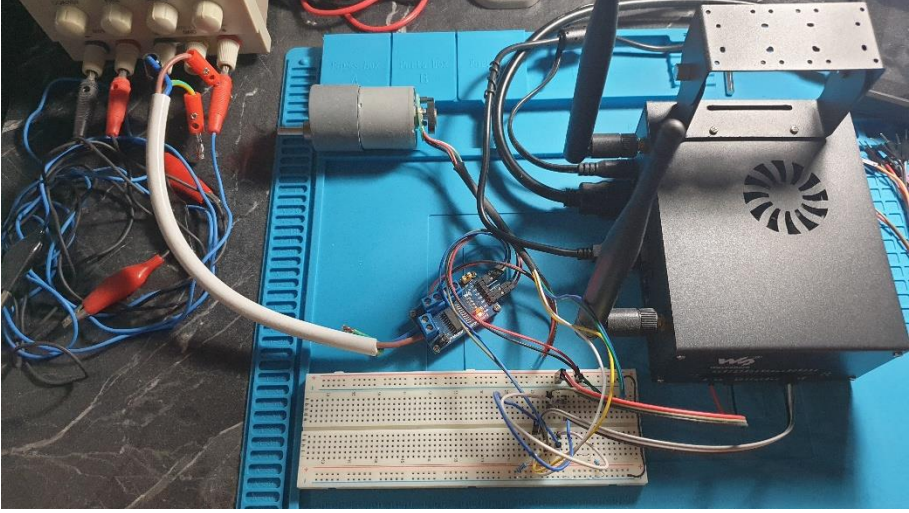


Figure 2: The experimental setup.

A 2-DOF PID position control algorithm for the Jetson Nano was developed in the Simulink environment. The *MATLAB Coder Support Package for NVIDIA Jetson and NVIDIA DRIVE Platforms* was used for this purpose (MathWorks, 2026). The developed block diagram is shown in Figure 3. This diagram was deployed to the Jetson Nano platform using Simulink's *Embedded Coder* tool.

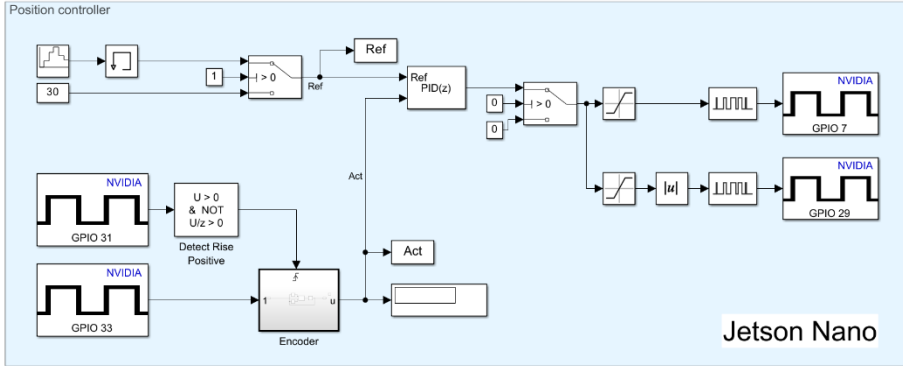


Figure 3: Simulink block diagram for Jetson Nano.

A user panel has been designed in the Simulink environment for the remote monitoring and control of a Jetson Nano-based DC motor control system. The Simulink *Dashboard* library was used for this purpose. The designed panel is presented in Figure 4.

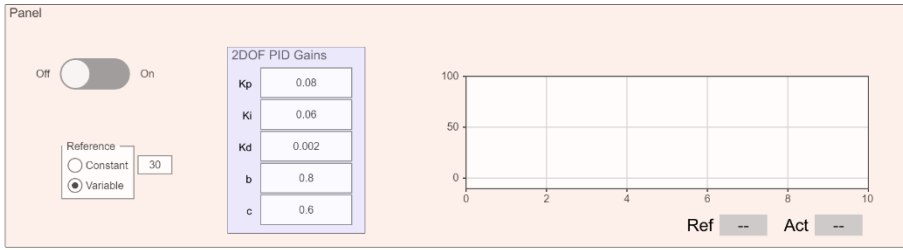


Figure 4: User control panel.

Via the panel shown in Figure 4, the system can be turned on and off, the reference position can be selected as constant or variable, the 2DOF PID gains (K_p , K_i , K_d , b and c) can be adjusted, reference and actual position data can be monitored in real time.

RESULTS AND DISCUSSION

The 2DOF PID position control system was tested on an experimental setup. The reference speed was changed every two seconds to 0, 25, 60, 85, 50 and 30, respectively, and the system's response was analyzed. The obtained results are presented in Figure 5.

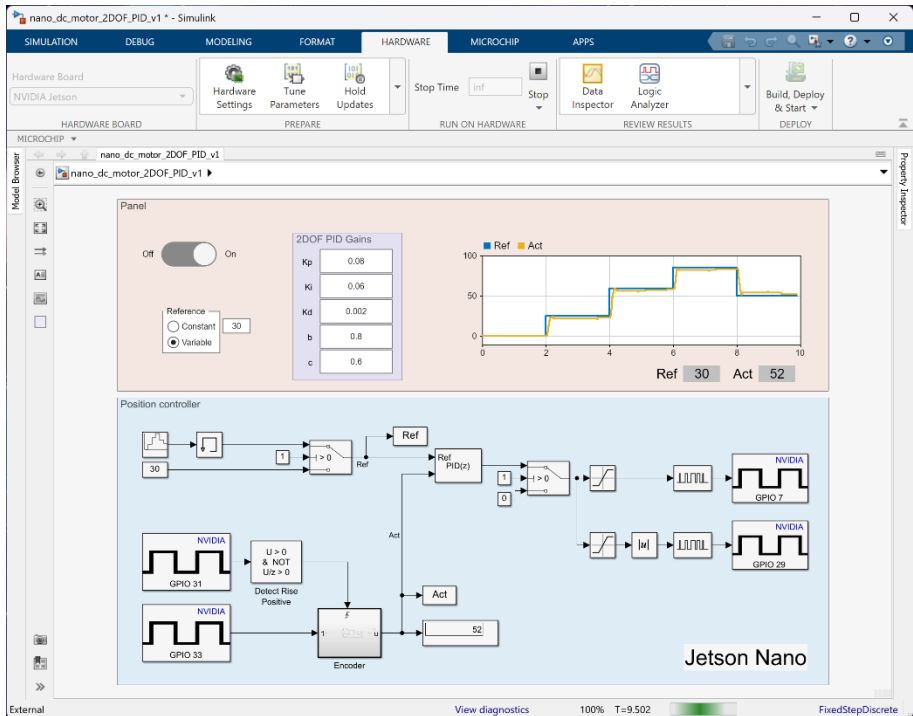


Figure 5: Test results.

In the experiment with results presented in Figure 5, the 2DOF PID gains were selected as $K_p = 0.08$, $K_i = 0.06$, $K_d = 0.02$, $b = 0.8$ and $c = 0.6$. It is observed that the system operates stably even with these unoptimized parameters. It can be stated that the system would perform much more effectively with optimized gains.

REFERENCE

- Alfaro, V. M., & Vilanova, R. (2013). Simple robust tuning of 2DoF PID controllers from a performance/robustness trade-off analysis. *Asian Journal of Control*, 15(6), 1700–1713.
- Bilgen, M., & Ilten, E. (2026). Conformable Fractional Order PID Trajectory Tracking Controller for a Quadrotor Unmanned Aerial Vehicle. *2026 11th International Conference on Recent Advances in Air and Space Technologies (RAST)*, 1–6. IEEE.
- Bingi, K., Ibrahim, R., Karsiti, M. N., Hassan, S. M., & Harindran, V. R. (2018). A comparative study of 2DOF PID and 2DOF fractional order PID controllers on a class of unstable systems. *Archives of Control Sciences*, 635–682.
- Bingi, K., Ibrahim, R., Karsiti, M. N., Hassan, S. M., & Harindran, V. R. (2019). Real-time control of pressure plant using 2DOF fractional-order PID controller. *Arabian Journal for Science and Engineering*, 44, 2091–2102.
- Guras, R., Strambersky, R., & Mahdal, M. (2022). The PID and 2DOF control of the integral system-influence of the 2DOF parameters and practical implementation. *Measurement and Control*, 55(1–2), 94–101.
- Ilten, E. (2025a). 2-DOF PID Position Control Application for DC Motor with Arduino and Simulink External Mode. *4th International Conference on Trends in Advanced Research*, 166–171.
- Ilten, E. (2025b). DC Motor Wireless Speed Control Panel Design for Android Device with Using Arduino Nano 33 IoT. In *Pioneer Research in Engineering* (pp. 31–41).
- Ilten, E. (2025c). Implementation of Tilt Integral Derivative (TID) Position Controller for DC Motor System with Using Simulink and Arduino Nano 33 IoT. *8th International Conference on Applied Engineering and Natural Sciences*, 309–314.
- Ilten, E. (2025d). Raspberry Pi Touch Panel Design for DC Motor Control System. *5th International Conference on Frontiers in Academic Research*, 1, 256–262.
- Ilten, E. (2026a). Fractional Order PID Position Control Application for DC Motor with Jetson Nano Platform. *8th International Conference on Engineering, Natural and Social Sciences*, 262–272. All Sciences Academy.
- Ilten, E. (2026b). Position Control Application of a DC Motor with Jetson Nano via Simulink Embedded Coder. *4th International Conference on Pioneer and Innovative Studies*, 233–238.
- Ilten, E. (2026c). Real-Time Implementation of Deep Learning Based Leaf Deficiency Classification on Raspberry Pi Platform. *International Journal of Advanced Natural Sciences and Engineering Researches*, 10(4), 89–100.
- Ilten, E. (2026d). Sliding Mode Position Control Application of a DC Motor based on

- Jetson Nano Platform with Simulink. *9th International Conference on Applied Engineering and Natural Sciences*, 258–263. All Sciences Academy.
- Ilten, E., & Bilgen, M. (2026a). Drone Control Panel Design on Simulink External Mode for Pixhawk-Based Quadrotor. *9th International Conference on Scientific and Academic Research*, 143–148. All Sciences Academy.
- Ilten, E., & Bilgen, M. (2026b). Pixhawk-Based Quadrotor Design and Sensor-Actuator Testing via Simulink External Mode. *International Journal of Advanced Natural Sciences and Engineering Researches*, 10(5), 88–99.
- Ilten, E., Duzgun, E., & Gullu, S. I. (2026). Deep Learning Model for Liquid Level Estimation via Webcam and its Implementation with Raspberry Pi. In *Sustainable and Trending Studies in Engineering* (pp. 44–55). All Sciences Academy.
- Küçükdermenci, S. (2024). Enkoderli DC Motorda PWM Kontrolü. In *Mühendislikte Yeni Trendler ve Sınırlar* (pp. 223–230). All Sciences Academy.
- Küçükdermenci, S. (2025a). Modeling and Simulation of DC Motor Speed PID Control Using Outseal with LabVIEW Integration. *International Journal of Advanced Natural Sciences and Engineering Researches*, 9, 171–179.
- Küçükdermenci, S. (2025b). Real-time PID-Based DC Motor Position Control Interface Using LabVIEW and Proteus Co-Simulation. *1st International Conference on Pioneer and Academic Research*, 274–281.
- Küçükdermenci, S., & Ilten, E. (2025). Arduino Based Fire Extinguisher Vehicle Design and Application. *International Journal of Advanced Natural Sciences and Engineering Researches*, 9(2), 223–232. <https://doi.org/10.5281/zenodo.14897860>
- MathWorks. (2026). *MATLAB documentation*. The MathWorks.
- Valladares, S., Toscano, M., Tufiño, R., Morillo, P., & Vallejo-Huanga, D. (2021). Performance evaluation of the Nvidia Jetson Nano through a real-time machine learning application. *International Conference on Intelligent Human Systems Integration*, 343–349. Springer.
- Zhang, Z.-D., Tan, M.-L., Lan, Z.-C., Liu, H.-C., Pei, L., & Yu, W.-X. (2022). CDNet: A real-time and robust crosswalk detection network on Jetson nano based on YOLOv5. *Neural Computing and Applications*, 34(13), 10719–10730.

Encoding Inertial Reference Unit Data Using ARINC 429 and Software-Based Verification

Murat EKİCİ

Lecturer, PhD, Dokuz Eylul University, Efes Vocational School, Department of Transportation Services,
Izmir, Türkiye murat.ekici@deu.edu.tr | ORCID: 0000-0002-8875-0775

ABSTRACT

An inertial reference unit (IRU) is a safety-relevant avionics source that continuously supplies aircraft attitude, angular-rate, and acceleration data. Reliable use of these data requires the data dictionary, ARINC 429 word structure, transmission timing, and receiver-side checks to be designed together. This chapter presents a method and system-design approach that links requirements to verification for transmitting IRU data over ARINC 429. A representative nine-parameter interface was defined, covering binary number representation (BNR), binary-coded decimal (BCD), discrete encoding, source/destination identifier, sign/status matrix, odd parity, update rates, and semantic checks. The Python prototype was evaluated with 18 unit tests, 2,000 samples per BNR parameter, 46 BCD/discrete conformance cases, and an exhaustive receiver-side scan of 4,752 unique single- and double-bit corruptions across nine valid words. Maximum quantization error did not exceed the half-resolution bound. Parity detected every single-bit corruption but no double-bit corruption. End-to-end validation rejected 2,749 of 4,464 double-bit corruptions (61.58%) because of label, status/encoding, discrete-combination, or physical-range violations. The defined schedule produced channel utilizations of 19.296% at 100 kbit/s and 154.368% at 12.5 kbit/s; therefore, it is infeasible on the low-speed channel. The findings show that ARINC 429 assurance cannot be reduced to parity alone and must combine ICD consistency, value validity, freshness, and traceability. The study is limited to the logical word layer; physical BPRZ signalling, electromagnetic compatibility, environmental qualification, and certification compliance require separate verification.

Keywords: inertial reference unit, ARINC 429, avionics data bus, BNR, fault injection, software verification

INTRODUCTION

Navigation and flight-control functions in modern aircraft depend on many sensors and computing units sharing data on a consistent time base. An inertial measurement unit produces raw gyroscope and accelerometer measurements, whereas the inertial reference function converts them into usable attitude, rate, and acceleration information through calibration, axis transformation, error estimation, and integration. Computational errors and sensor biases can accumulate over time in strapdown architectures (Titterton & Weston, 2004; Groves, 2013). Consequently, not only the estimation algorithm but also the resolution and validity indications used to deliver the information to consumers become safety-relevant system-design variables.

ARINC 429 is a long-established, unidirectional, and strongly deterministic interface for digital data transfer between line-replaceable units in commercial aircraft. Despite its relatively low data rate, its simple topology, bounded receiver count, and mature certification history have sustained its

use, particularly for distributing sensor and status information. Protocol simplicity, however, does not eliminate implementation responsibility. A 32-bit word may be formally valid yet semantically incorrect if the label, source/destination identifier, data field, sign/status matrix, or parity field is misinterpreted (ARINC, 2019a, 2019b).

A substantial portion of the literature focuses on FPGA implementation of the ARINC 429 core, multichannel controllers, or laboratory test cards. Mukherjee et al. (2011) addressed a multichannel transmitter controller, Tamrakar et al. (2016) developed an Ethernet-based analyzer, and Vyas et al. (2016) presented an FPGA-based system-on-chip approach. In their UVM studies, Ayaz et al. (2021, 2023) placed requirements-based hardware verification in the context of DO-254, whereas Zuo et al. (2024) showed that validity decisions for continuously buffered data can become complicated when multiple sources are involved. Collectively, these studies indicate that successful bit transfer alone does not demonstrate correct avionics functionality.

The contribution of this chapter is to treat the IRU–ARINC 429 interface not merely as an encoding problem but as an integrated design problem comprising measurement definition, timing, receiver assurance, and reproducible test evidence. The proposed method addresses four research questions: (i) with what resolution and range should engineering units be placed in the 19-bit data field, (ii) which fault classes are covered by word-level odd parity and where does it remain insufficient, (iii) how do semantic receiver checks extend fault coverage, and (iv) are the selected transmission rates consistent with channel capacity? These questions were evaluated using an executable Python prototype and numerical experiments.

AVIONICS CONTEXT OF INERTIAL REFERENCE UNIT DATA

IRU functional chain

An inertial reference chain begins with three-axis gyroscopes and three-axis accelerometers. Gyroscopes measure body angular rates, whereas accelerometers measure specific force. After compensating for error terms such as scale factor, axis misalignment, temperature effects, and bias, the measurements are integrated within the navigation equations. Depending on application requirements, the output may be provided as Euler angles, quaternions, a direction cosine matrix, body angular rates, linear accelerations, velocity, or position. For every quantity selected for the interface, the physical definition, coordinate frame, sign convention, unit, operating range, resolution, update period, and invalid-data behavior must be stated explicitly.

Coordinate-frame ambiguity is among the most hazardous systematic interface errors. For example, if a yaw rate defined as positive by the right-hand rule is interpreted with a different axis or sign convention by the receiving system, the data word can remain fully valid with respect to parity. Similarly, if the ICD does not specify whether heading is represented over $[0,$

360) degrees or as a signed angle, discontinuities can arise at the wrap boundary. The ARINC 429 encoder should therefore be treated not as a generic packing routine appended to a navigation algorithm but as an executable part of the contract between the data producer and consumer.

For IRU data, temporal behavior is as important as magnitude. Rapidly varying quantities such as angular rate and vertical acceleration require short update periods, whereas alignment time or status words may be transmitted less frequently. The receiver should monitor the age of the last valid word and must not silently continue using the previous value after the specified freshness window has expired. This rule identifies a stalled data source even when the retained value appears to remain within range. Figure 1 shows the information flow from physical measurement to receiver validation and the cross-cutting role of the ICD and requirements baseline.

End-to-end path from IRU data generation to receiver validation

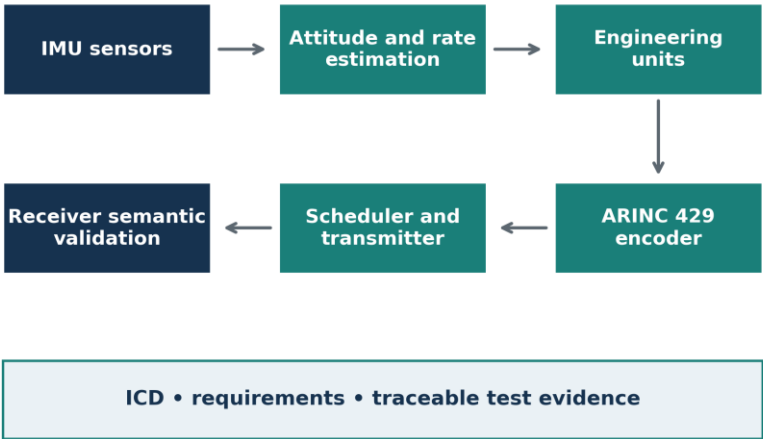


Figure 1. End-to-end generation, encoding, and validation architecture for IRU data (created by the author).

Requirements and ICD perspective

An interface control document (ICD) defines the physical and logical contract between two devices. A robust IRU ICD should specify, at minimum, channel identity, bit rate, wire direction, octal label representation, SDI usage, data encoding, bit weight or scale, unit, valid range, SSM interpretation, transmission period, timeout threshold, and startup behavior. When any of these items is missing, developers make assumptions; inconsistent assumptions become costly and difficult-to-diagnose integration faults. In accordance with the system-development logic of ARP4754B, interface requirements should be derived from higher-level aircraft functions and remain traceable to software and hardware items (SAE International, 2023a).

Table 1. Mapping of requirements to evidence for the IRU–ARINC 429 interface.

Requirement area	Question addressed in the ICD	Verification evidence
Semantics	What are the quantity, axis, and sign convention?	Reference vectors and boundary cases
Encoding	BNR, BCD, or discrete; what is the bit weight?	Known-word test vectors
Status	What do the SSM codes mean for this data type?	Positive and negative tests for each state
Timing	What are the period, jitter, and timeout limit?	Timestamped transmission/reception log
Integrity	Which semantic checks supplement parity?	Fault injection and coverage matrix
Physical layer	What are the rate, amplitude, rise time, and connection?	Oscilloscope/analyzer and environmental tests

ARINC 429 ARCHITECTURE AND WORD STRUCTURE

Topology and physical layer

An ARINC 429 link provides unidirectional communication between one transmitter and one or more receivers. A separate channel is required for data in the opposite direction. This approach simplifies timing because it does not require shared-medium access or collision resolution, although it can increase wiring and channel count. The standard defines low- and high-speed classes; 12.5 kbit/s and 100 kbit/s are commonly used in practice. The scheduling analysis in this study evaluates both rates (ARINC, 2019a).

Physical transmission uses balanced bipolar return-to-zero signalling over two conductors. Logical ONE and ZERO are represented by opposite differential polarities, whereas NULL is represented by a differential level near zero. This three-level timing contributes to identifying bit boundaries within a word. The 32-bit integer produced by the software prototype is not, however, the physical waveform itself. Line-driver and receiver thresholds, rise and fall times, cable impedance, common-mode behavior, and electromagnetic compatibility constitute a separate level of hardware verification. DO-160G frames environmental conditions and test procedures, whereas ARINC 429 specifies the interface-specific electrical characteristics (RTCA, 2010; ARINC, 2019a).

Another implementation risk concerns label-bit representation. In ARINC 429, the label occupies bits 1–8; however, the physical serial transmission order and the octal or binary representation used by software tools can create the appearance of bit reversal. An octal label displayed by an analyzer need not appear in the same textual direction as the corresponding byte in memory. The ICD should therefore provide not only the label number but also a golden test word and the expected trace. The prototype uses the logical word layout;

when it is connected to a physical transmitter, the label-conversion contract of the driver library must be verified separately.

The 32-bit word



Figure 2. ARINC 429 logical word fields and bit numbers (drawn by the author based on ARINC 429P1-19).

Table 2. Design interpretation of ARINC 429 word fields.

Bit(s)	Field	Use in this study	Primary risk
1–8	Label	Parameter type	Octal-display/bit-order confusion
9–10	SDI	Source or subsystem selection; default 00	Mismatch with receiver expectations
11–29	Data	BNR, BCD, or discrete data	Scale, sign, or bit-weight error
30–31	SSM	Encoding-specific validity/sign state	Confusion between BNR and BCD meanings
32	Parity	Odd parity over the entire word	Failure to detect an even number of bit changes

The label field selects the meaning of the word within the data dictionary. The source/destination identifier (SDI) may distinguish a source or destination in some architectures, but its use is interface-specific. The 19-bit data field carries numeric or discrete information. The sign/status matrix (SSM) should not be interpreted merely as a sign field: depending on the encoding type, it may indicate normal operation, failure warning, no computed data, sign, or another state. Interpretation must therefore depend on the data type and ICD rather than on a universal SSM dictionary.

The parity bit is selected so that the total number of ones in the 32-bit word is odd. If any single bit changes, the odd/even property reverses and the fault is detected. Two simultaneous bit changes can restore the original parity. Parity also cannot detect an incorrect label transmitted with correct parity, stale data within the valid range, or a value packed with an incorrect scale.

Parity is therefore a fault-detection layer, not authentication, error correction, or evidence of semantic correctness.

DATA ENCODING METHOD

BNR encoding and quantization

Binary number representation (BNR) is suitable for continuous quantities such as attitude angles, angular rates, and acceleration. In this study, the data field is interpreted as a 19-bit two's-complement signed integer. For engineering value x and resolution Δx , the encoder first generates integer q , and the decoder reconstructs an approximate value using the same scale:

$$\begin{aligned} q &= \text{round}_e(x / \Delta x), & \hat{x} \\ &= q \cdot \Delta x \end{aligned} \quad (1)$$

In this study, the round_e operator denotes round half to even: exact half steps are rounded to the nearest even integer. Thus, 0.5 raw steps rounds to 0, 1.5 to 2, -0.5 to 0, and -1.5 to -2 . Scaling is performed with Decimal arithmetic to prevent binary floating-point representation from altering half-step decisions. In an operational implementation, the rounding policy must be defined explicitly in the ICD and applied consistently by the encoder and decoder.

When the defined rounding policy is applied without overflow, the absolute error caused solely by quantization satisfies the following bound:

$$|e| = |x - \hat{x}| \leq \Delta x / 2 \quad (2)$$

Range selection must satisfy two conditions simultaneously. First, it must represent the maximum value expected from the sensor and aircraft dynamics. Second, the selected resolution must not exceed the capacity of the 19-bit signed integer. Overflow is not converted into wraparound; the prototype rejects it explicitly. This behavior is safer than generating an apparently plausible but physically incorrect word. Boundary values, points on both sides of zero, and the maximum positive and negative raw values were used as separate unit-test vectors.

BCD and discrete data

Binary-coded decimal (BCD) can be used for counter and time data when human-readable decimal digits must be preserved. In the prototype, alignment time is encoded in a five-digit arrangement with 0.1 s resolution. The decoder checks the permitted range of 0–9 for every digit; an invalid four-bit group (nibble) is not silently converted into a number. The readability provided by BCD is obtained at lower numeric efficiency than BNR for the same number of bits.

A discrete field is suitable for carrying flags such as IRU normal operation, alignment, degraded operation, and failure. In the representative dictionary, bits 11–14 are assigned to normal, aligning, degraded, and failed states, respectively, and an exactly-one-active-bit rule is enforced. Correct

parity does not legitimize two contradictory status bits being active simultaneously; the receiver must validate the combination.

SSM, SDI, and parity policy

For normal numeric data, this study evaluates the SSM using an acceptance table that depends on encoding type. The receiver checks parity first, then the label, followed by the label-dependent relationship between encoding and SSM. This sequence is not merely an optimization; it separates fault classes. A corrupted word does not reach the decoder, an incorrect scale is not applied to an unknown label, and a numeric value with invalid status is not delivered to the application. Although SDI is set to 00 by default, the source-multiplexing rule in an operational integration must be derived from the ICD.

Design notice: The label values used in this chapter are instructional and experimental; they are neither official ARINC assignments nor values from the ICD of a particular aircraft. In an airworthy implementation, labels, SDI, SSM, scales, and transmission rates must be obtained from an approved interface control document.

PROPOSED IRU DATA DICTIONARY

A representative data dictionary comprising seven BNR parameters, one BCD parameter, and one discrete parameter was defined for the experiment. Labels are shown in octal form. Resolutions of 0.01° for roll and pitch angles, 0.01°/s for body angular rates, and 0.001 g for vertical acceleration were selected. Heading is represented over [0, 359.99]. Dynamic parameters are transmitted at periods of 10–40 ms, the status word at 100 ms, and alignment time at 1,000 ms.

Table 3. Experimental IRU data dictionary. Labels are representative, non-official assignments.

Parameter	Label *	Code	Range	Resolution	Period
Roll angle	360 _s	BNR	−180...180°	0.01°	20 ms
Pitch angle	361 _s	BNR	−90...90°	0.01°	20 ms
Heading	362 _s	BNR	0...359.99°	0.01°	40 ms
Roll rate	363 _s	BNR	−500...500°/s	0.01°/s	10 ms
Pitch rate	364 _s	BNR	−500...500°/s	0.01°/s	10 ms
Yaw rate	365 _s	BNR	−500...500°/s	0.01°/s	10 ms
Vertical acceleration	366 _s	BNR	−16...16 g	0.001 g	10 ms
Alignment time	367 _s	BCD	0...7,999.9 s	0.1 s	1,000 ms
IRU status	370 _s	Discrete	Bits 11–14	1 bit	100 ms

The resolutions in the dictionary are example system requirements, not mandatory protocol values. For an operational IRU interface, the performance budget should consider sensor noise, navigation accuracy, consumer-algorithm sensitivity, and word-field capacity together. An unnecessarily fine scale narrows the representable range, whereas an excessively coarse scale may cause quantization error to consume a meaningful portion of the system budget. Similarly, selecting an unnecessarily high update rate can increase channel and processing loads without increasing the actual information bandwidth.

SOFTWARE AND SYSTEM DESIGN

Layered software architecture

The prototype is divided into four responsibility layers. The data-dictionary layer stores the parameter name, label, encoding type, range, resolution, and transmission period. The core layer implements BNR, BCD, and discrete encoding; word assembly; field extraction; and odd parity. The validation layer performs label allowlisting, SSM, parameter-specific physical-range, and freshness checks. The experiment layer generates boundary values and samples uniformly distributed across the full range, injects faults, and reports metrics.

An engineering value supplied to the encoder is checked first for numeric type and finiteness and then against the defined physical range. In BNR, no word is generated if the scaled integer exceeds the 19-bit signed bounds. In BCD, each decimal digit is assigned to its field; for discrete data, only defined bit masks are permitted. After the word fields are assembled, the number of ones in bits 1–31 is evaluated and bit 32 is set so that the total is odd.

Receiver-side validation is applied in the sequence parity → label → data-type-specific SSM → decoding → physical range → freshness. Freshness is derived not from a timestamp contained in the word but from the time of the last valid reception recorded by the receiver. Setting the timeout threshold equal only to the nominal period can produce false alarms under reasonable jitter or scheduler delay, whereas an excessively wide threshold prolongs hazardous use of stale data. The threshold should be justified through the system safety assessment and end-to-end latency budget.

The executable prototype is a research tool rather than a certification product. Python was selected to make bit fields visible, simplify test generation, and improve experimental reproducibility. The language, runtime, compiler, coding standard, and tool qualification used for airborne software must be managed separately under DO-178C life-cycle objectives. Similarly, if the ARINC core is implemented in an FPGA, DO-254 design assurance, independent verification, and coverage evidence are required (RTCA, 2000, 2011a; Ayaz et al., 2023).

Timing and channel utilization

For m distinct words with transmission frequency f_i , a word length of 32 bits, an assumed interword gap of g bit times, and channel rate R_β , the approximate utilization of the ideal schedule is calculated as follows:

$$U = [\sum_i (32 + g) \cdot f_i] / R_\beta \tag{3}$$

Table 4. Utilization of the same transmission schedule at two ARINC 429 channel rates.

Channel rate	Utilization	Engineering assessment
12.5 kbit/s	154.368%	The schedule exceeds capacity and is infeasible
100 kbit/s	19.296%	Nominal capacity margin is available

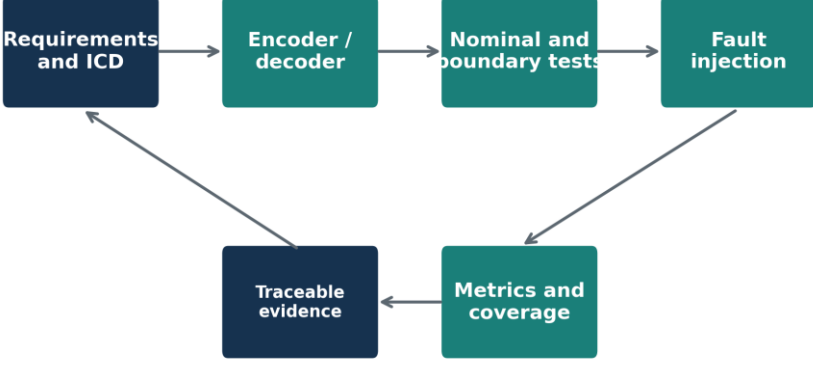
With $g = 4$, the nine words in Table 3 produce a total utilization of 19.296% on a 100 kbit/s channel. The same schedule reaches 154.368% on a 12.5 kbit/s channel and therefore cannot be assigned to the low-speed link; transmission periods must be relaxed or traffic distributed to another channel. These values are only an initial logical estimate. Actual transmitter FIFO behavior, driver-call latency, periodic jitter, other labels on the same channel, reconfiguration, and maintenance words must be added to the assessment. Low channel utilization alone also does not guarantee that a critical word meets its deadline; the worst-case response time of the schedule must be examined separately.

Table 5. Evidence gap between prototype decisions and an airworthy implementation.

Design item	Prototype decision	Additional evidence for airborne use
Time base	Software timestamp	Clock-accuracy and synchronization budget
Scheduling	Fixed parameter periods	Worst-case latency and jitter analysis
Buffering	Immediate logical processing	FIFO-depth, overflow, and backpressure tests
Parity	Software-generated odd parity	Electrical verification with line receiver and analyzer
Fault logging	Exception and counter	Maintenance message, health management, and persistence

VERIFICATION METHOD

Experimental design



Failed test → controlled update of requirement, design, or test

Figure 3. Requirements-based verification loop with fault injection (created by the author).

Verification combines fixed requirements-based tests, full-range sampling, and receiver-side bit-corruption scans. The tests cover BNR boundaries, overflow, and half-step rounding; BCD boundaries and every invalid nibble value; four valid discrete states and all invalid state combinations; parity; unknown labels; invalid SSM; physical range; and timeout behavior. All 18 automated unit tests passed. All 46 BCD and discrete-encoding conformance cases produced the expected outcomes.

For each BNR parameter, deterministic boundary points at the minimum, maximum, zero, and around half-resolution steps were added to the quantization sample set; the remaining values were generated uniformly over the entire defined physical range. A total of 2,000 round-trip samples per parameter, or 14,000 samples overall, were used to calculate MAE, RMSE, and maximum absolute error:

$$MAE = (1/n) \cdot \sum_i |x_i - \hat{x}_i| \quad (4a)$$

$$RMSE = \sqrt{[(1/n) \cdot \sum_i (x_i - \hat{x}_i)^2]} \quad (4b)$$

In the fault-injection experiment, every single-bit and double-bit corruption of nine valid words containing BNR, BCD, and discrete encodings was scanned from the receiver perspective. The single-bit space comprised $9 \times 32 = 288$ cases, and the double-bit space comprised $9 \times C(32,2) = 4,464$ unique cases. This exhaustive design eliminates the risk that random sampling

with replacement might repeat the same bit pairs or omit some combinations. For every case, the parity-only outcome was recorded first, followed by the end-to-end outcome including label, SSM/encoding, discrete-combination, and physical-range checks. The experiment covers only independent bit inversions; burst errors, incorrect bit rate, interword-gap violations, amplitude degradation, and loss of synchronization were not modeled.

The semantic-fault experiments comprise four controlled cases: a label absent from the dictionary, an SSM rejected by the definition, a decoded value outside the physical limits, and a timeout after the last valid reception. Each case was executed once and the expected rejection reason was recorded. These individual tests are deterministic requirements tests that demonstrate the existence of the validation path, not estimates of statistical success rates.

Validity and reproducibility

The experiments were rerun in Python 3.12.14. A fixed seed of 429 was used for quantization sampling; the exhaustive single- and double-bit experiments used no randomness. The execution command is `python -m arinc429_iru.experiment`. Each run produces `summary.json`, `quantization_metrics.csv`, and `fault_detection.csv`. The research package, comprising the parameter dictionary, tests, result files, and source code, is maintained by the author with version integrity and can be provided upon reasonable request. Nevertheless, the encoder and decoder share the same scale constants, which can conceal a common-cause error. Independent test vectors, manually calculated parity examples, and subsequent cross-validation with a commercial ARINC analyzer are needed to reduce this threat. Tool-based test evidence should be assessed in the context of DO-330 for defects that might be missed if the tool malfunctions (RTCA, 2011b).

RESULTS

Quantization accuracy

Table 6. BNR round-trip quantization results (n = 2,000 per parameter).

Parameter	Resolution	MAE	RMSE	Maximum error	Theoretical bound
Roll	0.01°	0.002543°	0.002930°	0.005000°	0.005°
Pitch	0.01°	0.002500°	0.002896°	0.005000°	0.005°
Heading	0.01°	0.002506°	0.002904°	0.005000°	0.005°
Roll rate	0.01°/s	0.002544°/s	0.002930°/s	0.005000°/s	0.005°/s
Pitch rate	0.01°/s	0.002492°/s	0.002874°/s	0.005000°/s	0.005°/s
Yaw rate	0.01°/s	0.002476°/s	0.002850°/s	0.005000°/s	0.005°/s
Vertical acceleration	0.001 g	0.000249 g	0.000287 g	0.000500 g	0.0005 g

For every parameter, the maximum absolute error remained within the theoretical half-resolution bound. MAE values near one-half of that bound are consistent with the expected quantization residuals of values uniformly distributed over the operating range. This result demonstrates encoding accuracy but does not, by itself, provide evidence of physical sensor accuracy.

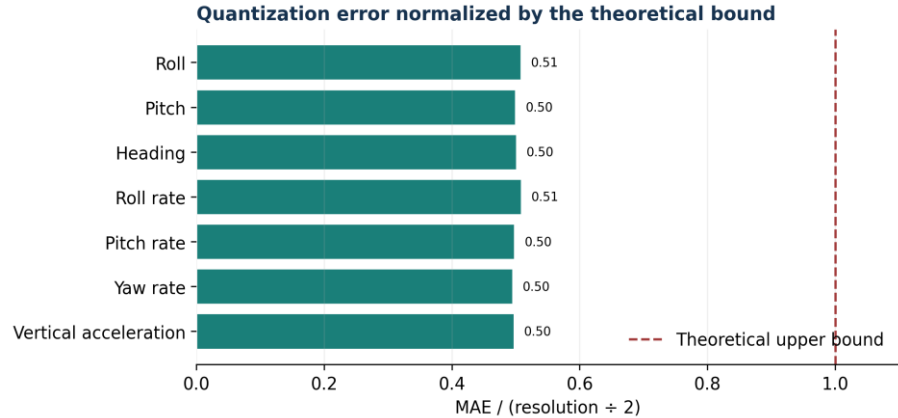


Figure 4. Ratio of mean absolute quantization error to the half-resolution bound.

BCD and discrete-data conformance

Table 7. BCD and discrete-encoding conformance results.

Conformance group	Cases	Passed	Expected behavior
BCD boundary round trip	6	6	Correct encoding and decoding
Invalid BCD nibble values	24	24	Rejection of A–F values
Valid discrete one-hot states	4	4	Correct status decoding
Invalid discrete combinations	12	12	Rejection of zero/multiple active states

In the BCD boundary experiment, alignment-time values of 0.0, 0.1, 9.9, 10.0, 999.9, and 7,999.9 s were all encoded and decoded correctly. The invalid-digit experiment generated 24 unique words by placing hexadecimal values A–F individually into each of the four BCD nibble fields; the receiver rejected all of them. In the discrete-state experiment, the four valid one-hot words representing normal, aligning, degraded, and failed states were decoded correctly. The all-zero state and 11 combinations with two or more active bits were rejected. Thus, all 46 conformance cases produced the expected outcome.

Fault detection

Table 8. Receiver-based bit-corruption results. *Single-case deterministic test; not to be interpreted as a statistical success rate.

Fault/check	Trials	Parity	End-to-end	Explanation
Single-bit corruption	288	288 (100%)	288 (100%)	Rejected at the parity stage
Double-bit corruption	4,464	0 (0%)	2,749 (61.58 %)	1,326 label; 738 semantic; 685 range
Unknown label	1	—	1/1*	Label allowlist
Invalid SSM	1	—	1/1*	Encoding-specific status
Physical range	1	—	1/1*	181° roll value
Data timeout	1	—	1/1*	Freshness monitoring

Parity detected every single-bit corruption and no double-bit corruption, which is a structural consequence of odd parity. When the same double-bit space was submitted to receiver validation, however, 2,749 words (61.58%) were rejected: 1,326 became unknown labels, 738 violated SSM/encoding or discrete-combination rules, and 685 violated physical ranges. The remaining 1,715 words (38.42%) passed the defined checks. These percentages are exact exhaustive results for 4,464 unique double-bit corruptions of the selected nine nominal words; they must not be interpreted as a general ARINC 429 fault-detection probability. Therefore, “not detected by parity” and “not detected by the system” are not equivalent outcomes.

Semantic checks target fault dimensions not covered by parity. The label allowlist identifies an unexpected data type; SSM and encoding checks identify invalid status; the physical-range check identifies a value that conflicts with the ICD; and freshness monitoring identifies a value that is no longer being updated. Because some double-bit corruptions still pass these checks, the layers provide neither error correction nor cryptographic integrity.

COMPARATIVE DISCUSSION WITH THE LITERATURE

Hardware implementation and verification trends

The dominant direction of ARINC 429 research is implementation of transmitter and receiver logic on FPGAs or dedicated interface hardware. Kamaraju et al. (2010) designed and simulated a VHDL-based core; Liu et al. (2015) presented a PC/104-based structure combining an FPGA with a dedicated transceiver integrated circuit; and Vyas et al. (2016) developed an SoC approach between RS-232 and ARINC 429. Türk and Erdem (2023) evaluated six receiver and three transmitter channels using experimental boards in a multichannel design, whereas Shen et al. (2025) proposed a multichannel simulation card adaptive to 12.5 and 100 kbit/s. These studies

provide strong evidence on real-time bit flow and resource utilization, but engineering values, data freshness, and consumer-specific semantic validation are not addressed with the same level of detail in every design.

In verification research, Yang and Jin (2020) tested repeater, transmitter, and receiver modes using three observation methods; Ayaz et al. (2021, 2023) linked requirements, assertions, and coverage through UVM in the context of DO-254. Madhura et al. (2023) reported 98.43% functional coverage in a loopback UVM environment. Çiçek and Çiçek (2025) developed a portable verification module using a USB–SPI bridge and a dedicated ARINC transceiver. The present chapter does not replace these hardware approaches; it details a data-dictionary and software-oriented reference-model layer that can precede them.

The validity-decision problem for continuously buffered data from multiple sources examined by Zuo et al. (2024) shows that inspecting only the current word at the receiver may be insufficient. Validity can depend on source identity, time of last reception, expected update rhythm, and system mode. The freshness monitoring used in this chapter addresses the same problem on a smaller scale.

ARINC 429, ARINC 664, and ARINC 825

Selecting a data bus for a new aircraft architecture is not merely a bandwidth comparison. ARINC 429 is effective for sensor distribution because of its simple single-transmitter links and predictable word flow. ARINC 664 Part 7/AFDX carries much larger data volumes in an integrated modular avionics environment through virtual links and traffic regulation over switched full-duplex Ethernet. ARINC 825 standardizes CAN-based communication for airborne use and includes CAN FD capabilities in its current revision (ARINC, 2018). A migration decision should jointly consider bandwidth, wiring, segregation, determinism, certification experience, the supplier ecosystem, and interfaces of existing line-replaceable units.

Table 9. Qualitative design comparison of avionics data buses.

Feature	ARINC 429	ARINC 664 P7 / AFDX	ARINC 825
Basic structure	Single transmitter, multiple receivers, unidirectional	Switched Ethernet, virtual links	CAN-based multiple nodes
Data unit	32-bit word	Ethernet frame	CAN/CAN FD frame
Bandwidth	Low	High	Medium; revision/mode dependent
Determinism	Strong with simple periodic transmission	Requires traffic contracts and network analysis	Requires priority arbitration and load analysis

Feature	ARINC 429	ARINC 664 P7 / AFDX	ARINC 825
Suitable use	Mature sensor/LRU interface	Integrated high-volume network	Distributed control and utility systems
Primary design risk	ICD/scale errors and channel proliferation	Configuration and network latency	Priority, fault containment, and load

Cybersecurity and false-data risk

The historical design of ARINC 429 provides neither cryptographic authentication nor confidentiality. Although this limitation was operationally accepted in conventional architectures with tightly controlled physical access, maintenance connections, data converters, and more connected aircraft architectures can expand the attack surface. Dorigatti et al. (2024) demonstrated attacks in which avionics data flow could be suppressed or taken over using electrically legitimate-looking signals. Trask et al. (2024) investigated the absence of authentication in ARINC 429 using a hardware-in-the-loop environment with actual avionics hardware and produced a dataset for attack detection based on physical signal characteristics.

In this context, label, range, and freshness checks are insufficient as cybersecurity mechanisms; an attacker can generate a physically plausible but incorrect value with the expected label and correct parity. Nevertheless, these checks are useful inputs to anomaly detection. Detectability can improve when source fingerprinting, line-voltage characteristics, cross-sensor consistency, the flight-envelope model, and mode-dependent update profiles are used together. The security architecture must also justify false-positive behavior, latency, and failure responses so that safety functions are not impaired.

CERTIFICATION AND ASSURANCE PERSPECTIVE

Generating a word that conforms to ARINC 429 is not equivalent to airworthiness approval. At the system-development level, ARP4754B and ARP4761A frame requirements allocation and safety assessment. DO-178C and FAA AC 20-115D, which recognizes it as an acceptable means of compliance, require software evidence; DO-254 addresses hardware; and DO-160G addresses environmental qualification. These standards require distinct but complementary evidence sets (Federal Aviation Administration, 2017; RTCA, 2000, 2010, 2011a; SAE International, 2023a, 2023b).

An example high-level requirement for an IRU–ARINC interface may state: ‘In normal navigation mode, roll angle shall be transmitted at intervals of 20 ms or less, with a resolution of 0.01° and the defined normal SSM state.’ Low-level requirements define the raw-integer calculation, bit placement, overflow behavior, and parity. Tests cover boundary values, nominal examples, invalid status, and timing behavior. When bidirectional links are established among requirements, design, code, tests, and results, the effects of change can be assessed and missing verification can be identified earlier.

DO-331 and DO-333 provide additional guidance when model-based development or formal methods are used (RTCA, 2011c, 2011d). Automatic generation of tool output does not remove verification responsibility. If a defect in a code generator, model checker, or test tool could make certification objectives appear to have been satisfied, tool qualification is addressed under DO-330. The Python prototype used in this chapter is research evidence; it is neither a qualified toolchain nor airborne code.

Table 10. Assurance frameworks and example evidence for interface development.

Assurance objective	Applicable framework	Example IRU–ARINC evidence
System requirements and allocation	SAE ARP4754B	Data-accuracy, latency, and interface budgets
Safety assessment	SAE ARP4761A	Failure conditions for lost, erroneous, and stale attitude data
Airborne software	RTCA DO-178C	Traceable encoding and receiver-validation requirements
FPGA/ASIC	RTCA DO-254	Verification of transmitter/receiver core and timing logic
Environmental qualification	RTCA DO-160G	EMI/EMC, temperature, vibration, and power tests
Tool qualification	RTCA DO-330	Assurance rationale for automated code/test tools

IMPLEMENTATION ROADMAP

The following staged path is proposed for progressing from this software-level study to a laboratory demonstration. Each stage tests the assumptions of the previous stage through independent measurement:

1. ICD baseline: Approve labels, SDI, scales, SSM, and periods from actual aircraft and LRU requirements; remove representative values.
2. Independent reference model: Generate golden words with a technology or algorithm different from the encoder and perform bit-for-bit comparison.
3. ARINC interface hardware: Transmit and receive at 12.5 and 100 kbit/s using a non-certification laboratory board; correlate logical words with analyzer traces.
4. Physical-layer measurement: Verify differential amplitude, NULL level, rise/fall time, interword gap, and cable loading with an oscilloscope.
5. Hardware-in-the-loop: Test flight profiles, IRU mode transitions, power interruptions, realignment, FIFO overflow, and wire/bit fault scenarios in real time.
6. Assurance package: Complete requirements traceability, structural and functional coverage, problem reports, configuration management, and independent review records.

Full-scale airborne hardware is not required at the first stage of this roadmap. The logical data model and fault checks can be developed on a computer, and a low-cost USB interface can be used for training and preliminary integration. Once physical compliance, environmental robustness, or certification is claimed, however, dedicated test equipment and a controlled laboratory become unavoidable. The compact USB-based solution of Çiçek and Çiçek (2025) demonstrates the feasibility of early verification tools in terms of cost and portability.

LIMITATIONS

The first limitation is that the results apply to synthetic data and the logical 32-bit word layer. The BPRZ waveform, line-driver and receiver thresholds, cable length, termination, common-mode noise, electromagnetic interference, and environmental effects were not modeled. Accordingly, the 100% single-bit detection result must not be interpreted as 100% detection of all faults occurring on the electrical link.

The second limitation is the use of representative labels and a representative data dictionary. Parameter ranges and periods were selected to demonstrate the method and were not taken from a specific aircraft type, IRU manufacturer, or approved ICD. Third, fault injection is limited to all independent single- and double-bit inversions in nine nominal words; 61.58% is an exact result only for this experimental universe. Future experiments should add burst faults, incorrect rate, corrupted interword gaps, label-bit ordering errors, source conflicts, timing jitter, and FIFO overflow.

The fourth limitation is that the encoder and decoder reside in the same Python codebase. A shared incorrect assumption can cancel itself in a round-trip test. An independent analyzer, a second reference implementation, and standards-based golden vectors would reduce this threat. Finally, the study makes no claim of compliance with DO-178C or DO-254; certification objectives require life-cycle data, independence, coverage, configuration management, and quality-assurance activities.

CONCLUSION

This chapter treated the encoding of IRU data over ARINC 429 as a method that maintains consistency among requirements, the data dictionary, timing, receiver validation, and verification evidence. The representative nine-parameter dictionary was evaluated using 18 automated tests, 46 BCD/discrete conformance cases, 14,000 BNR samples covering the full operating ranges, and 4,752 unique single- and double-bit corruptions. Every quantization error remained within the half-resolution bound. Channel utilization for the same transmission schedule was 19.296% at 100 kbit/s and 154.368% at 12.5 kbit/s, demonstrating that the low-speed channel cannot support this schedule.

Odd parity detected every single-bit corruption but no double-bit corruption. Receiver validation rejected 61.58% of the double-bit corruptions of the selected nine words because of label, semantic/encoding, or physical-range violations; 38.42% passed the defined checks. This exact percentage is specific to the experimental universe and is not a general ARINC 429 fault-detection probability. The findings show that parity coverage and system-level fault-detection coverage should be reported separately. Because cryptographic authentication is absent, these checks alone cannot prevent intentional protocol-compliant false data.

The recommended next step is to replace the representative data dictionary with an actual approved ICD, create independent golden vectors, and then proceed to physical-layer measurements using a USB or PCIe ARINC interface and analyzer. This staged approach exposes scale, sign, status, and timing errors before expensive hardware integration. For a final airborne product, the ARP4754B/ARP4761A system-assurance process and the required DO-178C, DO-254, and DO-160G evidence must be completed in accordance with project objectives.

REFERENCES

- Aeronautical Radio, Inc. (ARINC). (2018). ARINC Specification 825-4: General standardization of controller area network (CAN) bus protocol for airborne use. SAE Industry Technologies Consortia.
- Aeronautical Radio, Inc. (ARINC). (2019a). ARINC Specification 429P1-19: Digital information transfer system (DITS), Part 1—Functional description, electrical interface, label assignments and word formats. SAE Industry Technologies Consortia.
- Aeronautical Radio, Inc. (ARINC). (2019b). ARINC Specification 429P2-17: Digital information transfer system (DITS), Part 2—Discrete word data standards. SAE Industry Technologies Consortia.
- Ayaz, I., Sakarya, U., and Hökelek, I. (2021). Universal verification methodology application of ARINC-429 for airborne electronic hardware certification. In 2021 13th International Conference on Electrical and Electronics Engineering (ELECO) (pp. 162–166). IEEE. <https://doi.org/10.23919/ELECO54474.2021.9677697>
- Ayaz, I., Sakarya, U., and Hökelek, I. (2023). Universal verification methodology-based airborne electronic hardware certification approach for ARINC-429. *Aircraft Engineering and Aerospace Technology*, 95(7), 1145–1150. <https://doi.org/10.1108/AEAT-09-2022-0239>
- Çiçek, F., and Çiçek, I. (2025). A compact USB-based interface module for the verification of ARINC429 avionics systems. In 2025 6th International Conference on Communications, Information, Electronic and Energy Systems (CIEES) (pp. 1–4). IEEE. <https://doi.org/10.1109/CIEES66347.2025.11300008>
- Dorigatti, D., Strohmeier, M., Neuhaus, S., Zhou, J., and Chattopadhyay, S. (2024). Air-bus hijacking: Silently taking over avionics systems. In *Proceedings of the 10th ACM Cyber-Physical System Security Workshop* (pp. 53–63). ACM. <https://doi.org/10.1145/3626205.3659144>

- Federal Aviation Administration. (2017). Advisory Circular 20-115D: Airborne software development assurance using EUROCAE ED-12() and RTCA DO-178(). U.S. Department of Transportation.
- Groves, P. D. (2013). Principles of GNSS, inertial, and multisensor integrated navigation systems (2nd ed.). Artech House.
- Kamaraju, M., Tilak, A. V. N., Kishore, K. L., and Baburao, K. (2010). VHDL implementation and verification of ARINC-429 core. arXiv. <https://doi.org/10.48550/arXiv.1011.5374>
- Liu, H., Gu, C., Ma, S., Li, X., and He, G. (2015). ARINC-429 airborne communications transceiver system based on FPGA implementation. In International Conference on Information Sciences, Machinery, Materials and Energy (pp. 42–46). Atlantis Press. <https://doi.org/10.2991/icismme-15.2015.8>
- Madhura, R., Hirebasur Krishnappa, K., Shashidhar, R., Shwetha, G., Yashaswini, K. P., and Sandya, G. R. (2023). UVM methodology for ARINC 429 transceiver in loop back mode. In 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC) (pp. 1–7). IEEE. <https://doi.org/10.1109/ICMNWC60182.2023.10435932>
- Mukherjee, D., Kumar, N., Singh, K., Mondal, H., and Reddy, B. V. R. (2011). Implementation of ARINC 429 16 channel transmitter controller on FPGA. In V. V. Das, G. Thomas, and F. Lumban Gaol (Eds.), Information technology and mobile communication (Communications in Computer and Information Science, Vol. 147, pp. 245–247). Springer. https://doi.org/10.1007/978-3-642-20573-6_39
- Radio Technical Commission for Aeronautics (RTCA). (2000). DO-254: Design assurance guidance for airborne electronic hardware. RTCA.
- Radio Technical Commission for Aeronautics (RTCA). (2010). DO-160G: Environmental conditions and test procedures for airborne equipment. RTCA.
- Radio Technical Commission for Aeronautics (RTCA). (2011a). DO-178C: Software considerations in airborne systems and equipment certification. RTCA.
- Radio Technical Commission for Aeronautics (RTCA). (2011b). DO-330: Software tool qualification considerations. RTCA.
- Radio Technical Commission for Aeronautics (RTCA). (2011c). DO-331: Model-based development and verification supplement to DO-178C and DO-278A. RTCA.
- Radio Technical Commission for Aeronautics (RTCA). (2011d). DO-333: Formal methods supplement to DO-178C and DO-278A. RTCA.
- SAE International. (2023a). ARP4754B: Guidelines for development of civil aircraft and systems. SAE International.
- SAE International. (2023b). ARP4761A: Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. SAE International.
- Shen, Z., Zhang, Q., and Wang, L. (2025). Design and implementation of an ARINC429 bus simulation card. In Proceedings of the 9th International Conference on Electronic Information Technology and Computer Engineering. ACM. <https://doi.org/10.1145/3766671.3766817>
- Tamrakar, I., Ananda, C., Kumar, P. B., and Sree Ramani, P. (2016). Design and implementation of ethernet based analyzer for avionics serial bus on FPGA. In

- 2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (pp. 1637–1641). IEEE. <https://doi.org/10.1109/RTEICT.2016.7808110>
- Titterton, D. H., and Weston, J. L. (2004). Strapdown inertial navigation technology (2nd ed.). Institution of Engineering and Technology. <https://doi.org/10.1049/PBRA017E>
- Trask, C., Movit, S., Clutter, J., Clark, R., Herrera, M., and Tran, K. (2024). ARINC 429 cyber-vulnerabilities and voltage data in a hardware-in-the-loop simulator. arXiv. <https://doi.org/10.48550/arXiv.2408.16714>
- Türk, S., and Erdem, S. S. (2023). Implementation and performance analysis of multi channel ARINC 429 controller on FPGA. In 2023 10th International Conference on Electrical and Electronics Engineering (ICEEE) (pp. 405–410). IEEE. <https://doi.org/10.1109/ICEEE59925.2023.00079>
- Vyas, R., Bhattacharjee, A. K., and Khandelwal, A. (2016). Development of SoC for interfacing avionics units with ARINC 429 bus. In 2016 International Conference on Communication and Electronics Systems (ICCES) (pp. 1–5). IEEE. <https://doi.org/10.1109/CESYS.2016.7889931>
- Yang, J.-H., and Jin, T. (2020). Research and implementation of ARINC429 bus based on three verification methods. In Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing (Article 53, pp. 1–6). ACM. <https://doi.org/10.1145/3448823.3448862>
- Zuo, Z., Liang, J., and Sun, K. (2024). Research on ARINC 429 data validity check algorithm based on FPGA implementation. IET Conference Proceedings, 2024(13), 385–389. <https://doi.org/10.1049/icp.2024.2888>

Radar Vital Signs Detection by Using Optimized CNN+ GRU Model

Umut OZKAYA¹

¹ Department of Electrical and Electronics Engineering, Konya Technical University, Konya 42250, Turkey. Correspondence should be addressed to Umut Özkaya; uozkaya@ktun.edu.tr
<https://orcid.org/0000-0002-9244-0024>

ABSTRACT

The use of non-contact sensors in the detection of vital signals offers significant advantages. Heart and respiratory rate data obtained from wearable technologies may not contain sufficient information for evaluation. With the remote sensing capability of radar sensors, it is possible to analyze vital signals in individuals. In this study, vital signals obtained from radar sensors were evaluated with the end-to-end and high-performance Convolutional Neural Network (CNN) model. By using the spectrogram method, one-dimensional raw radar signals were converted into two-dimensional data with their components in the time and frequency domain. The spectrogram method should be used for training and testing of pre-trained CNN models with two-dimensional input. High-level features obtained from the trained CNN models were fused with Canonical Correlation Analysis (CCA). These features are given as an input to the block structure included in Gated Repetitive Units (GRUs) and the parameters of the GRU structures are optimized with the Whale Optimization algorithm (WOA) during the training process. The performance metrics of our optimized CNN+GRU model are 95.52% Accuracy (ACC), 94.74 % Sensitivity (SEN), 98.29% Specificity (SPE), 96.35% Precision (PRE), 95.40% F1-Score and 93.96% Matthews Correlation Coefficient (MCC). Considering the performance metrics, it has been proven that the proposed method has high performance on the data of non-contact radar technologies.

Keywords: Deep Learning; Vital Signs Detection; Spectrogram; GRU; Biomedical Signals

INTRODUCTION

Obstructive Sleep Apnea (OSA) is a deadly disease worldwide, with more sleep disturbances in males than in females [1]. OSA occurs when the air flow in the upper airway of the individual is interrupted during sleep. In addition, there is the possibility of showing symptoms at every stage of the individual's life. According to the information from the statistics, it is seen that the rate of men with OSA is two times as high as women [2]. While OSA is a chronic disease, it also triggers conditions such as is-chemic heart disease, increased prevalence of stroke, coronary artery disease, atrial fibrillation (AF), chronic heart failure (CHF), and cardiac sudden death [3]. It is thought that OSA is triggered by factors such as increased cholesterol, hypertension, type 2 diabetes, and cancer mortality in the individual [4]. In addition, OSA is predicted to cause oxygen desaturations, oxidative stress, blood pressure, heart rate changes, and interrupted sleep in the body [5].

Detection of OSA disease is possible by monitoring certain activities of the individual during the night with a polysomnography device. Although the polysomnography system is preferred in terms of reliability and quality at the diagnosis stage, it negatively affects long-term and continuous follow-up as it provides limited mobility for individuals. It may cause some irritation and stains on the body of the patient [6]. These disadvantages of polysomnography monitoring technique have led to increased demand for non-contact measurement and tracking systems.

It was discovered in the 1970s that it was possible to monitor the vital activities of the individual without contact, thanks to Doppler radars. Initially, studies on non-contact evaluation of respiratory and heart rate appeared in the literature. Then, it was proved that OSA syndrome can be detected without contact in non-ideal environment conditions.

To obtain an accurate measurement, it depends on factors such as background clutter, phase-nulling or null point, DC offsets, motion artefacts and electromagnetic interferences [7]. In addition, the operability of non-contact sleep monitoring systems is negatively affected by environment complexity, unpredictable body mobility, back-ground noise, sleep position variation, and undesired harmonics [8].

In the literature, Wang et al. examined phase difference data from WIFI devices to analyse respiration and heart rate [9]. Quayum et al. tried to detect different human movements with the radar system in a simulation environment [10]. Saluja et al. ensured an accurate measurement by eliminating the non-linear harmonics that occur in the non-contact detection of heartbeat with the gamma filter, which is a machine learning method [11]. Erdogan et al. used a 24 GHz microwave Doppler radar sensor to detect vital signs. It has been observed that the results obtained are in agreement with the results obtained with the respiratory belt [12]. Saluja et al. eliminated the harmonics in 5.8 GHz Doppler radar signals with a controlled gamma filter [13]. Chang et al. used Newtonized Orthogonal Matching Pursuit based on deep learning in predicting the vital signal. The proposed method to show its superiority over existing systems with its success in both simulation and real measurement data [14]. Lai used deep neural network (DNN) to detect vital signs obtained with 24 GHz Doppler radar [15]. Zhang tried to classify heart rate signals with 90.82% by extracting a common feature from electrocardiogram (ECG) and radar signals [16]. Cardillo et al. proposed a new method framework for detecting vital signals by reducing radar self-motion effects. In this way, small and large movements can be detected [17]. Slapnicar et al. de-signed a deep learning model to detect vital signals with contact and non-contact methods. With the proposed method, they achieved 88% accuracy in contact measurements, 83% in non-contact measurements,

and 88% with fusion processes [18]. Savage et al. analysed vital signals obtained from 75 patient individuals by using pat-tern recognition techniques. As a result, they obtained 85.8% accuracy, 70% sensitivity and 89% specificity performance metrics [19]. Tran et al. achieved an accuracy of 91.29% heart rate with their remote sensing biosensor [20]. Zaffaroni et al. designed SleepMinder's AHI to achieve a sensitivity of 89% and a specificity of 94% [21]. Lai et al. performed deep learning methods on data obtained from OSA patients [22]. Han et al. proposed CNN+LSTM structure to detect sleep apnea with above 90% accuracy [23]. Chen et al. detected apnea in two different environments as hospital and home by using millimeter wave radar. They reached 93% accuracy rate performed with machine learning methods [24]. Baboli et al. studied wireless-based sleep apnea detection with overall sensitivity of 86%, the specificity of 91% and accuracy of 92% [25]. Kwon et al. proposed CNN-LSTM to detect Apnea and Non-Apnea Events with accuracy of 88.5% [26].

In this study, a highly efficient Convolutional Neural Network (CNN) architecture has been proposed for sleep apnea detection. In this context, it is aimed to achieve high performance for the amount of data that is quite low for deep learning algorithms. Regarding this subject, it has been stated in the literature that a large and sufficient amount of data is required for the training of CNN architectures. In the proposed method, high quality features obtained with Googlenet and VGG-16 CNN models were fused with Canonical Correlation Analysis (CCA) method. The obtained fused features were given as inputs to a network mechanism that included the Gated Recurrent Unit (GRU) and the bias and weights of this designed network have been optimized using whale optimization algorithm (WOA). The contributions of the proposed method can be summarized as follows:

- Thanks to the proposed method, high classification performance is obtained for data sets containing fewer samples.
- WOA was chosen as the optimization technique for proposed method and on top of that performance was increased.
- Suitable for end-to-end training.
- It performs more effectively than current state-of-the-art methods.

This article is organized as follows: Section 2 describes non-contact radar systems and characteristics of radar signal dataset are introduced. Section 3 describes the methods within the scope of deep learning and the proposed method. Findings and discussion are included in Section 4. Section 5 contains conclusion.

MATERIALS AND METHODS

Radar Background

The radar technology includes Six-Port and on top of that it is portable. A bistatic antenna design has been developed to increase signal quality. The angle of inclination of the antenna beam for the receiver (Rx) and transmitter (Tx) antenna is $\pm 10^\circ$. In addition, there is a laser-based positioning system between these two antennas. Thanks to this system, the detection of the focal point becomes very easy. In Figure 1, designed Continuous Wave (CW) radar system for sleep apnea detection with a frequency of 24 GHz is visualized [27].

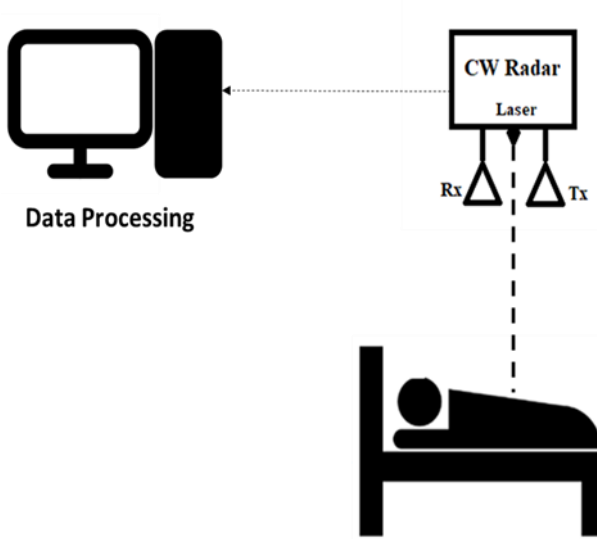


Figure 1: Description of Radar System.

One of the most important features of the Six-Port structure in the system is its high phase resolution. A movement in front of the receiver and transmitter antennas occur a phase change $\Delta\phi$. The amount of displacement Δx that the object has realized can be determined by using the wavelength λ in the transmitting antenna signal.

$$\Delta x == \frac{\Delta\lambda\phi}{4\pi} \quad (1)$$

Radar raw signals with two components named In-Phase (I) and Quadrature (Q) are obtained from the designed radar system. These signals are instantly digitized thanks to Texas Instruments ADS1298. $\Delta\phi$ is calculated by making

arctangent demodulation on I and Q signals. Digitized data with a sampling rate of 2000 Hz is stored using the XMC4500 microcontroller. At the same time, the synchronization process is provided by the microcontroller. As soon as 50 samples per channel are reached, it is transferred to the host computer via Ethernet.

Dataset Description

Radar measurement was carried out within the framework of a specific procedure. Volunteers were rested for at least 10 minutes prior to radar measurement. At this time, volunteers were asked to avoid sudden movements and to breathe normally [27]. There are four classes in the data set. These classes are Resting, Valsalva, Apnea and Tilt respectively. Radar measurements were carried out for at least 10 minutes. Resting situation arises at the beginning of the measurement scenario. Meanwhile, the heart rate of the subject is 54 BPM. After that, Valsalva takes place three times, with certain pauses. Valsalva maneuver is seen as a strong expiration for 20-21 s. In this case, hemodynamic changes are observed in the volunteers' circulatory system. In the case of apnea, volunteers use their breathing in two different ways as inhaled and ex-haled. These situations are realized by being defined for a long time. Tilt condition is carried out in two ways within the scope of measurement. These are Tilt up and Tilt down respectively. In the Tilt up process, the measurement table is tilted and the autonomic nervous system is triggered. In this case, there is a significant change in blood pressure and heart rate. Radar measurement process continues until the volunteers table reaches 70°. Then, Tilt down operation is performed. This process is applied until the measurement table is in a horizontal position. As in the tilt up process, autonomic nervous system activities take place. Figure 2 contains radar measurement class signals.

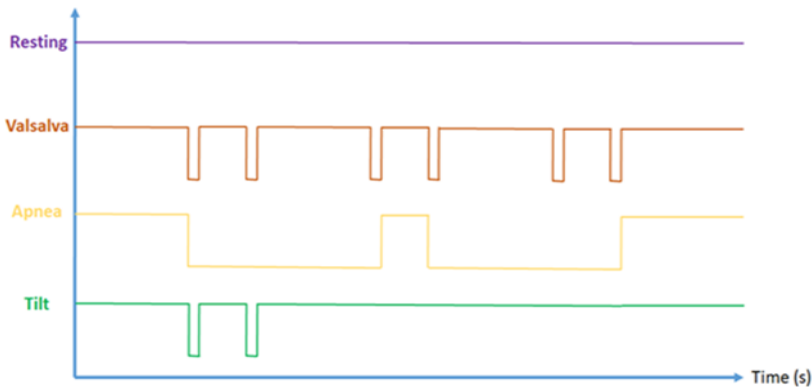


Figure 2: Radar Measurement Signals for Different Scenarios.

The Background in the Deep Learning Models

The concept of deep learning, which has recently emerged in the field of artificial intelligence, represents a deep network structure. Deep learning models aim to gain experience for solving problems. In order to obtain higher performances, it is necessary for getting more data and designing deeper networks. CNN architecture is widespread to suggest solution for more challenging problems [28]. CNN ideally analyses different sizes of matrix or vector forms. Within the scope of this section, 2D-CNN structures are mostly included. CNN architecture consists of different layers.

One of the main layers of CNN architecture is the convolution layer. There are convolution layers with different numbers and sizes of kernels in the architecture. These kernel parameters are updateable and ability to learn features related to the problem. 2-D convolution process is applied in the convolution layer. One of the most important advantages of this process is the sharing of parameters. In this way, the total number of parameters are reduced in the architecture. Pooling layer is one of the other important layers in CNN architecture. The use of the pooling layer following the convolution layer or rectified linear unit (ReLU) is seen. Sub-sampling is provided by the pooling layer. In this study, the maximum pooling layer was used. One of the other important layers is ReLU. It is often used with convolution layer. It is provided that CNN architecture gain a non-linear feature. ReLU layer sets the resulting negative values equal to zero. The fully connected layer in the CNN architecture acts as a classical neural network. The classification process is performed by processing the high-level features obtained with CNN architecture with the nodes in the fully connected layer. The following equation is used for calculations in CNN architecture, which basically consists of three main layers.

$$f(I_{next}) = pool_{n \times n}(\sigma(w * [D_{in}] + b)) \quad (2)$$

In Equation 2, I_{next} represents the input for the next layers. $pool_{n \times n}$ refers to the $n \times n$ size pooling process. Where σ , w and b represent ReLU process, convolution kernel and bias respectively. Softmax is found at the last of the network for classification. In Equation 3, a probabilistic calculation is performed for m classes.

$$P_j = \frac{e^{x_j}}{\sum_{m=1}^m e^{x_m}} \quad (3)$$

VGG-16 architecture consists of 138 million updateable parameters in total [29]. 13 of these layers are convolution layers and the remaining layers are Fully Connected Layers (FCL). All convolution layers have 3x3 size filters

with 1 stride. Pooling layers consist of 2x2 windows in 2 strides. Two FCL consist of 4,096 nodes while the last FCL consists of 1000 nodes. It has a total structure of 138 million trainable variables. ResNet-50 architecture, which is one of the pre-trained networks, consists of residual connections. Although it has a deeper structure than VGG-16 architecture, it contains less updatable parameters. It has a total of 50 layers. Finally, it has 23 million updateable parameters [30].

Spectrogram

Spectral content in the signal is only obtained by using Fourier Transform (FT). Spectrogram is also defined as Short Time Fourier Transform (STFT) that analyses the signal in time domain with small window size. It performs these operations using Equation 4.

$$X[t, f] = \sum_{-\infty}^{\infty} x[r] \cdot \omega[t - \tau] \cdot e^{-j\pi\tau} \quad (4)$$

While FT has only frequency function, STFT includes both frequency and time domain. The obtained result after STFT process is in matrix form. The resolution problem does not occur in the frequency domain for FT process. In the STFT process, the resolution in the frequency domain changes due to using finite size of windows and the piecemeal analysis of the signal [31]. Choosing a small window for STFT improves the resolution in the time domain. If the window is chosen large, the resolution decreases in time domain, although the resolution increases in the frequency domain.

Canonical Correlation Analysis for Feature Fusion

Canonical Correlation Analysis (CCA) obtains a single feature vector by fusing different feature vectors. Relationships between variables are analyzed within the scope of CCA [32]. If X and Y are defined as two different feature matrices; S_{xx} and S_{yy} represent the intra-cluster covariance matrix, and S_{xy} the inter-cluster covariance matrix. S contains information about feature pairs.

$$S = \begin{pmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{pmatrix} \quad (5)$$

To obtain a linear combination between CCA and features, the binary correlation objective functions must be maximized.

$$corr(W_x^T X, W_y^T Y) = \frac{cov(W_x^T X, W_y^T Y)}{var(W_x^T X) \cdot var(W_y^T Y)} \quad (6)$$

The fusion process is realized by adding the feature vectors after the transformation process is completed.

$$Z = \begin{pmatrix} W_x & 0 \\ 0 & W_y \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} \quad (7)$$

Where Z is defined as discriminant features of CCA.

Gated Recurrent Unit (GRU)

GRU, one of the recurrent neural network types, was developed in order to solve the gradient problems in concept of deep learning. GRU has a simpler structure compared to long-short-term memory (LSTM) [33]. There are two gates in the GRU such as reset gate and update gate. Where r and z donate reset and update gate respectively. Equations 8-11 are used to calculate transition functions for GRU.

$$z_t = \sigma(W^z x_t + V^z h_{t-1} + b^z) \quad (8)$$

$$r_t = \sigma(W^r x_t + V^r h_{t-1} + b^r) \quad (9)$$

$$h_t^1 = \tanh(W^c x_t + V^c (r_t o h_{t-1})) \quad (10)$$

$$h_t = (1 - z_t) o h_{t-1} + z_t o h_t^1 \quad (11)$$

Where W , V and b donate as shared parameters. o is element-wise product.

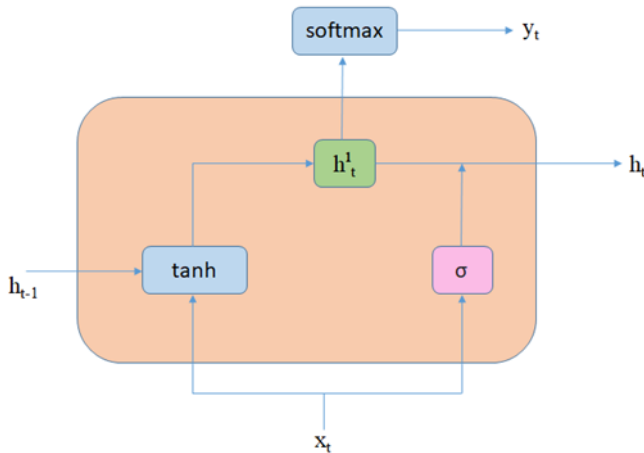


Figure 3: Block diagram of GRU.

The Proposed Method

GRU architecture is a Recurrent Neural Network (RNN) that enables the prediction of future information using available information. Besides the estimation of time series, RNN are also more efficient to solve classification problems. It is possible to increase network depth with GRU structures. In addition to GRUs being connected to each other in series, the connection can also be made in a residual way. $\tanh$, is an activation function in GRU, provides non-linearity characteristics. Training process could be carried out using several back propagation algorithms. The selection of optimization algorithms is very important in obtaining the most appropriate updateable parameters. The most common optimization algorithm is Stochastic Gradient Descent (SGD) [34]. It may take a lot of time to reach the global minimum. At the same time, problems such as gradient vanishing and explosion can be encountered. In order to overcome these problems, the weight and bias values of GRU structures within the proposed method were obtained by using the Whale Optimization Algorithm (WOA) [35]. There are three different updateable parameters in GRU structures such as Input weights, recurrent weights and biases. WOA, which is one of the best algorithms in meta-heuristic type, simulates the hunting strategy of humpback whales. In the WOA algorithm, the hunting search process is performed by mathematical modeling of the spiral bubble net feeding maneuver. During the hunting phase, the location of the prey is determined and surrounded by humpback whales. Once the best hunter is identified, other hunters start updating their location. This process ensures the best position. Equation 12-13 describe this process.

$$D = |C \cdot X^*(t) - X(t)| \quad (12)$$

$$X(t + 1) = X^*(t) - A \cdot D \quad (13)$$

X^* is the best position. The current position of X is updated in order to reach the best position. C and A symbolize the coefficient vectors and also are calculated according to Equation 14-15.

$$A = 2 \cdot a \cdot r - a \quad (14)$$

$$C = 2 \cdot r \quad (15)$$

There are two different phases to describe the behaviour of humpback whales in the WOA algorithm as shrinking encircling mechanism and spiral updating position. By decreasing the A value, the global minimum search space is narrowed. At this stage, the coefficient matrix A has assigned a random value between $-a$ and a . In the other phase, a spiral equation is defined a distance

between current whale and prey. By the way, the whales start updating their position in a spiral way. Equation 16 expresses the spiral motion of whales.

$$X(t + 1) = |X^*(t) - X(t)| \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \quad (16)$$

in which, b and l variables are used to define the spiral shape. Initially, random position is assigned to the humpback whales. A value is taken as greater than 1 or less than -1 to fulfil this condition. Mathematical expression of this situation is given in Equation 17-18.

$$D = |C \cdot X_{rand} - X| \quad (17)$$

$$X(t + 1) = X_{rand} - A \cdot D \quad (18)$$

STFT process has been applied in order to provide more information in time and frequency domain from raw radar signals. 2-D spectrogram matrices obtained by STFT were given as input to different pre-trained CNN networks. Three different pre-trained CNN networks were trained within the scope of comparative analysis. GoogleNet and VGG-16 models, which achieved the most successful results among these networks, were used within the framework of the proposed method. Thanks to these models, high level features for training and testing data have been obtained. These features were extracted from the fully connected layer of GoogleNet and VGG-16. They have been subjected to fusion process with CCA method by using Equation 5-7. The new fused features are given as input to the designed GRU block structure. There are 2 GRU layers and 2 dropout layers in the GRU block structure. The number of hidden neurons in GRU layers is 100 and 125, respectively. The dropout factor was chosen as 0.5. This structure was connected as end to end. The hyperparameters were decided with grid search. During the training phase, the parameters in the GRU block structure were updated with WOA algorithm instead of SGD. Mean Squared Error (MSE) was chosen as loss function and it was tried to be minimized. Proposed method is schematized in Figure 4. The main limitation of the proposed method is required to proceed two CNN structures owing to the fact that it has end-to-end structure.

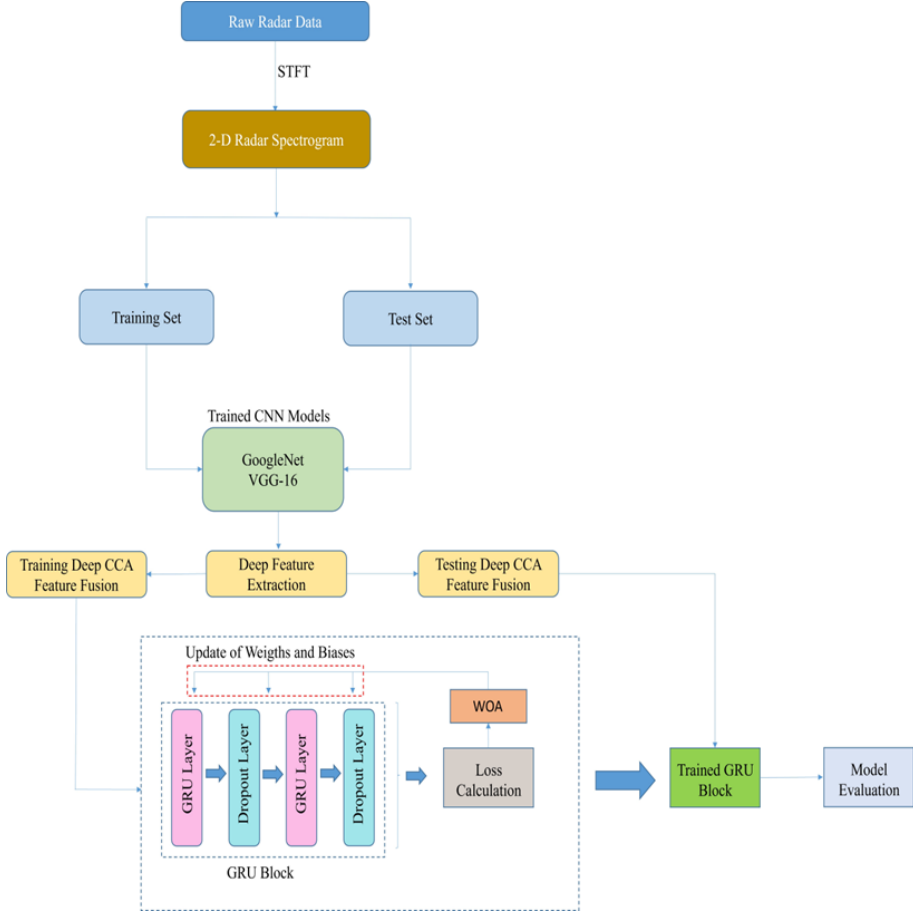


Figure 4: Diagram of the proposed method.

RESULTS AND DISCUSSION

A workstation with 2.8 GHz Intel Core i7-7700 HQ CPU, 16 GB RAM and NVIDIA GTX 1080 GPU was used for processing raw radar data, training pre-trained CNN networks and optimization of designed GRU block. Matlab 2020a program has been chosen to implement all process.

Six different classification metrics were used to evaluate the performance of pre-trained CNN networks within the scope of the comparative analysis. These metrics are Accuracy (ACC), Sensitivity (SEN), Specificity (SPE), F1-Score, Precision (PRE) and Matthews Correlation Coefficient (MCC), respectively. In the calculation of these metrics, True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) indices were used [36].

$$Accuracy = (TP + TN)/(TP + FN + TN + TP) \quad (19)$$

$$Sensitivity = TP/(TP + FN) \quad (20)$$

$$Specificity = TN/(TN + FP) \quad (21)$$

$$Precision = TP/(TP + FP) \quad (22)$$

$$F1 - Score = (2 \times TP)/(2 \times TP + FN + FP) \quad (21)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (22)$$

The performance of pre-trained CNN networks has been increased by using transfer learning in the training of deep learning models. GoogleNet, VGG-16 and ResNet-50 structures were used as pre-trained network. During the training phase, the number of mini-batches was 64. The maximum number of epochs was 20 for ResNet-50 and GoogleNet. VGG-16 has 50 maximum number of epoch owing to its linear structure. The initial learning rate was set at 0.1. The learning dropout factor was chosen as 0.1 in 10 epoch drop period. At the end of the training process, the test and validation process was carried out.

The hyper parameters for the proposed method are different from pre-trained CNN networks. Training process has been ended up to 500 epochs. The training chart of the proposed method is included in Figure 5. Blue and red curves in the graphs show the plot of accuracy and loss for training data, respectively. Black curves provide information about test data.

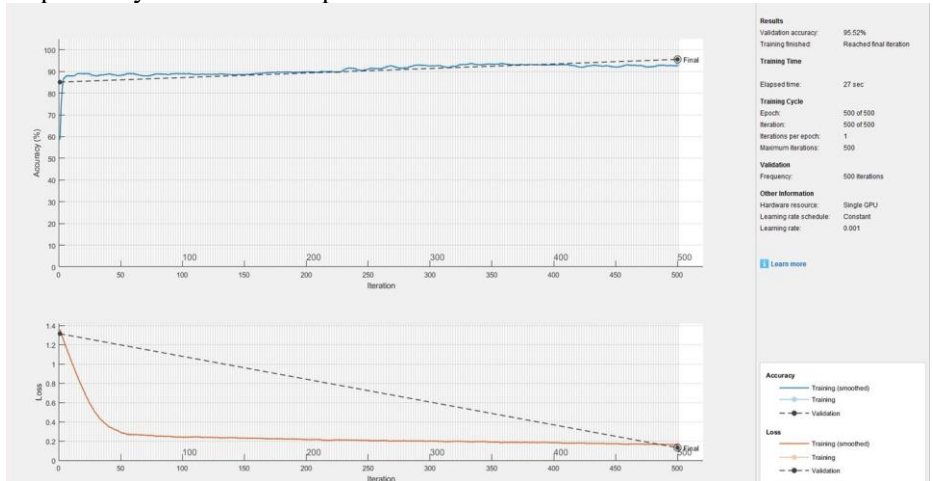


Figure 5: Diagram of the proposed method.

It varies according to the number of updateable parameters and training data within the training periods of the pre-trained CNN networks. The shortest training period belongs to GoogleNet with 3 min 18 sec. VGG-16 model has the longest training time with 9 min 55 sec. In the proposed method, it is necessary for additional 55 sec to train GRU block. Comparative performance metrics are included in Table 1.

Table 1: Comparison of Metric Performance for CNN Structures.

Methods	Evaluation Metrics (%)					
	ACC	SEN	SPE	PRE	F1-Score	MCC
VGG-16	73.13	71.52	89.93	80.54	74.34	65.62
Resnet-50	68.66	69.23	88.45	73.41	69.64	59.42
GoogleNet	76.12	82.71	92.09	79.76	79.07	72.96
Proposed Method	95.52	94.74	98.29	96.35	95.40	93.96

When the methods were evaluated in terms of performance, ResNet-50 model showed the lowest performance. The classification metrics obtained with this model are 68.66% ACC, 69.23% SEN, 88.45% SPE, 73.41% PRE, 69.64% F1-Score and 59.42% MCC. In these pre-trained networks, the highest performance was achieved with the training of the GoogleNet model. Its performance metrics were 76.12% ACC, 82.71% SEN, 92.09% SPE, 79.76% PRE, 79.07 F1-Score and 72.96% MCC. The proposed method in terms of performance has made quite a difference to the pre-trained CNN networks. The performance metrics of the proposed method are 95.52% ACC, 94.74% SEN, 98.29% SPE, 96.35% PRE, 95.40% F1-Score and 93.96% MCC. Confusion matrices for test data of pre-trained CNN networks are included in Figure 6.

True Class	Apnea				Accuracy	Error	
	12				100%		
	Resting		13	1	1	86.7%	13.3%
	Tilt			27		100%	
	Valsalva			1	12	92.3%	7.7%
Accuracy	100%	100%	93.1%	92.3%			
Error			6.9%	7.7%			
		Apnea	Resting	Tilt	Valsalva		
		Predicted Class					

Figure 6: Confusion Matrix for Test Data of Proposed Method.

When the findings in the confusion matrix were examined, 100% performance was achieved in Apnea and Tilt class accuracy. For the Resting class, an accuracy performance of 86.7% was obtained and it has a 13.3% error rate. Valsalva class accuracy is 92.3%. The error rate is 7.7%. Performance comparison between proposed method and state-of-arts is given in Table 2. The lowest performance in the literature belongs to Savage et al. [19]. Its performance is 85.8% accuracy, 70.0% sensitivity and 79.0% specificity. The highest performance is Tran et al. [22] with an accuracy of 93.8%. The performance of the proposed method is higher than the studies in literature. Our proposed method achieved an increase of 1.72% in accuracy, 1.97% in sensitivity and 3.13% in specificity.

Table 2. Performance Comparison between Proposed Method and State-of-Arts.

Methods	Number of Classes	Evaluation Metrics (%)			
		ACC	SEN	SPE	PRE
Zhang [16] (2021)	2	90.82	-	-	-
Slapnicar et al. [18]	5	88	-	-	-
Savage et al.	4	85.8	70	89	-

[19] (2016)					-
Tran et al. [20] (2015)	-	91.29	-	-	-
Zaffaroni et al. [21] (2009)	4	-	89	94	
Lai et al. [22] (2023)	4	93.8	-	-	
Han et al. [23] (2022)	3	90.1	-	-	91.0
Chen et al. [24] (2022)	2	93.0	92.77	94.12	98.72
Baboli et al. [25] (2020)	2	92.0	86.0	91.0	-
Kwon et al. [26] (2021)	4	93.0	78.1	95.6	-
Proposed Method	4	95.52	94.74	98.29	96.35

CONCLUSIONS

In summary, high performance was achieved in the classification of vital signals based on pre-trained CNN networks in the concept of proposed method. Spectrogram process was performed on the raw radar data of the public database. In this way, time and frequency domain of the radar data was investigated. The obtained 2D dimensional radar spectrograms were used for training and testing of pre-trained CNN networks. The obtained high level features from VGG-16 and GoogleNet networks were fused with CCA analysis process. These fused features were used as inputs to GRU block. The parameter optimization of the GRU structure was performed with WOA algorithm and its effects on performance were investigated. In future studies, attention network integration can be provided in addition to deep learning architectures. In addition, it is anticipated that the performance will be

increased by combining the data obtained from different measurement devices.

REFERENCES

- [1] Young, T.; Palta, M.; Dempsey, J.; Skatrud, J.; Webber, S.; Badr, S. The Occurrence of Sleep-Disordered Breathing Among Middle-Aged Adults. *N. Engl. J. Med.* 1993, 328, 1230–1235.
- [2] American Academy of Sleep Medicine. International Classification of Sleep Disorders, Revised: Diagnostic and Coding Manual; American Academy of Sleep Medicine: Chicago, IL, USA, 2001.
- [3] Leung, R.S.T.; Bradley, T.D. Sleep Apnea and Cardiovascular Disease. *Am. J. Respir. Crit. Care Med.* 2001, 164, 2147–2165.
- [4] Seicean, S.; Strohl, K.P.; Seicean, A.; Gibby, C.; Marwick, T.H. Sleep Disordered Breathing as a Risk of Cardiac Events in Subjects With Diabetes Mellitus and Normal Exercise Echocardiographic Findings. *Am. J. Cardiol.* 2013, 111, 1214–1220.
- [5] Nieto, F.J.; Peppard, P.E.; Young, T.; Finn, L.; Hla, K.M.; Farré, R. Sleep-disordered Breathing and Cancer Mortality: Results from the Wisconsin Sleep Cohort Study. *Am. J. Respir. Crit. Care Med.* 2012, 186, 190–194.
- [6] Lie, D.Y.C.; Ichapurapu, R.; Jain, S.; Lopez, J.; Banister, R.E.; Nguyen, T.; Griswold, J. A 2.4 GHz Non-Contact Biosensor System for Continuous Monitoring of Vital-Signs. In *Telemedicine Techniques and Applications*; Graschew, G., Ed.; InTech: Rijeka, Croatia, 2011; pp. 211–238.
- [7] Kagawa, M.; Ueki, K.; Tojima, H.; Matsui, T. Noncontact Screening System with Two Microwave Radars for the Diagnosis of Sleep Apnea-Hypopnea Syndrome. In *Proceedings of the 35th Annual International Conference of the IEEE EMBS*, Osaka, Japan, 3–7 July 2013; pp. 2052–2055.
- [8] Kagawa, M.; Yoshida, Y.; Kubota, M.; Kurita, A.; Matsui, T. An overnight vital signs monitoring system for elderly people using dual microwave radars. In *Proceedings of the Asia-Pacific Microwave Conference 2011*, Melbourne, Australia, 5–8 December 2011; pp. 590–593.
- [9] X. Wang, C. Yang and S. Mao, "PhaseBeat: Exploiting CSI Phase Data for Vital Sign Monitoring with Commodity WiFi Devices," 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 2017, pp. 1230-1239, doi: 10.1109/ICDCS.2017.206.
- [10] F. Quaiyum, N. Tran, T. Phan, P. Theilmann, A. E. Fathy and O. Kilic, "Electromagnetic Modeling of Vital Sign Detection and Human Motion Sensing Validated by Noncontact Radar Measurements," in *IEEE Journal of Electromagnetics, RF and Microwave in Medicine and Biology*, vol. 2, no. 1, pp. 40-47, March 2018, doi: 10.1109/JERM.2018.2807978.
- [11] J. J. Saluja, J. Lin and J. Casanova, "A Supervised Learning Approach for Real Time Vital Sign Radar Harmonics Cancellation," 2018 IEEE International Microwave Biomedical Conference (IMBioC), Philadelphia, PA, USA, 2018, pp. 67-69, doi: 10.1109/IMBIOC.2018.8428911.
- [12] S. Erdoğan, Ş. Yılmaz and A. Öncü, "Microwave Noncontact Vital Sign Measurements for Medical Applications," 2019 IEEE International

- Symposium on Medical Measurements and Applications (MeMeA), Istanbul, Turkey, 2019, pp. 1-5, doi: 10.1109/MeMeA.2019.8802165.
- [13] J. Saluja, J. Casanova and J. Lin, "A Supervised Machine Learning Algorithm for Heart-Rate Detection Using Doppler Motion-Sensing Radar," in *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, vol. 4, no. 1, pp. 45-51, March 2020, doi: 10.1109/JERM.2019.2923673.
 - [14] H. Chang, C. Lin, Y. Lin, W. Chung and T. Lee, "DL-Aided NOMP: a Deep Learning-Based Vital Sign Estimating Scheme Using FMCW Radar," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-7, doi: 10.1109/VTC2020-Spring48590.2020.9128552.
 - [15] W. -C. Lai, "Design of Receiver Frontend with Deep Neural Network for Doppler Radar Heart Rate Detection," 2020 IEEE 5th International Conference on Integrated Circuits and Microsystems (ICICM), Nanjing, China, 2020, pp. 121-124, doi: 10.1109/ICICM50929.2020.9292303.
 - [16] G. Baldoumas, D. Peschos, G. Tatsis, V. Christofilakis, S. K. Chronopoulos, P. Kostarakis, P. A. Varotsos, N. V. Sarlis, E. S. Skordas, A. Bechlioulis, L. K. Michalis, K. K. Naka. (2021) Remote sensing natural time analysis of heartbeat data by means of a portable photoplethysmography device. *International Journal of Remote Sensing* 42:6, pages 2292-2302.
 - [17] E. Cardillo, C. Li and A. Caddemi, "Vital Sign Detection and Radar Self-Motion Cancellation Through Clutter Identification," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 69, no. 3, pp. 1932-1942, March 2021, doi: 10.1109/TMTT.2021.3049514.
 - [18] Slapničar G, Wang W, Luštrek M. Classification of Hemodynamics Scenarios from a Public Radar Dataset Using a Deep Learning Approach. *Sensors*. 2021; 21(5):1836. <https://doi.org/10.3390/s21051836>.
 - [19] Savage, H. O., Khushaba, R. N., Zaffaroni, A., Colefax, M., Farrugia, S., Schindhelm, K., Teschler, H., Weinreich, G., Grueger, H., Neddermann, M., Heneghan, C., Simonds, A., and Cowie, M. R. (2016) Development and validation of a novel non-contact monitor of nocturnal respiration for identifying sleep-disordered breathing in patients with heart failure. *ESC Heart Failure*, 3: 212– 219. doi: 10.1002/ehf2.12086.
 - [20] Tran, V. P., & Al-Jumaily, A. A. (2015). Non-contact dual pulse Doppler system based real-time relative demodulation and respiratory & heart rates estimations for chronic heart failure patients. *Procedia Computer Science*, 76, 47-52.
 - [21] Zaffaroni, A., De Chazal, P., Heneghan, C., Boyle, P., Mppm, P. R., & McNicholas, W. T. (2009, September). SleepMinder: an innovative contact-free device for the estimation of the apnoea-hypopnoea index. In 2009 annual international conference of the IEEE engineering in medicine and biology society (pp. 7091-9094). IEEE.
 - [22] Lai, D. K. H., Zha, L. W., Leung, T. Y. N., Tam, A. Y. C., So, B. P. H., Lim, H. J., ... & Cheung, J. C. W. Dual ultra-wideband (UWB) radar-based sleep posture recognition system: Towards ubiquitous sleep monitoring. *Engineered Regeneration*, (2023), 4(1), 36-43.

- [23] Han, Y., Yarovoy, A., & Fioranelli, F. (2022, April). An approach for sleep apnea detection based on radar spectrogram envelopes. In 2021 18th European Radar Conference (EuRAD) (pp. 17-20). IEEE.
- [24] Chen, F. K., Wang, Y. K., Lin, H. P., Chen, C. Y., Yeh, S. M., & Wang, C. Y. (2022, November). Feasibility study for apnea screening in patients' homes using radar and machine learning method. In 2022 IEEE 22nd International Conference on Bio-informatics and Bioengineering (BIBE) (pp. 282-287). IEEE.
- [25] Baboli, M., Singh, A., Soll, B., Boric-Lubecke, O., & Lubecke, V. M. (2019). Wireless sleep apnea detection using continuous wave quadrature Doppler radar. *IEEE Sensors Journal*, 20(1), 538-545.
- [26] Kwon, H. B., Son, D., Lee, D., Yoon, H., Lee, M. H., Lee, Y. J., ... & Park, K. S. (2021). Hybrid CNN-LSTM network for real-time apnea-hypopnea event detection based on IR-UWB radar. *IEEE Access*, 10, 17556-17564.
- [27] Schellenberger S, Shi K, Steigleder T, et al. A dataset of clinically recorded radar vital signs with synchronised reference sensor signals. *Sci Data* 7, 291 (2020). doi: 10.1038/s41597-020-00629-5
- [28] Öztürk Ş. Stacked auto-encoder based tagging with deep features for content-based medical image retrieval. *Expert Systems with Applications*. 2020; 161(5): 113693. doi: 10.1016/j.eswa.2020.113693
- [29] Zhang X, Zou J, He K, Su J. Accelerating very deep convolutional networks for classification and detection. *IEEE transactions on pattern analysis and machine intelligence*. 2015; 38(10), 1943-1955. doi: 10.1109/TPAMI.2015.2502579
- [30] Fulton LV, Dolezel D, Harrop J, Yan Y, Fulton CP. Classification of Alzheimer's disease with and without imagery using gradient boosted machines and ResNet-50. *Brain sciences*. 2019; 9(9), 212. doi: 10.3390/brainsci9090212
- [31] Giv HH. Directional short-time Fourier transform. *Journal of Mathematical Analysis and Applications*. 2013; 399(1): 100-107. doi: 10.1016/j.jmaa.2012.09.053
- [32] Haghighat M, Abdel-Mottaleb M, Alhalabi W. Fully automatic face normalization and single sample face recognition in unconstrained environments. *Expert Systems with Applications*. 2016; 47, 23-34. doi: 10.1016/j.eswa.2015.10.047
- [33] Zhao R, Wang D, Yan R, Mao K, Shen F, Wang J. Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks. *IEEE Transactions on Industrial Electronics*. 2018; 65(2): 1539-1548. doi: 10.1109/TIE.2017.2733438
- [34] Wu Z, Ling Q, Chen T, Giannakis GB. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*. 2020; 68: 4583-4596. doi: 10.1109/TSP.2020.3012952
- [35] Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in engineering software*. 2016; 95, 51-67. doi: 10.1016/j.advengsoft.2016.01.008
- [36] Khadem, H., Nemat, H., Elliott, J., Benaissa, M. Blood Glucose Level Time Series Forecasting: Nested Deep Ensemble Learning Lag Fusion. *Bioengineering*, 2023; 10(4), 487. doi: 10.3390/bioengineering10040487

