MDPI

Article

cStick 2.0: An IoMT-Edge Based Vision-Enabled Smart System for Personalized Fall Prediction and Detection

Laavanya Rachakonda [1], Sai Sri Harsha Chakravarthula [2], Saraju P. Mohanty [3] and Elias Kougianos [4]

[1] Dept. of Computer Science; rachakondal@uncw.edu
[2] Dept. of Computer Science and Engineering; SaiSriHarshaChakravarthula@my.unt.edu
[3] Dept. of Computer Science and Engineering; smohanty@ieee.org
[4] Dept. of Computer Science and Engineering; elias.kougianos@unt.edu
* Correspondence: rachakondal@uncw.edu (L.R.), SaiSriHarshaChakravarthula@my.unt.edu (S.S.H.C.)

Abstract

Falls among older adults can result in severe injuries, reduced independence, and substantial healthcare burden. cStick 2.0 presents a vision-enabled, IoMT-edge based smart walking-stick architecture that combines multimodal fall-risk classification, embedded obstacle awareness, physiological and motion-related sensing, stick-pressure interaction, GPS acquisition, display output, and immediate buzzer feedback. The fall-risk classifiers were evaluated externally using a 9,670-record development dataset derived from 2,039 source records and 7,631 cleaned, transformed, and augmented records. The compact cStick Deep Neural Network achieved 95.40% accuracy, 92.35% balanced accuracy, a macro F1-score of 93.90%, and a ROC-AUC of 97.44%. Feature-ablation analysis further showed that the complete six-input configuration provided stronger and more balanced performance than individual or reduced feature groups. The Arduino Nicla Vision obstacle-awareness module was implemented using an INT8 Edge Impulse model with centroid-based direction assignment and time-of-flight distance measurement. A controlled evaluation involving 108 obstacle-present trials and 36 clear-path trials produced an obstacle-presence accuracy of 75.00%, with embedded operation at approximately 19–20 FPS. cStick 2.0 also implements CSV-based sensor-record accumulation, providing a pathway for proposed external retraining and future user-specific personalization. The current results establish prototype-level technical feasibility, while synchronized participant-level acquisition, older-adult evaluation, application connectivity, multimodal accessibility feedback, secure caregiver communication, and longitudinal personalization constitute the next validation stage.

Keywords: Obstacle Detection; Fall Prediction; Smart Assistive Device; Arduino Nicla Vision; Elderly Safety; Artificial Neural Networks

1. Introduction

Falls are among the most common and consequential adverse events experienced by older adults. Approximately one-third of adults aged over 65 years experience at least one fall annually, and the resulting injuries contribute substantially to disability, reduced mobility, loss of independence, hospitalization, and mortality [1,2]. The risk is particularly important because a previous fall increases the likelihood of subsequent incidents, while many older adults may not voluntarily report falls unless specifically asked by healthcare professionals [2,3]. These characteristics make early risk recognition and timely intervention important components of older-adult safety.

Received:
Accepted:
Published:

 https://doi.org/10.3390/electronics1010000

Falls commonly occur at ground level during routine activities such as walking, standing, turning, sitting, or navigating indoor and outdoor environments. Slippery surfaces, poor lighting, uneven flooring, clutter, nearby obstacles, inappropriate footwear, and limited awareness of the surrounding environment may increase the likelihood of losing balance [4]. Sensory impairments can further complicate safe navigation. Hearing loss, for example, has been associated with an increased likelihood of falling among older adults, potentially because auditory information contributes to environmental awareness and postural control [5].

Fall risk is also multidimensional and cannot be represented adequately by a single motion threshold. Gait impairment, muscle weakness, cognitive changes, cardiovascular conditions, neurological disorders, medication use, previous falls, and reduced balance can interact with environmental hazards and daily activity patterns [2,6]. Physiological information may therefore provide useful contextual evidence when interpreted together with motion-related and environmental observations. However, the presence of a physiological variable does not independently establish a causal relationship with a fall event; its contribution must be evaluated within the available dataset and acquisition conditions.

The consequences of falling extend beyond immediate physical injury. Fractures, traumatic brain injuries, and prolonged hospitalization can reduce functional independence, while fear of another fall may cause older adults to restrict daily activity, avoid social interaction, and reduce physical movement. Such activity restriction can further affect strength, balance, confidence, and quality of life [1,7]. Families and caregivers may consequently experience continued concern when older adults live independently or remain without direct supervision for extended periods.

These challenges indicate that older-adult safety requires more than a post-event alarm. A practical assistive platform should observe both the user and the surrounding environment, distinguish normal movement from potentially unsafe conditions, identify obstacles in the walking path, and provide an immediate response when risk is detected. Integrating complementary sensing modalities can support this objective by representing motion-related behavior, user–stick interaction, physiological context, and environmental proximity within a common decision framework.

Figure 1. Conceptual Illustration of an Older Adult Navigating an Obstacle-Prone Indoor Environment Using cStick 2.0.

Figure 1 illustrates the design motivation of cStick 2.0. During everyday movement, an older adult may encounter both user-related changes and external hazards. A change in motion, increased dependence on the walking stick, altered physiological context, or an unnoticed obstacle may not independently indicate a critical event. Their combined occurrence, however, can provide additional information for identifying a potentially unsafe condition. The figure represents the intended assistive scenario and does not represent a completed older-adult participant evaluation.

Internet of Things (IoT) technologies provide a foundation for integrating sensors, embedded processors, communication interfaces, and actuators, while the Internet of Medical Things (IoMT) extends this connectivity toward health-monitoring and assistive-care applications [8–10]. Edge-oriented processing further supports time-sensitive applications by placing selected sensing and response functions near the user, thereby reducing dependence on continuous remote processing and limiting unnecessary transmission of sensitive information [11].

Motivated by these requirements, cStick 2.0 presents an IoMT-edge based, vision-enabled smart walking-stick architecture for multimodal fall-risk classification and obstacle awareness. The hardware architecture integrates an Arduino UNO WiFi Rev2 for sensor acquisition and control, an Arduino Nicla Vision for embedded visual processing, motion and physiological sensors, a grip-pressure sensor, proximity sensing, GPS support, and a buzzer. The Arduino UNO WiFi Rev2 provides onboard Wi-Fi and Bluetooth Low Energy capability through the NINA-W102 module, together with an ATECC608 secure element, thereby supporting future secure IoMT communication without requiring a separate wireless module [12]. The Arduino Nicla Vision provides an embedded camera and local processing resources suitable for edge-oriented vision tasks [13].

The current prototype implements local sensor acquisition, GPS data acquisition, buzzer activation, CSV-based record storage, and embedded obstacle processing. The compact Deep Neural Network (DNN) and the comparison models are trained and evaluated externally rather than being executed directly on the Arduino UNO WiFi Rev2. Similarly, the companion application represents a proposed interface prototype and is not presently connected to the walking-stick hardware. Haptic feedback, spoken-audio notification, automatic caregiver communication, and full application synchronization are retained as structured extensions of the architecture rather than being reported as experimentally validated functions.

The fall-risk model is evaluated using a transformed, cleaned, expanded, and augmented version of the publicly available dataset associated with the earlier cStick framework [14,15]. The resulting development dataset contains 9,670 tabular records with distance, grip pressure, heart rate variability (HRV), externally supplied blood glucose, SpO_2, accelerometer-related information, and an assigned decision class. Blood glucose is supplied through a dataset or application interface rather than an onboard glucose sensor and is therefore treated as contextual health information rather than a continuously measured real-time signal. The available records were not acquired as synchronized multimodal streams from older adults using the complete cStick 2.0 prototype. Consequently, the model results represent prototype-level offline classification performance rather than clinical or synchronized participant-level validation.

The embedded obstacle-awareness module was evaluated separately using 144 controlled trials. The evaluation included a person, chair, and cardboard box positioned at 50, 100, and 150 cm under normal and dim indoor lighting. Stationary-camera and moving-camera conditions were included, together with 36 clear-path trials. The Arduino Nicla Vision obstacle module achieved an obstacle-presence accuracy of 75% under these controlled conditions. This result strengthens the hardware validation beyond frame-rate reporting while retaining a clear distinction between component-level testing and future end-to-end older-adult assessment.

Earlier cStick and Good-Eye systems established the feasibility of combining sensing, physiological information, and fall-related decision support [14,16]. cStick 2.0 extends these foundations through embedded Nicla Vision obstacle awareness, a wireless-capable layered hardware architecture, expanded machine-learning and TinyML benchmarking, detailed class-oriented evaluation, and an implemented CSV data-accumulation pathway that can support subsequent external model refinement. The CSV update mechanism permits newly acquired records to be appended to the existing repository, while user-specific retraining and its quantitative benefit remain part of the next personalization stage.

The major contributions of cStick 2.0 are summarized as follows:

- **Integrated IoMT-edge assistive architecture:** cStick 2.0 combines motion sensing, HRV and SpO_2 monitoring, grip-pressure interaction, obstacle proximity, GPS acquisition, embedded vision, local buzzer feedback, and CSV-based sensor-record accumulation within a layered walking-stick architecture. The design clearly separates implemented embedded functions, external model development, and proposed wireless, application, personalization, and caregiver-support extensions.
- **Embedded vision-based obstacle awareness with controlled validation:** The Arduino Nicla Vision module performs obstacle-presence and direction estimation on embedded hardware. A controlled 144-trial evaluation was conducted across three obstacle types, three reference distances, two lighting conditions, stationary and moving camera operation, and clear-path conditions. The resulting 75% obstacle-presence accuracy provides quantitative component-level evidence for the proposed vision module.
- **Expanded multimodal fall-risk classification and deployment-oriented comparison:** The compact cStick DNN is compared with Logistic Regression, SVM, Random Forest, XGBoost, TinyML Dense, TinyML INT8, wearable accelerometer baselines, and a majority baseline using a common evaluation framework. The comparison includes accuracy, balanced accuracy, macro F1-score, ROC-AUC, model size, and inference latency. The cStick DNN achieves 95.40% accuracy, 92.35% balanced accuracy, 93.90% macro F1-score, and 97.44% ROC-AUC on the transformed development dataset. Feature-group analysis further examines the contribution of the heterogeneous input variables rather than relying only on intuitive associations.

The current evidence therefore combines a quantitatively evaluated offline fall-risk classifier, an embedded obstacle-awareness module evaluated under controlled conditions, and a functionally demonstrated sensing and response prototype. The next development stage will extend cStick 2.0 toward synchronized timestamped acquisition, participant- and session-separated validation, quantitative personalization assessment, complete device-to-application communication, caregiver usability studies, and older-adult evaluation conducted with institutional approval and informed consent.

2. Related Work

Research related to cStick 2.0 spans wearable fall monitoring, physiological fall-risk assessment, machine-learning classification, adaptive personalization, obstacle-aware mobility assistance, IoMT architectures, and commercial emergency-support devices. These directions provide important foundations, but they frequently address fall detection, fall-risk assessment, obstacle navigation, or emergency communication as separate functions. The following discussion examines these areas and identifies the design space addressed by cStick 2.0.

2.1. Wearable, Physiological, and Learning-Based Fall Monitoring

Wearable fall-detection systems commonly use accelerometers, gyroscopes, and barometric sensors to recognize abrupt motion, changes in body orientation, impact, and posture transitions. Wu et al. developed a wearable system using accelerometer and barometric measurements to distinguish fall-related motion and changes in height [17]. Such approaches demonstrate the effectiveness of lightweight wearable sensing, although many are designed primarily to recognize an event after a fall-related movement has occurred.

Broader reviews of wearable fall-risk assessment show that inertial sensors, plantar-pressure sensors, electromyography, and other wearable modalities can characterize gait, balance, and movement patterns associated with fall risk [18]. Machine-learning studies have extended these systems beyond fixed thresholds by applying decision trees, SVMs, neural networks, and ensemble methods to wearable and non-wearable data. However, reported performance is strongly influenced by dataset composition, sensor placement, activity selection, validation design, and the difference between simulated falls and naturally occurring events [19,20].

Pressure-based systems provide another perspective on movement and balance. Instrumented shoes and insoles use force-sensitive or plantar-pressure sensors to estimate gait phase, stance behavior, and pressure distribution across the foot [21]. These systems offer detailed biomechanical information but require additional wearable placement and generally do not provide direct awareness of obstacles or environmental hazards. cStick 2.0 instead uses grip pressure to represent user interaction with the walking stick. This value is distinct from clinical blood pressure and from plantar-pressure measurements.

Physiological and profile-based fall-risk prediction has also been investigated. Razmara et al. applied an artificial neural network to physiological and functional profiles collected from older adults, demonstrating that multidimensional profiles can support fall-risk classification [6]. Such findings motivate the consideration of complementary physiological context, although the clinical meaning of each variable depends on how, when, and from whom it was acquired. For cStick 2.0, HRV and SpO_2 are treated as physiological inputs, while manually supplied blood glucose is treated as an external contextual feature rather than an onboard continuous measurement.

Personalization is relevant because walking patterns, physiological baselines, and sensor responses vary among users. Nho et al. proposed a user-adaptive fall detector using a fusion of accelerometer and heart-rate information, while Ngu et al. investigated personalized fall detection using user-specific movement data [22,23]. These approaches demonstrate the potential of adapting a classifier to individual behavior. In cStick 2.0, newly acquired records can be appended to a CSV repository, creating an implemented data-accumulation pathway for subsequent external retraining. Quantitative before-and-after evidence of personalized improvement remains a separate validation stage.

The earlier Good-Eye framework combined computer vision with physiological sensing for fall prediction and detection, while the original cStick framework introduced a calm walking-stick concept for fall prediction, detection, and control in an IoMT setting [14,16]. cStick 2.0 advances this progression by incorporating Arduino Nicla Vision-based obstacle awareness, a controlled obstacle-evaluation protocol, stronger machine-learning and TinyML baselines, class-oriented performance metrics, feature-group analysis, and a layered separation between embedded operation and external model development.

2.2. Obstacle-Aware Assistive Mobility and IoMT Architectures

Obstacle-aware walking aids have been developed using ultrasonic sensors, infrared sensors, cameras, LiDAR, GPS, and combinations of embedded processing and wireless communication. Ikram et al. presented an IoT-based obstacle-detection and warning system for visually impaired users, demonstrating the role of connected sensors in navigation assistance [24]. Farooq et al. developed an IoT-enabled intelligent stick combining ultrasonic sensing, visual object recognition, GPS, and user-feedback mechanisms [25]. These systems emphasize environmental awareness and navigation, but they are not primarily designed to interpret multimodal physiological and movement-related fall-risk classes.

More advanced smart-cane systems combine vision with depth or laser sensing. Mai et al. integrated 2D LiDAR and an RGB-D camera to support navigation, localization, and obstacle recognition [26]. Such platforms provide rich environmental information but require greater sensing and computational resources. The Arduino Nicla Vision module in cStick 2.0 follows a more compact embedded approach intended for lightweight obstacle-presence and direction estimation within the walking-stick architecture.

IoMT architectures connect sensing, processing, communication, and healthcare-facing services. Islam et al. described the enabling architectures, communication technologies, applications, and security challenges of IoT-based healthcare systems [9]. Edge-computing architectures complement IoMT by processing selected data near the sensing device, improving responsiveness and reducing continuous dependence on remote infrastructure [11]. The Arduino UNO WiFi Rev2 provides cStick 2.0 with onboard Wi-Fi and BLE capability, while the Nicla Vision performs local visual processing. Complete wireless synchronization, authenticated caregiver access, and secure device-to-application communication are positioned as future extensions rather than as evaluated communication results.

Table 1 summarizes representative peer-reviewed approaches and their relationship to the design scope of cStick 2.0.

2.3. Commercial Fall-Related Devices and Assistive Applications

Commercial smartwatches, medical-alert systems, and mobile safety services provide widely accessible fall-related functions. Apple Watch, Samsung Galaxy Watch, Google Pixel Watch, and selected Garmin devices use wrist-worn motion sensing to support hard-fall or incident detection, emergency calling, and location sharing [27–30]. These systems provide important post-event assistance but are centered on wrist-worn operation and do not directly include cane-mounted grip sensing or embedded forward-facing obstacle awareness.

Dedicated medical-alert products from ADT Health, Bay Alarm Medical, Lively, MobileHelp, Lifeline, Medical Alert, and UnaliWear provide combinations of emergency buttons, optional fall detection, two-way communication, GPS support, and caregiver or monitoring-center contact [31–37]. Their main strength is rapid access to assistance after an emergency. Their design objectives generally differ from proactive multimodal fall-risk classification and obstacle-aware walking support.

Commercial and research-oriented smart sticks focus primarily on navigation and hazard detection. Ultrasonic, infrared, camera, and LiDAR-based systems can identify obstacles and provide sound or vibration warnings [24–26]. cStick 2.0 occupies a complementary design space by integrating walking-stick interaction, physiological context, motion-related information, controlled embedded obstacle testing, and three-state fall-risk classification within one architecture. The current implementation should nevertheless be distinguished from its proposed extensions: buzzer response and local sensing are demonstrated, while haptic feedback, spoken audio, application synchronization, and automatic caregiver notification remain future modules.

Table 2 summarizes representative commercial device categories and their relationship to the design scope of cStick 2.0.

The comparison shows that existing wearable and commercial systems provide valuable fall detection, location sharing, obstacle warning, and emergency-response capabilities.

Table 1. Summary of Peer-Reviewed Fall Monitoring, Personalization, Obstacle-Awareness, and IoMT Approaches.

Study/Approach	Main Technology	Primary Focus	Relationship to cStick 2.0
Good-Eye and cStick [14,16]	Vision, physiological sensing, walking-stick sensing, and IoMT concepts	Fall prediction, detection, and assistive monitoring	Provide the foundation extended by cStick 2.0 through embedded obstacle validation, expanded baselines, feature analysis, and a layered hardware architecture.
Wearable fall detection [17]	Accelerometer and barometric sensing	Recognition of fall-related motion and posture transitions	Demonstrates lightweight event detection but does not integrate grip interaction, physiological context, and obstacle awareness within a walking stick.
Wearable fall-risk assessment [18]	Inertial, pressure, physiological, and wearable sensors	Assessment of gait, balance, and fall-related risk factors	Establishes the value of multimodal sensing while highlighting variations in sensor placement, datasets, and validation methods.
Machine-learning fall systems [19,20]	Wearable and non-wearable sensing with ML classifiers	Fall detection, prevention, and classification	Motivates stronger model comparison and class-oriented evaluation rather than reliance on a single accuracy value.
Instrumented gait systems [21]	Plantar-pressure sensors and instrumented footwear	Gait-phase and pressure-distribution analysis	Provides detailed gait information but requires a wearable shoe or insole and does not directly provide cane-based obstacle awareness.
Physiological-profile prediction [6]	Physiological and functional profiles with ANN	Older-adult fall-risk prediction	Supports multidimensional risk modeling while differing from the sensor-oriented and environmental architecture of cStick 2.0.
User-adaptive fall detection [22,23]	User-specific movement, accelerometer, and heart-rate information	Personalized fall detection	Motivates the CSV accumulation and future external retraining pathway in cStick 2.0; personalized improvement is not yet quantitatively claimed.
IoT-enabled obstacle sticks [24,25]	Ultrasonic, camera, GPS, IoT communication, and user alerts	Obstacle recognition and assistive navigation	Provides strong environmental assistance but does not focus on three-state multimodal fall-risk classification.
LiDAR and RGB-D smart cane [26]	2D LiDAR, RGB-D camera, IMU, GPS, and embedded processing	Navigation, localization, and object recognition	Provides richer depth and navigation functions with higher sensing complexity than the compact Nicla Vision-based module.
IoMT and edge healthcare [9,11]	Connected healthcare sensing and edge-computing architectures	Communication, local processing, responsiveness, and privacy	Provides the architectural basis for the wireless-capable and edge-oriented structure of cStick 2.0.

Table 2. Summary of Representative Commercial Fall-Related Devices and Assistive Applications.

Device/Application	Main Technology	Primary Function	Difference from the cStick 2.0 Design Scope
Apple Watch [27]	Wrist accelerometer, gyroscope, and location services	Hard-fall detection, emergency calling, and location sharing	Wrist-centered post-event support without cane-mounted grip sensing or forward-facing obstacle vision.
Samsung Galaxy Watch [28]	Wrist motion sensing and safety services	Hard-fall detection and emergency-contact notification	Does not integrate walking-stick interaction, obstacle direction, or tabular multimodal fall-risk classification.
Google Pixel Watch [29]	Smartwatch sensors, emergency calling, and location services	Fall detection and emergency assistance	Focuses on wearable incident response rather than combined walking-aid and environmental sensing.
Garmin devices [30]	Activity sensors, GPS, and Garmin Connect	Incident detection and contact notification during supported activities	Primarily activity-oriented and does not provide grip sensing or cane-mounted vision.
Medical-alert systems [31–36]	Help buttons, pendants, mobile units, GPS, and optional fall detection	Emergency response and monitoring-center or caregiver contact	Emphasize rapid assistance after an event rather than multimodal pre-event fall-risk classification.
Kanega Watch [37]	Medical-alert watch, fall detection, GPS, and voice interaction	Wearable emergency assistance	Provides emergency-support functions but is not a walking stick and does not include forward obstacle sensing.
Smart sticks and obstacle aids [24–26]	Ultrasonic, camera, LiDAR, GPS, audio, and vibration	Obstacle awareness, route assistance, and navigation	Strong in environmental assistance but generally does not combine grip interaction, physiological context, and three-state fall-risk classification.

cStick 2.0 builds on these directions by organizing fall-risk classification, embedded vision, cane interaction, sensing, local response, data accumulation, and future personalization within a common layered architecture. Its present contribution is a prototype-level technical integration supported by offline model evaluation and controlled obstacle-module testing, while synchronized real-world and older-adult validation constitute the next stage of development.

3. The Proposed IoMT-Based cStick System

cStick 2.0 follows a layered sensing, processing, response, and external model-development methodology for prototype-level fall-risk classification and embedded obstacle awareness. The architecture combines user-side sensing, local embedded processing, immediate buzzer feedback, wireless-capable hardware, external DNN development, CSV-based data accumulation, and a structured pathway for future personalization and caregiver-support services. Hardware-specific implementation procedures, training configurations, controlled obstacle testing, and quantitative results are presented separately in the implementation and results sections.

3.1. Study Hypothesis and Validation Scope

The working hypothesis of cStick 2.0 is that environmental, motion-related, physiological, and user–stick interaction variables can provide more discriminative information when evaluated together than when used independently. For the n-th tabular record, the multimodal input vector is represented as

$$\mathbf{x}_n = [d_n, p_n, h_n, g_n, s_n, a_n]^T, \tag{1}$$

where d_n denotes obstacle-distance information, p_n denotes grip-pressure information, h_n denotes heart rate variability (HRV), g_n denotes externally supplied blood glucose, s_n denotes SpO_2, and a_n denotes the accelerometer-derived motion feature. Equation (1) organizes the six variables according to their environmental, interaction-related, physiological, and motion-related roles.

The hypothesis does not require every variable to function as an independently validated or causal predictor of falls. Instead, the measurable contribution of an input or feature group is evaluated through controlled ablation. For a performance measure $\mathcal{M}$, the contribution associated with removing feature i is defined as

$$\Delta\mathcal{M}_i = \mathcal{M}(\mathcal{X}_{\text{all}}) - \mathcal{M}(\mathcal{X}_{\text{all}} \setminus \{x_i\}), \tag{2}$$

where $\mathcal{X}_{\text{all}}$ denotes the complete six-input configuration and $\mathcal{X}_{\text{all}} \setminus \{x_i\}$ denotes the configuration after removing feature i. Equation (2) provides a quantitative basis for examining feature complementarity through accuracy, balanced accuracy, macro F1-score, class-specific recall, and ROC-AUC rather than relying only on intuitive relationships.

The current 9,670-record development dataset contains independent transformed tabular observations derived from the publicly available cStick dataset. It does not preserve synchronized participant-specific sensor streams, original sampling windows, session identifiers, or independently annotated real-world fall events. The present evaluation therefore measures the ability of the tested models to reproduce the assigned three-class structure of the transformed dataset. The resulting evidence supports prototype-level multimodal classification but is not presented as a causal, clinical, or synchronized spatiotemporal validation.

A complete participant-level assessment will extend this foundation through timestamped sensor acquisition, participant- and session-separated evaluation, independently

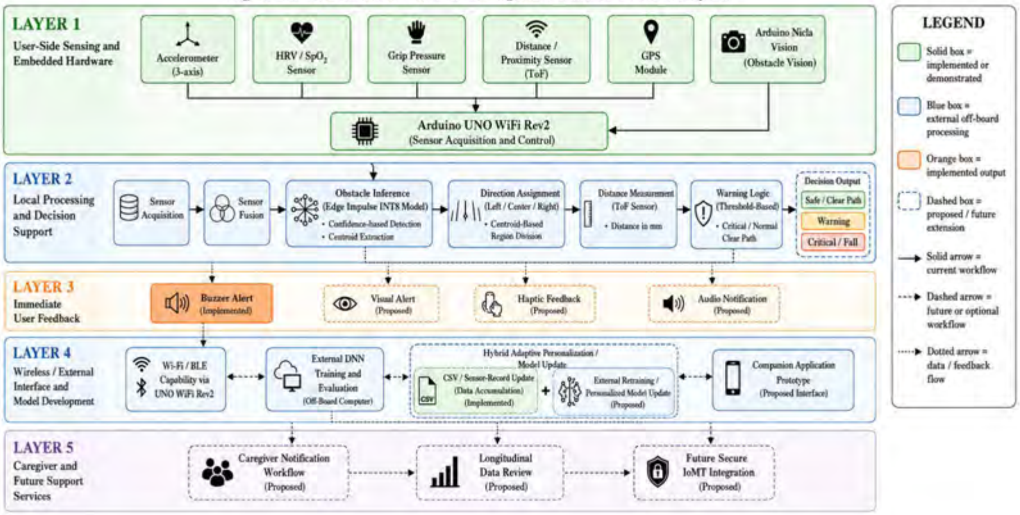

Figure 2. Layered cStick 2.0 System Framework Showing Implemented Sensing and Processing, Immediate Buzzer Feedback, Wireless Capability, External DNN Development, CSV Record Updates, and Proposed Personalization and Caregiver Support.

annotated walking events, and synchronized observation windows. Such an evaluation can determine how the multimodal variables evolve before, during, and after naturally occurring mobility-related conditions.

3.2. Proposed IoMT-Edge Architecture

The cStick 2.0 architecture follows a layered sensing–processing–response structure. Figure 2 organizes the framework into five functional layers and visually distinguishes implemented modules, externally executed processing, hardware-supported communication capability, and proposed extensions.

Layer 1 contains the user-side sensing and embedded hardware. Motion, HRV and SpO_2, grip pressure, obstacle proximity, and GPS information are acquired through the corresponding sensing modules. The Arduino UNO WiFi Rev2 supports sensor acquisition and control, while the Arduino Nicla Vision provides a dedicated embedded camera and local visual-processing platform. The UNO WiFi Rev2 also provides onboard Wi-Fi and Bluetooth Low Energy capability through the NINA-W102 module, thereby establishing a hardware-supported pathway for future connected IoMT operation without requiring an additional wireless module [12]. The Nicla Vision integrates a camera and dual-core embedded processor suitable for local computer-vision execution [13].

Layer 2 represents local processing and decision support. The implemented embedded workflow includes sensor acquisition, local data handling, obstacle-presence processing, obstacle-direction estimation, GPS acquisition, CSV record generation, and actuator-control logic. The DNN is not trained directly on the Arduino UNO WiFi Rev2. Instead, the compact DNN and comparison models are developed and evaluated on an external computing platform. Their evaluated output structure defines the prototype response states used by the hardware-control workflow.

Layer 3 represents immediate user feedback. Buzzer activation is implemented and functionally demonstrated in the present prototype. Visual alerts, haptic feedback, and spoken-audio notification are retained as proposed accessibility extensions. These proposed channels broaden the framework toward users with different visual, hearing, and tactile requirements without being presented as completed experimental functions.

Layer 4 represents wireless capability, external model development, and hybrid adaptive support. Newly acquired sensor records can be stored in the CSV repository through an implemented data-update pathway. The accumulated records can subsequently support external DNN retraining and user-specific model refinement. The CSV update is therefore implemented, whereas the retraining and quantitative personalization stages remain proposed and require before-and-after validation. The companion application is also represented as a proposed interface prototype and is not currently synchronized with the walking-stick hardware.

Layer 5 represents future caregiver and longitudinal-support services. Caregiver notification, long-term record review, secure application synchronization, and authenticated IoMT communication are structured extensions of the architecture. These functions are separated from the implemented prototype path through dashed boxes and arrows in Figure 2.

The layered representation prevents externally trained DNN models, proposed personalization, application interfaces, and caregiver services from being confused with functions already executed on the embedded hardware. At the same time, the architecture demonstrates how the implemented modules provide a practical foundation for progressively connected and personalized deployment.

3.3. Role of Sensor Inputs in Fall-Risk Classification and Assistive Response

Each cStick 2.0 input represents a distinct contextual dimension. Their roles support multimodal classification and assistive response, while their measurable value is examined through the feature-group evaluation rather than being assumed from their presence alone.

- **Accelerometer-derived motion information:** Represents changes in movement intensity and motion-related behavior. The transformed development dataset contains a scalar accelerometer-derived feature for each record rather than the original timestamped three-axis motion sequence.
- **Heart rate variability:** Represents cardiovascular timing variation and provides physiological context. HRV is normally derived from a sequence of heartbeat intervals and therefore operates over a different observation interval from instantaneous motion or proximity measurements.
- **Blood glucose:** Represents externally supplied contextual health information. cStick 2.0 does not contain an onboard glucose sensor, and the glucose field is not interpreted as a continuously measured or temporally synchronized signal. Its isolated and combined statistical contribution is examined through the feature-ablation analysis.
- **SpO_2:** Represents blood-oxygen saturation context derived from a physiological observation window. The current offline record contains a single SpO_2 value rather than the complete photoplethysmographic waveform used to obtain the estimate.
- **Grip pressure:** Represents the user's interaction with and dependence on the walking stick. Grip pressure is distinct from clinical blood pressure. The transformed dataset contains a grip-pressure feature, while the prototype architecture includes stick-mounted pressure sensing for future synchronized acquisition.
- **Obstacle distance:** Represents environmental proximity. The proximity sensor supports distance measurement, while the Nicla Vision module is evaluated separately for obstacle presence and left-, center-, or right-direction identification.
- **GPS location:** Supports location acquisition for future location-aware emergency services. GPS is not one of the six DNN classification inputs and is not used to determine the offline fall-risk class.

Together, the six model inputs describe environmental proximity, user–stick interaction, physiological context, and motion-related behavior. GPS provides an additional response-support function outside the DNN feature vector. This separation prevents location acquisition from being incorrectly interpreted as an input to the current classifier.

The conceptual roles of the variables do not establish independent clinical associations with observed falls. Blood glucose is externally supplied, HRV and SpO_2 represent physiological estimates derived over observation windows, and obstacle distance characterizes the environment rather than physiological status. Their measurable relationship with the assigned dataset classes is therefore reported through the feature-group analysis in Section 5.3.

3.4. Temporal Characteristics and Multimodal Alignment

The cStick 2.0 inputs operate at different temporal and contextual scales. Accelerometer, grip-pressure, obstacle-distance, and vision outputs can change rapidly during walking. HRV and SpO_2 are derived over physiological observation windows, while blood glucose is supplied less frequently through a CSV record or application interface. A timestamp-aware architecture must therefore evaluate measurement freshness before combining the inputs.

For a measurement from modality i, its temporal displacement from the reference inference time is defined as

$$\Delta t_{i,n} = \left| t_{i,n} - t_{\mathrm{ref},n} \right|, \tag{3}$$

where $t_{i,n}$ is the acquisition timestamp of modality i for inference record n, and $t_{\mathrm{ref},n}$ is the reference inference time. Based on Equation (3), the temporal-validity indicator is defined as

$$m_{i,n} = \begin{cases} 1, & \Delta t_{i,n} \le \tau_i, \\ 0, & \Delta t_{i,n} > \tau_i, \end{cases} \tag{4}$$

where τ_i is the permitted freshness interval for modality i. A value of $m_{i,n} = 1$ identifies a measurement that remains valid at the reference inference time, whereas $m_{i,n} = 0$ identifies a stale or temporally incompatible value.

The synchronized feature representation can subsequently be expressed as

$$\mathbf{x}_n^{\mathrm{sync}} = \left[m_{d,n} d_n, m_{p,n} p_n, m_{h,n} h_n, m_{g,n} g_n, m_{s,n} s_n, m_{a,n} a_n \right]^T, \tag{5}$$

where the validity indicators in Equation (4) determine whether the corresponding measurements may participate in the synchronized inference vector. In a deployable implementation, invalid values would be flagged for reacquisition, missing-data handling, or exclusion rather than silently interpreted as true zero-valued measurements.

Equations (3)–(5) formalize the planned timestamp-aware alignment mechanism. They were not applied to the present offline dataset because participant identifiers, session identifiers, original acquisition timestamps, sampling intervals, and source observation windows are unavailable.

Table 3 summarizes the temporal characteristics and current representation of the six classification inputs.

The current classifier is therefore evaluated as a multimodal tabular model rather than a synchronized temporal-fusion model. This distinction does not reduce the value of the feature-group analysis; instead, it defines precisely what the present model results demonstrate. The evaluation measures statistical complementarity within the transformed dataset, while Equations (3)–(5) establish the formal pathway for future synchronized acquisition.

Table 3. Temporal Characteristics and Current Evaluation Treatment of the cStick 2.0 Classification Inputs.

Input	Architectural Source	Temporal Characteristic	Current Offline Representation	Current Validation Boundary
Accelerometer	Motion-sensing interface	Rapidly varying motion signal	Single transformed accelerometer-derived scalar per record	Original three-axis waveform, timestamp, and motion window are unavailable
Grip pressure	Stick-mounted pressure interface	Rapidly varying user–stick interaction	Single transformed grip-pressure value per record	Pressure evolution and synchronized sampling interval are unavailable
Obstacle distance	Proximity-sensing interface	Sample-level environmental measurement	Single transformed distance value per record	Not temporally synchronized with the remaining offline features
HRV	Physiological-sensing interface	Derived from multiple heartbeat intervals	Single HRV value per record	Original heartbeat intervals and derivation window are unavailable
SpO_2	Physiological-sensing interface	Window-based physiological estimate	Single SpO_2 value per record	Original waveform, estimation window, and timestamp are unavailable
Blood glucose	CSV or application-supported external entry	Sparse contextual measurement	Single externally supplied glucose value per record	Not continuously measured or synchronized by the prototype

3.5. DNN-Based Fall-Risk Classification Model

The fall-risk classification module uses a compact DNN to process the six-dimensional vector defined in Equation (1). After feature normalization, the predicted class is represented as

$$\widehat{y}_n = f_{\boldsymbol{\theta}}(\widetilde{\mathbf{x}}_n), \qquad \widehat{y}_n \in \{0,1,2\}, \tag{6}$$

where $\widetilde{\mathbf{x}}_n$ denotes the normalized input vector, $f_{\boldsymbol{\theta}}(\cdot)$ denotes the DNN parameterized by $\boldsymbol{\theta}$, and $\widehat{y}_n$ is the predicted class. In Equation (6), class 0 denotes the assigned safe or no-fall state, class 1 denotes the assigned slip, trip, or fall-predicted warning state, and class 2 denotes the assigned critical or definite-fall label. These labels originate from the cStick dataset structure and should not be interpreted as independently adjudicated clinical events in the transformed dataset.

The compact DNN is selected as a nonlinear tabular classifier capable of representing interactions among normalized environmental, pressure, physiological, and motion-related inputs while maintaining a small model size. Its selection is evaluated through comparison with Logistic Regression, SVM, Random Forest, XGBoost, TinyML Dense, TinyML INT8, accelerometer-only models, and a majority baseline.

The complete and reduced feature configurations are evaluated using a common data split and evaluation protocol. Equation (2) is then applied conceptually to interpret changes in accuracy, balanced accuracy, macro F1-score, critical-class recall, false-negative rate, and ROC-AUC when individual variables or feature groups are removed.

The DNN is trained and evaluated on an external computing platform. The Arduino UNO WiFi Rev2 performs sensor acquisition and response control, while the Arduino Nicla Vision executes the embedded obstacle-processing component. This separation preserves a practical embedded architecture without implying that DNN training or validated DNN inference was performed directly on the Arduino UNO WiFi Rev2.

3.6. Hybrid Data Accumulation and Adaptive Personalization Pathway

Personalization is an important design objective because walking behavior, grip patterns, physiological baselines, and mobility conditions vary among users. Prior adaptive fall-detection studies demonstrate the value of incorporating user-specific observations into model refinement [22,23]. cStick 2.0 supports this objective through a hybrid implementation that separates the completed data-accumulation stage from the proposed adaptive-retraining stage.

During the implemented stage, newly acquired sensor records are appended to the existing CSV repository. Let $\mathcal{D}_t$ denote the available dataset at update cycle t, and let $\mathcal{D}_{\text{new},t}$ denote the newly collected records. The updated repository is defined as

$$\mathcal{D}_{t+1} = \mathcal{D}_t \cup \mathcal{D}_{\text{new},t}. \tag{7}$$

Equation (7) represents the implemented CSV sensor-record update mechanism. The new rows retain the same feature structure as the existing repository and can be inspected before subsequent model development.

The second stage uses the updated repository for proposed external retraining. The corresponding parameter-estimation objective is

$$\boldsymbol{\theta}_{t+1} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{t+1}; \boldsymbol{\theta}), \tag{8}$$

where $\mathcal{L}$ denotes the DNN training loss and $\boldsymbol{\theta}_{t+1}$ denotes the parameters obtained after external retraining. Equation (8) is not executed as an on-device Arduino training procedure. It represents the external refinement stage supported by the implemented data-accumulation pathway.

The quantitative benefit of personalization can later be measured using

$$\Delta\mathcal{M}_u = \mathcal{M}\big(\boldsymbol{\theta}_{t+1}, \mathcal{D}_u^{\text{test}}\big) - \mathcal{M}\big(\boldsymbol{\theta}_t, \mathcal{D}_u^{\text{test}}\big), \tag{9}$$

where $\mathcal{D}_u^{\text{test}}$ is a user-specific held-out evaluation set and $\Delta\mathcal{M}_u$ measures the change in a selected metric after personalization. Equation (9) defines the required before-and-after evaluation but is not reported as a completed result because longitudinal user-specific testing has not yet been conducted.

Figure 2 reflects this hybrid status through a solid box and arrow for the implemented CSV sensor-record update, followed by a dashed box and arrow for proposed external retraining and personalized model refinement. This representation preserves the demonstrated data-update contribution while avoiding an unsupported claim of measured personalization improvement.

3.7. Obstacle Detection and Warning Logic

cStick 2.0 integrates embedded visual inference and time-of-flight (ToF) proximity sensing to provide obstacle awareness within the user's walking path. The Arduino Nicla Vision operates as the dedicated edge-vision processor, while the Arduino UNO WiFi Rev2 performs sensor integration, display control, and buzzer activation.

The obstacle-detection model was developed using the Edge Impulse platform. Indoor obstacle and clear-path images were prepared for model training, validation, and testing, and the obstacle regions were labeled during dataset preparation. After training, the model was quantized to an INT8 representation to reduce its memory and computational requirements. The quantized model was then exported as OpenMV-compatible firmware and deployed directly to the Arduino Nicla Vision.

During operation, the onboard camera continuously captures the area in front of the walking stick. Each camera frame is resized and formatted according to the input dimensions and pixel representation used during model development. The preprocessed frame is then supplied to the locally deployed Edge Impulse inference model.

Let $q_t \in [0,1]$ denote the highest obstacle-confidence score produced for frame t, and let $\tau_q \in [0,1]$ denote the fixed confidence threshold used during the controlled evaluation. The obstacle-presence state o_t is defined as

$$o_t = \begin{cases} 1, & q_t \geq \tau_q, \\ 0, & q_t < \tau_q, \end{cases} \tag{10}$$

where $o_t = 1$ indicates an obstacle-present condition and $o_t = 0$ indicates a clear-path condition. Equation (10) was applied using the same confidence threshold throughout the controlled evaluation.

When an obstacle is detected, the horizontal centroid of the predicted obstacle region is used to determine its relative direction. Let $c_{x,t}$ denote the horizontal centroid coordinate, expressed in pixels, and let W denote the camera-frame width in pixels. The direction label r_t is defined as

$$r_t = \begin{cases} \text{Left}, & c_{x,t} < \dfrac{W}{3}, \\ \text{Center}, & \dfrac{W}{3} \leq c_{x,t} \leq \dfrac{2W}{3}, \\ \text{Right}, & c_{x,t} > \dfrac{2W}{3}. \end{cases} \tag{11}$$

Equation (11) divides the camera frame into three equal vertical regions and transforms the image-space centroid into left-, center-, or right-direction information. When $o_t = 0$, the direction output is assigned as $r_t = \text{None}$.

Obstacle distance is obtained independently using the ToF proximity-sensing component associated with the Nicla Vision implementation. Let d_t^{ToF} denote the ToF distance measured for frame t, expressed in millimeters. The distance measurement is recorded together with the obstacle-presence and direction outputs and is not estimated from the apparent width of the detected image region.

The warning state s_t is determined using the obstacle-presence decision and the measured ToF distance:

$$s_t = \begin{cases} \text{Critical}, & o_t = 1 \text{ and } d_t^{\text{ToF}} < \tau_c, \\ \text{Normal obstacle}, & o_t = 1 \text{ and } d_t^{\text{ToF}} \geq \tau_c, \\ \text{Clear path}, & o_t = 0, \end{cases} \tag{12}$$

where $\tau_c > 0$ denotes the configurable critical-distance threshold in millimeters. The current embedded code uses $\tau_c = 50$ mm as a bench-level threshold for a very close obstacle. This value supports controlled prototype operation and is not presented as a universal mobility-safety threshold. A deployment-oriented threshold can subsequently be configured according to walking speed, sensor placement, expected response time, and user-specific mobility requirements.

For every processed frame, the Nicla Vision produces an obstacle-presence state, direction label, ToF distance measurement, and processing frame rate. These values are transmitted through the serial interface to the Arduino UNO WiFi Rev2. The UNO uses the received information to update the display and activate the buzzer according to the warning state.

The complete obstacle-processing pathway is therefore

$$\begin{aligned} &\text{Camera acquisition} \rightarrow \text{Image preprocessing} \rightarrow \text{Edge Impulse inference} \\ &\rightarrow \text{Obstacle decision} \rightarrow \text{Direction assignment} \rightarrow \text{ToF distance measurement} \\ &\rightarrow \text{Serial communication} \rightarrow \text{Display and buzzer response.} \end{aligned}$$

This division allows the Arduino Nicla Vision to perform embedded visual inference and proximity acquisition locally, while the Arduino UNO WiFi Rev2 remains responsible for sensor integration and actuator control.

3.8. Implemented and Proposed Feedback Mechanisms

The response layer is organized according to implementation status. Buzzer activation is implemented and demonstrated as the immediate feedback mechanism. When the prototype response logic identifies a warning or critical state, the controller can activate the buzzer to draw the user's attention.

Visual mobile alerts, haptic feedback, and spoken-audio notification are proposed extensions. These channels broaden accessibility for users with different sensory requirements but are not reported as implemented or quantitatively evaluated functions in the current prototype.

The GPS module is integrated for location acquisition. This implemented capability provides the location input required for future emergency-support services. Automatic caregiver email, application-triggered emergency notification, and remote caregiver communication have not been transmitted by the current hardware and are therefore represented as proposed workflows in Figure 2.

This separation preserves the broader assistive vision of cStick 2.0 while ensuring that implemented buzzer and GPS functions are not combined with proposed communication and multimodal-feedback services under a single evaluation claim.

3.9. Edge-Based Data Handling and Privacy-Aware Design

The edge-oriented structure of cStick 2.0 places immediate sensing, obstacle processing, CSV record generation, GPS acquisition, and actuator-control functions near the user. Local execution supports responsive prototype operation and reduces the requirement to continuously transmit physiological, mobility, and location information.

The Arduino UNO WiFi Rev2 provides hardware support for Wi-Fi and Bluetooth Low Energy communication, while the onboard secure element provides a foundation for credential and key protection [12]. These capabilities establish a technically suitable path for future IoMT connectivity. However, the present evaluation does not report a completed sensor-to-application wireless link, communication latency, encrypted caregiver transmission, or authenticated remote access.

A deployable communication layer can extend the current architecture through authenticated device enrollment, transport-layer encryption, encrypted local or remote storage, role-based caregiver access, secure key management, and event-limited transmission of health and GPS data. These security functions are represented as future secure-IoMT integration in Figure 2.

The resulting architecture combines implemented local processing with a clearly structured pathway toward secure communication, external model refinement, application synchronization, and caregiver support. This layered separation allows each extension to be evaluated independently before inclusion in a complete end-to-end assistive deployment.

4. Implementation and Validation

The cStick 2.0 implementation combines embedded sensing, vision-based obstacle awareness, externally executed fall-risk classification, local actuator response, GPS acquisition, CSV-based record accumulation, and a standalone application interface prototype. The Arduino UNO WiFi Rev2 performs sensor acquisition, local data handling, display output, GPS acquisition, CSV record generation, and buzzer control. The Arduino Nicla Vision performs embedded obstacle-presence processing and directional localization. The compact Deep Neural Network (DNN) and comparison models are trained and evaluated on an external computing platform, after which the predicted class is supplied to the prototype response logic for automatic buzzer activation and display output.

This separation assigns time-sensitive sensing and response functions to the embedded hardware while retaining computationally intensive model training and evaluation on an external platform. The implementation is organized into component-status reporting, obstacle-module validation, DNN-based fall-risk classification, hardware integration, hybrid data accumulation and personalization support, GPS operation, and the cStick 2.0 application interface prototype.

4.1. Component Implementation and Evaluation Status

The cStick 2.0 architecture contains modules at different stages of implementation and evaluation. Table 4 distinguishes quantitatively evaluated modules, functionally demonstrated hardware, implemented data-handling pathways, interface prototypes, and proposed extensions. A component is identified as quantitatively evaluated only when numerical performance evidence is available. A functionally demonstrated component was operated as part of the prototype but was not subjected to a participant-level or statistically powered evaluation.

4.2. Obstacle Detection System

The obstacle-awareness module was implemented using an INT8-quantized Edge Impulse object-detection model deployed directly on the Arduino Nicla Vision. The module performs local image acquisition, preprocessing, inference, obstacle-presence classification, centroid-based direction assignment, ToF distance measurement, and serial-output generation. Local inference eliminates the need to continuously transmit camera frames to an external computer or cloud service and supports responsive warning generation.

The obstacle-presence, direction, and warning decisions follow Equations (10)–(12), as defined in Section 3.7. Figure 3 shows the actual Arduino Nicla Vision/OpenMV inference output, including the detected obstacle region, confidence score, centroid-based direction assignment, processing frame rate, and user-facing warning status. The ToF distance and complete output record are transmitted separately to the Arduino UNO WiFi Rev2 through serial communication.

4.2.1. Edge Impulse Model Deployment and Embedded Inference

An object-detection model was developed using Edge Impulse. Images representing indoor obstacles and clear walking paths were organized into training, validation, and testing subsets. Obstacle regions were labeled during dataset preparation so that the model could learn to distinguish obstacle-containing frames from clear-path frames.

After training, the model was quantized to an INT8 representation. Quantization reduced the model memory requirement and computational complexity while maintaining an embedded-compatible inference structure. The resulting model was exported as OpenMV-compatible firmware and deployed to the Arduino Nicla Vision.

For each frame I_t, the captured image is resized and formatted according to the model input configuration. The preprocessed frame is supplied to the Edge Impulse model, which generates predicted obstacle regions and their associated confidence values. The highest confidence value q_t is compared with the fixed threshold τ_q using Equation (10). The same value of τ_q was maintained throughout the controlled evaluation to preserve a consistent inference configuration.

Table 4. Implementation and Evaluation Status of the cStick 2.0 Components.

Component	Current Status	Available Evidence	Current Validation Boundary
Fall-risk classifier	Implemented and quantitatively evaluated offline	Common held-out evaluation, model comparison, feature ablation, class-wise analysis, and critical-class false-negative analysis	Evaluated using an expanded tabular development dataset rather than synchronized participant-level streams; DNN execution was external to the Arduino UNO WiFi Rev2
Arduino Nicla Vision obstacle module	Implemented and quantitatively evaluated on embedded hardware	INT8 Edge Impulse inference, confidence-based obstacle detection, centroid-based direction assignment, onboard ToF distance measurement, serial output, 144 controlled physical trials, 75.00% obstacle-presence accuracy, and approximately 19–20 FPS	No older-adult route evaluation; complete physical-test confusion-matrix, direction-error, distance-error, and repeated frame-timing records were not retained for reproducible aggregate reporting
Arduino UNO WiFi Rev2 sensor interface	Hardware integrated and functionally demonstrated	Sensor acquisition, local record handling, reception of Nicla Vision serial output, GPS acquisition, display output, CSV storage, and actuator control	Long-duration operation, calibration drift, missing-data behavior, and complete end-to-end latency were not quantitatively evaluated
Buzzer response	Implemented and functionally demonstrated	Externally generated DNN output and Nicla Vision obstacle information were supplied to the response logic for automatic buzzer activation	Repeated activation reliability and response latency were not quantified
Display output	Implemented and functionally demonstrated	Sensor readings, fall-risk states, obstacle status, direction, and distance information were actively displayed during prototype operation	Formal readability, accessibility, refresh-rate, and user-response evaluations were not conducted
GPS module	Hardware integrated and functionally demonstrated	GPS location acquisition during prototype operation	Location error, time-to-first-fix, outdoor reliability, and remote transmission performance were not quantified
Temperature and humidity sensor	Auxiliary circuit component	Physically included in the prototype connection circuit	Not used as a DNN input, obstacle-processing input, or reported evaluation variable
CSV sensor-record update	Implemented and functionally demonstrated	New sensor records can be appended to the CSV repository using the established feature schema	Longitudinal participant-specific data accumulation was not evaluated
External adaptive retraining	Supported proposed workflow	The expanded CSV repository can be transferred to the external training pipeline for subsequent model refinement	No user-specific before-and-after accuracy, calibration, false-alert, or longitudinal personalization result is claimed
Application interface	Standalone interface prototype	Manual record entry, statistical menu access, and visualization of representative sensor and decision records	Not connected to the physical sensors or Arduino hardware and not evaluated through usability or caregiver studies
Visual, haptic, and spoken-audio feedback	Proposed accessibility extensions	Included in the layered system architecture	Not implemented or quantitatively evaluated in the current prototype
Caregiver and remote emergency notification	Proposed communication extension	GPS acquisition and wireless-capable hardware provide an architectural foundation	No email, caregiver alert, or remote emergency message was transmitted during the current evaluation
Secure wireless IoMT communication	Proposed system extension	UNO WiFi Rev2 wireless capability and available secure-element support	No complete authenticated and encrypted sensor-to-application communication pathway was evaluated
Older-adult end-to-end assessment	Planned validation stage	Controlled component-level and offline model evaluations establish the technical foundation	Requires institutional approval, informed consent, participant-level safety procedures, synchronized acquisition, and usability assessment

4.2.2. Direction Assignment and ToF Distance Measurement

When Equation (10) produces $o_t = 1$, the horizontal centroid $c_{x,t}$ of the highest-confidence obstacle region is obtained. The direction label r_t is then assigned using Equation (11). When no obstacle is detected, the direction is assigned as $r_t = \text{None}$.

The ToF proximity sensor produces the distance measurement d_t^{ToF} in millimeters. This measurement is obtained independently from the image-inference result and is included in the embedded output record. Equation (12) combines o_t and d_t^{ToF} to assign the warning state s_t as critical, normal obstacle, or clear path.

The current embedded implementation uses $\tau_c = 50$ mm as a bench-level threshold for identifying a very close obstacle. This threshold is configurable and can be revised during participant-level evaluation according to walking speed, reaction time, sensor placement, and user-specific mobility characteristics.

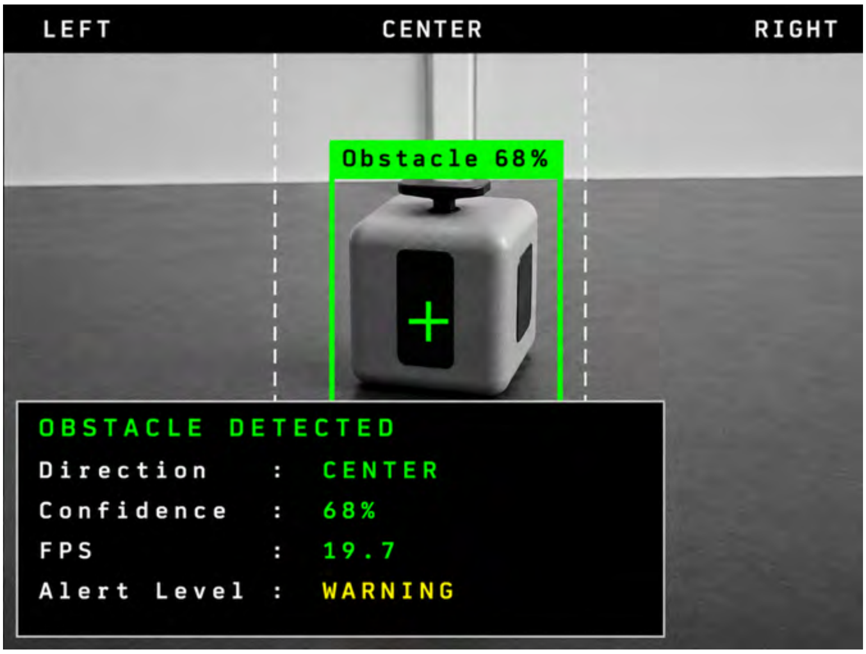

Figure 3. Arduino Nicla Vision Edge Impulse Output Showing Obstacle Confidence, Direction Assignment, Frame Rate, and Warning Status.

4.2.3. Serial Communication and Actuator Response

For each processed frame, the Arduino Nicla Vision generates four primary outputs: obstacle-presence state o_t, direction label r_t, ToF distance d_t^{ToF}, and processing frame rate FPS_t. These values are formatted as a compact serial record. A representative record is

```
OBS=1, DIR=CENTER, FPS=19.7
```

where `OBS=1` indicates obstacle presence, `DIR=CENTER` indicates the assigned horizontal direction, and `FPS=19.7` indicates the instantaneous processing rate. For a clear-path frame, the obstacle state is set to zero and the direction is reported as none.

In the user-facing display shown in Figure 3, a detected noncritical obstacle is presented using the "WARNING" label, corresponding to the normal-obstacle state defined in Equation(12).

The serial record is transmitted to the Arduino UNO WiFi Rev2. The UNO receives the obstacle state, direction, distance, and warning information and uses the received values to update the display and control the buzzer. A critical obstacle activates the immediate buzzer response, while normal-obstacle and clear-path states are displayed according to the implemented response logic.

Let $\Delta t_t > 0$ denote the complete processing time for frame t, expressed in milliseconds. The instantaneous processing rate is calculated as

$$\mathrm{FPS}_t = \frac{1000}{\Delta t_t}, \tag{13}$$

where the factor 1000 converts milliseconds to seconds. The measured processing interval includes camera acquisition, image preprocessing, Edge Impulse inference, output processing, ToF acquisition, and serial-record generation.

The observed embedded processing rate was approximately 19–20 FPS. A mean and standard deviation are not reported because complete repeated frame-level timing records were not retained for reproducible aggregate calculation.

The complete embedded processing procedure is summarized in Algorithm 1.

Algorithm 1 Edge Impulse Obstacle Detection, Direction Assignment, and ToF Measurement

Require: Camera stream I_t, deployed INT8 model, confidence threshold τ_q, critical-distance threshold τ_c, and frame width W

Ensure: Obstacle state o_t, direction r_t, ToF distance d_t^{ToF}, warning state s_t, and frame rate FPS_t

Initialize the Arduino Nicla Vision camera.
Initialize the deployed Edge Impulse INT8 model.
Initialize the ToF proximity sensor.
Initialize serial communication with the Arduino UNO WiFi Rev2.
while the obstacle module is active **do**
    Start the frame-processing timer.
    Capture camera frame I_t.
    Resize and format I_t according to the model input configuration.
    Execute Edge Impulse inference on the preprocessed frame.
    Obtain the highest obstacle confidence q_t.
    Determine o_t using Equation (10).
    Read the ToF distance d_t^{ToF}.
    if $o_t = 1$ **then**
        Obtain the horizontal centroid $c_{x,t}$ of the highest-confidence obstacle region.
        Assign r_t using Equation (11).
    else
        Set $r_t \leftarrow$ None.
    end if
    Assign s_t using Equation (12).
    Stop the frame-processing timer and obtain Δt_t.
    Calculate FPS_t using Equation (13).
    Transmit o_t, r_t, d_t^{ToF}, s_t, and FPS_t to the Arduino UNO WiFi Rev2.
    Update the display and buzzer according to s_t.
end while

Algorithm 1 separates embedded vision processing from system-level actuator control. The Arduino Nicla Vision performs image inference, direction assignment, and ToF distance acquisition, while the Arduino UNO WiFi Rev2 integrates the resulting information with the display and buzzer response.

4.2.4. Edge Impulse Vision-Model Validation

The Edge Impulse validation environment was used to evaluate the trained vision model before controlled physical prototype testing. Precision, recall, and F1-score are defined as

$$\mathrm{Precision} = \frac{TP}{TP + FP}, \qquad \mathrm{Recall} = \frac{TP}{TP + FN}, \tag{14}$$

and

$$\mathrm{F1} = 2\frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \tag{15}$$

where TP denotes true-positive detections, FP denotes false-positive detections, and FN denotes false-negative detections.

The results in Table 5 characterize the model-validation stage. They are reported separately from the controlled physical prototype accuracy because the two evaluations were performed in different environments.

Table 5. Edge Impulse Vision-Model Validation Metrics.

Metric	Result
Precision	0.73
Recall	0.67
F1-score	0.70

4.2.5. Controlled Physical Obstacle-Module Evaluation

The deployed obstacle module was evaluated through a controlled component-level experiment. The evaluation included three representative indoor obstacle types: a person, a chair, and a cardboard box. Each obstacle was positioned at reference distances of 50, 100, and 150 cm and at left, center, and right positions under normal and dim indoor lighting. Both stationary-camera and moving-camera conditions were included.

One trial was conducted for each unique obstacle–distance–direction–lighting–camera combination. The number of obstacle-present trials was therefore

$$N_{\text{obs}} = N_o N_d N_r N_l N_c = 3 \times 3 \times 3 \times 2 \times 2 = 108, \tag{16}$$

where N_o denotes the number of obstacle types, N_d denotes the number of reference distances, N_r denotes the number of horizontal directions, N_l denotes the number of lighting conditions, and N_c denotes the number of camera-motion conditions.

An additional 36 clear-path trials were conducted without an obstacle in the monitored region. The total number of controlled trials was

$$N_{\text{total}} = N_{\text{obs}} + N_{\text{clear}} = 108 + 36 = 144, \tag{17}$$

where N_{clear} denotes the number of clear-path trials.

A controlled trial was counted as correct when the module correctly distinguished an obstacle-present condition from a clear-path condition. Obstacle-presence accuracy was calculated as

$$\text{Accuracy}_{\text{obs}} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100, \tag{18}$$

where N_{correct} denotes the number of correctly classified trials. A total of 108 trials were classified correctly, resulting in

$$\text{Accuracy}_{\text{obs}} = \frac{108}{144} \times 100 = 75.00\%. \tag{19}$$

The controlled experiment was distributed across six sessions. Each session lasted approximately 10–12 min, producing a total evaluation duration of approximately 60–72 min. The prototype was operated by an adult researcher. No older-adult participants or unrestricted walking routes were included.

The controlled physical experiment demonstrates obstacle-presence processing under varied obstacle, distance, direction, lighting, and camera-motion conditions. Complete trial-level confusion-matrix, direction-label, distance-error, and repeated frame-timing records were not retained for reproducible aggregate analysis. Precision, recall, F1-score, false-positive rate, false-negative rate, direction accuracy, distance-estimation MAE, distance-estimation RMSE, mean FPS, and FPS standard deviation are therefore not numerically claimed for the physical evaluation.

Table 6. Controlled Evaluation of the cStick 2.0 Obstacle-Awareness Module.

Evaluation Property	Result
Obstacle types	Person, chair, and cardboard box
Reference distances	50, 100, and 150 cm
Horizontal directions	Left, center, and right
Lighting conditions	Normal and dim indoor lighting
Camera conditions	Stationary and moving
Trials per unique condition	1
Obstacle-present trials	108
Clear-path trials	36
Total controlled trials	144
Correct classifications	108
Incorrect classifications	36
Obstacle-presence accuracy	75.00%
Controlled sessions	6
Total evaluation duration	Approximately 60–72 min
Observed processing rate	Approximately 19–20 FPS
Older-adult participants	None
Walking routes	None
Evaluation operator	Adult researcher
Evaluation scope	Controlled component-level validation

4.3. DNN-Based Fall-Risk Classification

4.3.1. Dataset Origin, Transformation, and Synchronization Scope

The fall-risk classifiers were evaluated using a 9,670-record development dataset derived from the publicly available dataset associated with the earlier cStick framework [14, 15]. The source dataset contained 2,039 records. Data cleaning, numerical transformation, and controlled augmentation produced an additional 7,631 records while retaining the six-input and three-class organization of the source dataset. The resulting development dataset therefore contains

$$N_{\text{data}} = N_{\text{source}} + N_{\text{aug}} = 2039 + 7631 = 9670, \tag{20}$$

where N_{source} denotes the number of source records and N_{aug} denotes the number of augmented records.

Each record contains obstacle distance, stick pressure or grip pressure, HRV, externally supplied blood glucose, SpO_2, continuous acceleration information, and an assigned decision label. The augmentation process expands the available development data for common model comparison; it does not convert the dataset into a clinical deployment cohort.

The 9,670 records were not collected as synchronized measurements from older adults operating the complete cStick 2.0 prototype. Participant identifiers, session identifiers, acquisition timestamps, sensor-specific sampling intervals, and the original physiological and motion windows are unavailable. Each row is therefore evaluated as an independent transformed tabular observation.

The assigned decision labels follow the source cStick structure:

$$y_n = \begin{cases} 0, & \text{safe or no-fall-labeled condition,} \\ 1, & \text{warning, slip, trip, or fall-predicted condition,} \\ 2, & \text{critical or definite-fall-labeled condition.} \end{cases} \tag{21}$$

Equation (21) defines the supervised classes used in the current model evaluation. These assigned labels are not presented as independently adjudicated clinical fall events.

The current records are analyzed as independent tabular observations. Synchronized participant- and session-level acquisition will extend this dataset structure by incorporating timestamps, sensor-specific observation windows, and independently annotated mobility events.

Table 7. Composition and Evaluation Scope of the Fall-Risk Development Dataset.

Dataset Property	Description
Source dataset	Publicly available cStick fall prediction and detection dataset [14,15]
Source records	2,039 records
Augmented records	7,631 records generated through controlled data cleaning, transformation, and augmentation
Total development records	9,670 records
Input variables	Obstacle distance, stick pressure or grip pressure, HRV, externally supplied blood glucose, SpO_2, and continuous acceleration
Decision classes	Class 0: Safe; Class 1: Warning; Class 2: Critical/Fall
Feature representation	One six-dimensional multimodal feature vector and one assigned decision label per record
Blood-glucose representation	Externally supplied contextual health variable obtained from the source dataset, manual entry, or CSV-supported records
Model-development use	Training and comparison of statistical, ensemble, DNN, TinyML, accelerometer-only, and majority-baseline classifiers
Evaluation scope	Prototype-level offline fall-risk classification using independent tabular records
Planned extension	Synchronized, timestamped, participant- and session-structured multimodal data acquisition

4.3.2. DNN Architecture and Training

The compact DNN processes the six-dimensional multimodal vector defined in Equation (1). The accelerometer input is represented as a continuous acceleration feature; no binary accelerometer status or $\pm 3g$ threshold is used in the final model.

Before training, each feature is standardized using statistics obtained from the training partition. For feature x_i, the normalized value $\widetilde{x}_i$ is

$$\widetilde{x}_i = \frac{x_i - \mu_i}{\sigma_i}, \tag{22}$$

where μ_i and σ_i denote the training-set mean and standard deviation of feature i, respectively. Equation (22) prevents variables with larger numerical ranges from dominating model optimization.

The DNN contains two fully connected hidden layers with 20 neurons each and rectified linear unit activation. The hidden-layer outputs are

$$\mathbf{z}_1 = \mathrm{ReLU}(\mathbf{W}_1 \widetilde{\mathbf{x}} + \mathbf{b}_1), \tag{23}$$

and

$$\mathbf{z}_2 = \mathrm{ReLU}(\mathbf{W}_2 \mathbf{z}_1 + \mathbf{b}_2), \tag{24}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are trainable bias vectors, and $\mathbf{z}_1$ and $\mathbf{z}_2$ denote the hidden representations.

The output layer produces three logits $\mathbf{o} = [o_0, o_1, o_2]$. The probability assigned to class j is

Algorithm 2 External DNN Classification and Embedded Response Logic

Require: Six-feature vector $\mathbf{x}$, trained parameters $\boldsymbol{\theta}$, normalization statistics μ_i and σ_i
Ensure: Predicted class $\widehat{y}$ and prototype response state

```
Obtain distance, stick-pressure, HRV, blood-glucose, SpO2, and continuous-acceleration values.
for each feature x_i in x do
    Normalize x_i using Equation (22).
end for
Compute z_1 using Equation (23).
Compute z_2 using Equation (24).
Compute class probabilities using Equation (25).
Select ŷ using Equation (26).
Pass the externally generated class output to the Arduino UNO WiFi Rev2 response logic.
if ŷ = 0 then
    Display the safe state and continue monitoring.
else if ŷ = 1 then
    Display the warning state and automatically activate the buzzer.
else if ŷ = 2 then
    Display the critical/fall state and automatically activate the buzzer.
    Retain the acquired GPS coordinates for future location-aware response services.
end if
Return ŷ.
```

$$P(y = j \mid \mathbf{x}) = \frac{\exp(o_j)}{\sum_{k=0}^{2} \exp(o_k)}, \qquad j \in \{0, 1, 2\}. \tag{25}$$

The predicted class is selected as

$$\widehat{y} = \arg\max_{j \in \{0,1,2\}} P(y = j \mid \mathbf{x}). \tag{26}$$

Equation (26) maps the maximum Softmax probability to the safe, warning, or critical/fall response category.

Categorical cross-entropy is used as the training objective:

$$\mathcal{L} = -\sum_{j=0}^{2} y_j \log P(y = j \mid \mathbf{x}), \tag{27}$$

where y_j denotes the one-hot encoded assigned label for class j. The DNN parameters $\boldsymbol{\theta}$ are optimized through stochastic gradient descent:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t), \tag{28}$$

where η denotes the learning rate and t denotes the optimization iteration. The compact DNN was trained for 200 epochs using a batch size of 32 and a learning rate of 0.01.

The model is trained and evaluated externally rather than on the Arduino UNO WiFi Rev2. The external execution supports reproducible model comparison while allowing the embedded controller to focus on sensing, display output, CSV storage, GPS acquisition, and actuator response.

Algorithm 2 summarizes the external classification and embedded response sequence. It links the externally generated prediction to the implemented buzzer and display response

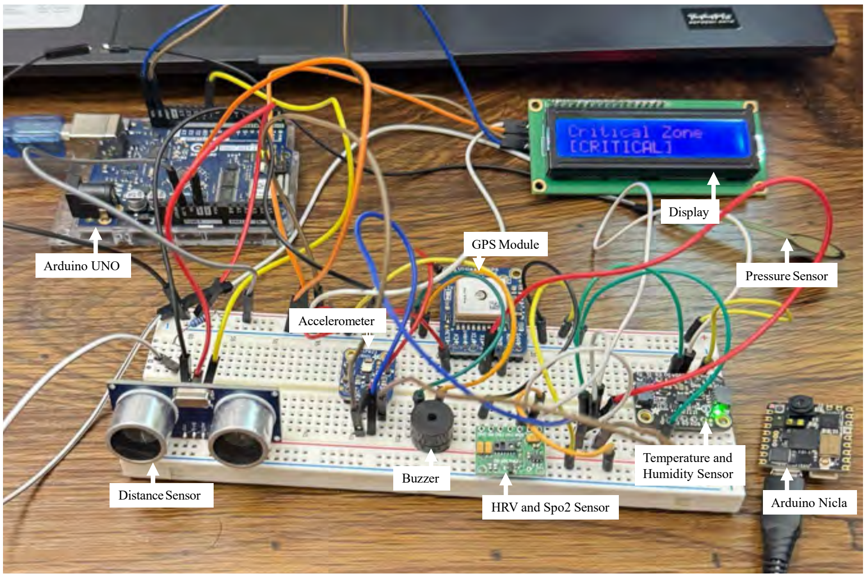

Figure 4. Hardware Implementation of the cStick 2.0 Prototype with Control, Motion, Physiological, Pressure, Distance, GPS, Display, Buzzer, and Environmental Sensing Modules.

without implying that DNN training or validated DNN inference occurs directly on the Arduino UNO WiFi Rev2. Visual alerts, haptic feedback, spoken-audio output, and remote caregiver communication remain proposed extensions.

4.4. Sensor Integration and Monitoring System

The cStick 2.0 hardware integrates motion, physiological, pressure, proximity, GPS, display, and buzzer components around the Arduino UNO WiFi Rev2. The Arduino Nicla Vision provides the dedicated embedded vision platform. Figure 4 identifies the principal sensing and response modules included in the prototype circuit.

The accelerometer provides continuous motion-related information. The pressure sensor measures stick pressure or grip pressure, which represents user interaction with the walking aid and does not represent clinical blood pressure. The HRV and SpO_2 interface provides physiological context, while the distance sensor supplies environmental proximity information. The GPS module acquires location coordinates, the display actively presents sensor and decision outputs, and the buzzer provides the implemented immediate warning response.

The temperature and humidity sensor shown in Figure 4 is physically included as an auxiliary circuit component. Its measurements are not included among the six DNN inputs, are not used by the obstacle-processing module, and are not included in the reported performance evaluation.

Blood glucose is supplied as external contextual health information. The current cStick 2.0 prototype does not contain an onboard glucose sensor and does not claim continuous glucose monitoring. A glucose value may be provided through the development dataset, manual interface entry, or CSV record. It is not interpreted as a real-time synchronized measurement.

The hardware architecture supports multimodal feature handling, but the current offline dataset does not preserve the timestamps and original observation windows required to verify temporal alignment among acceleration, pressure, HRV, SpO_2, blood glucose, and obstacle distance. Future synchronized records will follow the timestamp, freshness, and validity mechanism defined by Equations (3)–(5).

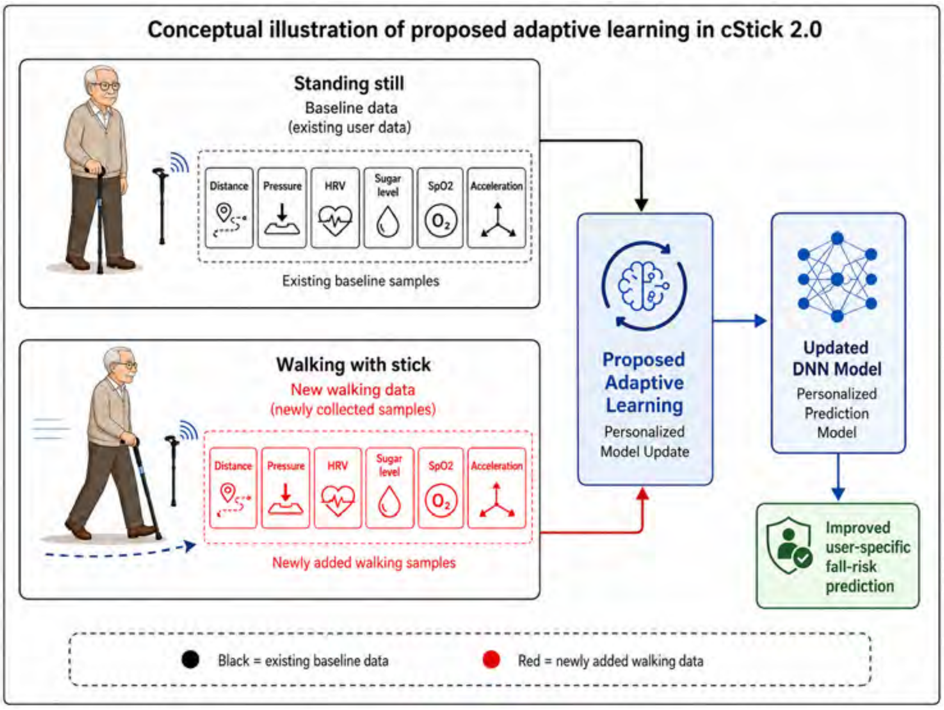

Figure 5. Conceptual Hybrid Adaptation Pathway in cStick 2.0 Showing Implemented Sensor-Record Accumulation and Proposed External Retraining for Future User-Specific Refinement.

4.5. Hybrid Data Accumulation and Adaptive Personalization

cStick 2.0 uses a hybrid adaptation pathway that separates implemented sensor-record accumulation from proposed external model retraining. Newly acquired records can be appended to the existing CSV repository using the same six-feature structure employed by the development dataset. This completed data-update capability provides a direct mechanism for constructing a progressively expanded record repository.

Figure 5 illustrates the hybrid pathway. Existing records are retained as the baseline repository, newly acquired sensor values are appended during subsequent operation, and the expanded CSV file can later support external model refinement. The figure represents the architectural adaptation process rather than a completed longitudinal participant experiment.

The implemented data-update stage follows Equation (7), in which newly collected records $\mathcal{D}_{\text{new},t}$ are appended to the existing repository $\mathcal{D}_t$. The proposed retraining stage follows Equation (8), in which the expanded repository is used to optimize a new set of DNN parameters externally.

Algorithm 3 summarizes the hybrid implementation.

The hybrid structure strengthens the personalization pathway without requiring computationally intensive training on the microcontroller. The current implementation demonstrates CSV-based accumulation, while future user-specific evaluation will quantify whether external retraining improves accuracy, calibration, critical-class recall, or false-alert performance.

4.6. Location Tracking and Future Emergency Support

The GPS module is integrated and functionally demonstrated for location acquisition. The controller can display or retain the acquired coordinates together with the current prototype response state. This location capability establishes the information source required for future caregiver or emergency-support communication.

Algorithm 3 Hybrid CSV Update and External Personalization Pathway

Require: Existing CSV repository $\mathcal{D}_t$ and newly acquired records $\mathcal{D}_{\text{new},t}$
Ensure: Updated repository $\mathcal{D}_{t+1}$ and optional externally refined model

1: Verify that each new row contains distance, stick pressure, HRV, blood glucose, SpO_2, continuous acceleration, and an assigned class field.
2: Validate the numerical format and column order of each record.
3: Append the accepted records using Equation (7).
4: Store the expanded CSV repository as $\mathcal{D}_{t+1}$.
5: **Implemented stage:** retain $\mathcal{D}_{t+1}$ for subsequent review and model development.
6: **Proposed stage:** transfer $\mathcal{D}_{t+1}$ to the external training environment.
7: **Proposed stage:** retrain the DNN using Equation (8).
8: **Proposed stage:** compare the original and refined models on a held-out user-specific evaluation set.
9: Return $\mathcal{D}_{t+1}$ and, when validated, the refined parameters.

The current prototype does not transmit email, caregiver alerts, or remote emergency messages. When $\hat{y} = 1$, the implemented response logic activates the buzzer and displays the warning state. When $\hat{y} = 2$, the buzzer and display communicate the critical/fall state, while the available GPS coordinates can be retained for a future authorized communication service.

The Arduino UNO WiFi Rev2 provides onboard Wi-Fi and Bluetooth Low Energy capability and includes hardware support for secure credential handling [12]. The Arduino Nicla Vision also includes hardware-level security capability suitable for future device authentication and protected key storage [13]. These hardware features provide a strong foundation for secure IoMT expansion without being presented as a completed end-to-end cybersecurity implementation.

A deployable communication pathway can extend the current architecture through authenticated device enrollment, encrypted transport, secure local or remote storage, role-based caregiver access, protected contact information, event-limited GPS transmission, audit logging, and secure key management. These functions are positioned as structured extensions that build upon the implemented local sensing, GPS acquisition, display, buzzer, and wireless-capable hardware.

4.7. cStick 2.0 Application Interface Prototype

The cStick 2.0 application was developed as a standalone interface prototype to demonstrate how sensor records, assigned decision classes, and analytical summaries may be presented in a future connected implementation. The application was created independently of the walking-stick hardware and was not synchronized with the Arduino UNO WiFi Rev2, Arduino Nicla Vision, or physical sensor network during the current evaluation.

Figure 6 presents the representative interface. The Add New Log screen provides fields for manually entering distance, pressure, HRV, blood glucose, SpO_2, and continuous acceleration information. The pressure field represents stick pressure or grip pressure and does not represent clinical arterial blood pressure. Clinical blood pressure is not included among the six DNN inputs.

As shown in Figure 6, the Statistics menu provides access to vital-sign trends, fall analytics, fall-versus-no-fall comparisons, average measurements, and decision-distribution views. The visualization panel illustrates how distance, stick pressure, HRV, warning states, and critical/fall-labeled records may be reviewed through a common interface.

The displayed values are representative interface records used to demonstrate organization and analytical presentation. They are not presented as a chronological participant

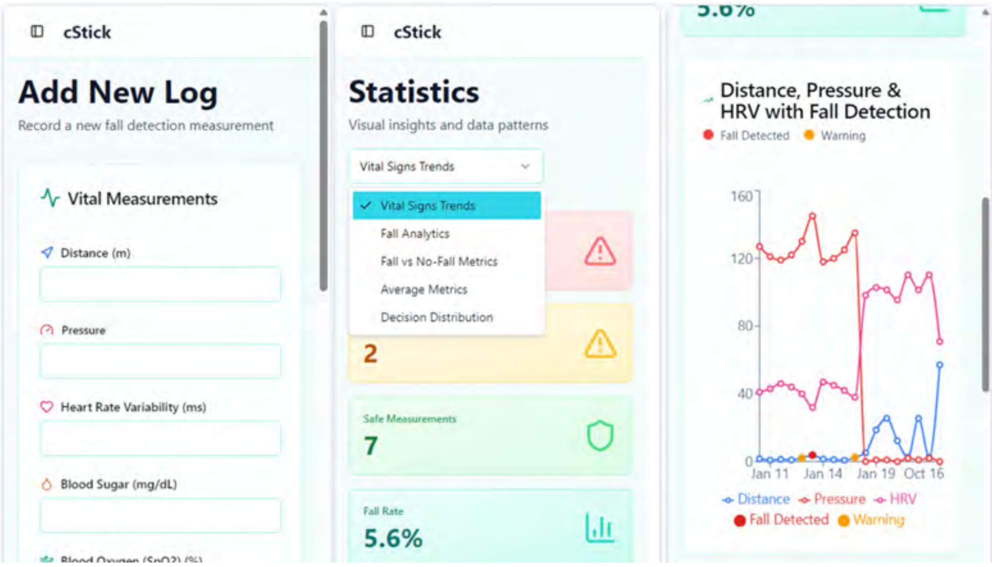

Figure 6. cStick 2.0 Application Prototype Showing Manual Sensor Entry, Statistical Views, and Fall-Risk Visualization. The Pressure Field Represents Stick or Grip Pressure, and the Displayed Records Are Illustrative Rather Than Synchronized Older-Adult Measurements.

history or as measurements obtained directly from the cStick 2.0 hardware. The application was not evaluated through usability, accessibility, caregiver, or task-completion studies.

Hardware-to-application synchronization, authenticated caregiver access, automatic email transmission, and remote emergency communication remain proposed extensions. Their future integration will use the hardware-supported wireless architecture after communication security, access control, data privacy, and delivery reliability have been validated.

4.8. System-Level Operation

At the system level, the Arduino UNO WiFi Rev2 acquires sensor information, stores CSV records, obtains GPS coordinates, controls the display, and activates the buzzer. The display actively communicates sensor readings and decision states. The Arduino Nicla Vision performs embedded obstacle-presence processing and direction estimation. The DNN is trained and evaluated externally, and its predicted class is supplied to the embedded response logic for automatic buzzer and display activation.

The implemented CSV-update pathway supports progressive record accumulation, while external retraining and user-specific performance refinement remain structured personalization stages. The application provides a representative standalone interface for manual data entry and analytical visualization without being presented as a hardware-connected application.

The resulting implementation combines quantitatively evaluated offline fall-risk classification, controlled embedded obstacle testing, functionally demonstrated sensing and response hardware, GPS acquisition, display output, automatic buzzer activation, and CSV-based data accumulation within one layered assistive architecture. These capabilities establish the technical basis for the comparative performance analysis, feature-group evaluation, and system-level discussion presented in the following section.

5. Results and Discussion

The cStick 2.0 evaluation combines three complementary forms of evidence: offline fall-risk classification using the expanded tabular development dataset, feature-group analysis using the compact DNN, and embedded obstacle-module evaluation using the Arduino Nicla Vision. The model-comparison results characterize classification and deployment-

Table 8. Model Performance and Deployment-Oriented Comparison in the External Benchmarking Environment.

Model	Accuracy (%)	Balanced Acc. (%)	Macro F1 (%)	ROC-AUC (%)	Size (KB)	Latency (ms)
TinyML INT8	**95.76**	**93.91**	**94.92**	–	3.336	0.024078
TinyML Dense	95.71	93.74	94.73	**97.60**	**1.074**	0.002247
cStick Compact DNN	95.40	92.35	93.90	97.44	2.434	0.002329
Random Forest	93.90	91.47	92.45	96.68	32350.045	0.082821
SVM (RBF)	93.74	92.42	92.37	96.92	138.943	0.112080
XGBoost	93.74	90.82	91.94	96.94	1166.093	0.013023
Logistic Regression	90.85	91.85	90.40	97.12	1.676	**0.000803**
Wearable Accelerometer RF	60.75	58.07	59.09	70.87	577.525	0.034217
Wearable Accelerometer LR	55.89	56.13	58.06	65.53	1.422	0.000700
Majority Baseline	58.43	33.33	24.59	50.00	0.481	0.000032

oriented tradeoffs in the external benchmarking environment, whereas the controlled obstacle experiment characterizes component-level embedded behavior. These evaluation modes are reported separately because the fall-risk classifiers and obstacle-awareness module use different inputs, execution platforms, and validation protocols.

5.1. Model Comparison and Deployment-Oriented Performance

The fall-risk classifiers were evaluated using the 9,670-record development dataset described in Section 4.3.1. Each record contains obstacle distance, stick pressure or grip pressure, HRV, externally supplied blood glucose, SpO_2, continuous acceleration, and an assigned safe, warning, or critical/fall class.

All comparison models were evaluated using a common data-partitioning and preprocessing procedure. Feature standardization was determined from the training partition, and the resulting transformation was applied to the held-out evaluation records. The reported model-size and latency measurements characterize the external benchmarking environment and should not be interpreted as measured inference latency on the Arduino UNO WiFi Rev2.

For a three-class problem containing $C = 3$ classes, balanced accuracy is defined as

$$\text{Balanced Accuracy} = \frac{1}{C}\sum_{c=0}^{C-1} \text{Recall}_c, \tag{29}$$

where Recall_c denotes the recall of class c. Equation (29) gives equal importance to the safe, warning, and critical/fall classes, independently of their relative frequency.

The macro F1-score is defined as

$$\text{Macro F1} = \frac{1}{C}\sum_{c=0}^{C-1} \text{F1}_c, \tag{30}$$

where F1_c denotes the F1-score of class c. The macro one-versus-rest ROC-AUC is expressed as

$$\text{Macro ROC-AUC} = \frac{1}{C}\sum_{c=0}^{C-1} \text{AUC}_c, \tag{31}$$

where AUC_c denotes the one-versus-rest area under the receiver operating characteristic curve for class c. Equations (29)–(31) provide a more complete interpretation than overall accuracy alone because they account for class balance, class-oriented error behavior, and decision separability.

Table 8 summarizes the statistical, ensemble, neural, TinyML, accelerometer-only, and majority-baseline results.

The three compact neural configurations achieved the strongest overall performance. TinyML INT8 obtained the highest accuracy of 95.76%, balanced accuracy of 93.91%, and

Table 9. Class-Oriented Performance of the Six-Input DNN Ablation Reference.

Assigned Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Safe	96.00	98.00	97.00	1130
Warning	92.00	92.00	92.00	608
Critical/Fall	98.00	86.73	92.00	196

macro F1-score of 94.92%. TinyML Dense achieved 95.71% accuracy and the highest reported ROC-AUC of 97.60%, while requiring only 1.074 KB in the external size estimate. The cStick compact DNN achieved 95.40% accuracy, 92.35% balanced accuracy, a macro F1-score of 93.90%, and a ROC-AUC of 97.44%.

The maximum accuracy difference among TinyML INT8, TinyML Dense, and the cStick compact DNN was only 0.36 percentage points. This close grouping indicates that compact neural architectures provide a favorable balance between classification performance and model size for the expanded development dataset. The cStick DNN remains the principal analytical model because its six-input structure is directly connected to the feature-group analysis and the proposed external personalization pathway.

Random Forest, SVM, and XGBoost also achieved strong performance, with accuracies between 93.74% and 93.90%. Random Forest required substantially more model storage than the compact neural models, while the RBF-SVM produced the highest measured latency among the evaluated classifiers. Logistic Regression was the smallest conventional multimodal baseline and produced a strong ROC-AUC of 97.12%, but its accuracy and macro F1-score remained below the compact neural models.

The accelerometer-only baselines were substantially weaker. Their accuracies ranged from 55.89% to 60.75%, and their macro ROC-AUC values remained below 71%. The majority baseline achieved 58.43% accuracy but only 33.33% balanced accuracy and 24.59% macro F1-score. This difference illustrates why overall accuracy alone is insufficient when one assigned class occurs more frequently than the others.

The deployment comparison and the feature-ablation analysis described in Section 5.3 represent separate experimental pipelines. The cStick DNN achieved 95.40% accuracy in the common deployment-oriented comparison, whereas the all-six-input reference model achieved 95.24% in the dedicated ablation analysis. The difference of 0.16 percentage points is retained transparently and does not alter the comparative interpretation of either experiment.

5.2. Class-Oriented DNN Performance

Safety-oriented evaluation requires particular attention to the critical/fall class because a missed critical record may be more consequential than an additional warning. For class c, the false-negative rate is defined as

$$\mathrm{FNR}_c = \frac{FN_c}{TP_c + FN_c} = 1 - \mathrm{Recall}_c, \tag{32}$$

where TP_c and FN_c denote the true-positive and false-negative counts for class c, respectively.

The dedicated all-six-input ablation reference contained 1,130 safe records, 608 warning records, and 196 critical/fall records in the held-out test partition. Table 9 summarizes the corresponding class-oriented results.

The critical/fall recall of 86.73% corresponds to 26 false-negative critical records among the 196 critical/fall records in the held-out partition. Substitution into Equation (32) gives a critical-class false-negative rate of 13.27%. The high critical-class precision indicates that predicted critical states were rarely assigned incorrectly, while the lower recall relative to

Table 10. Feature-Group Ablation Analysis for Fall-Risk Classification.

Input Configuration	Accuracy (%)	Balanced Acc. (%)	Macro F1 (%)	Class-2 Recall (%)	Class-2 FNR (%)	Macro ROC-AUC (%)
Continuous Acceleration Only	65.51	56.73	54.02	66.84	33.16	70.99
Physiological Inputs: HRV + SpO_2	61.07	45.57	42.48	36.73	63.27	69.17
Blood Glucose Only	58.43	33.33	24.59	0.00	100.00	54.71
Obstacle Distance Only	82.52	67.50	69.47	30.10	69.90	92.84
Stick Pressure Only	61.07	46.08	40.82	40.82	59.18	64.98
All Inputs Except Blood Glucose	92.76	89.33	90.91	83.16	16.84	97.16
All Inputs Except Stick Pressure	93.85	91.61	92.82	88.27	11.73	97.32
All Six Inputs	**95.24**	**92.47**	**93.87**	86.73	13.27	**97.46**

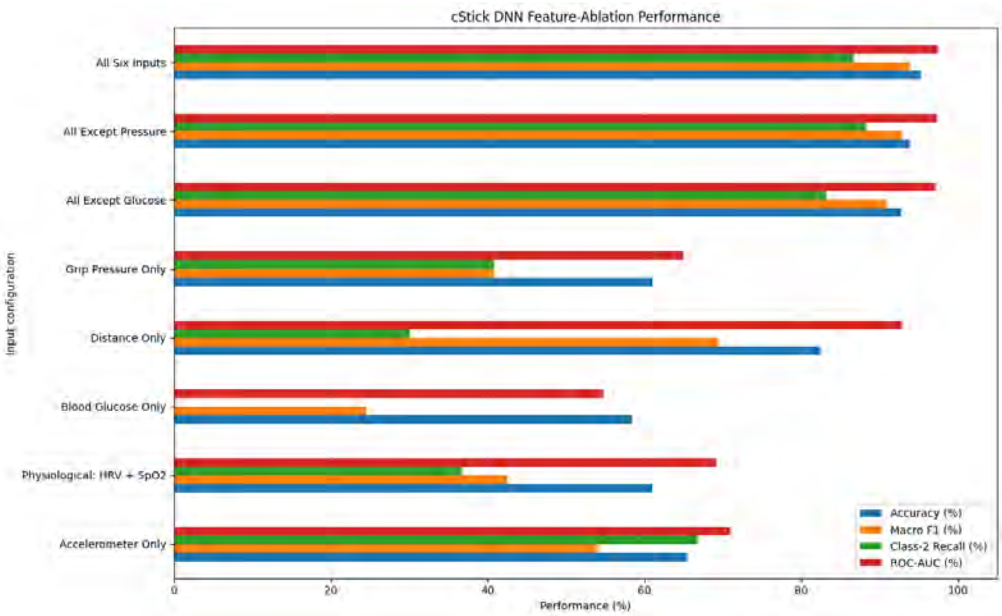

Figure 7. Feature-Ablation Performance of the Compact cStick DNN Across Complete and Reduced Input Configurations.

the safe and warning classes identifies missed critical records as an important target for subsequent model refinement.

The class-oriented analysis strengthens the overall accuracy result by showing that the six-input DNN does not obtain its performance solely through the larger safe class. Nevertheless, the critical-class false-negative rate remains a central safety metric for future participant- and session-independent evaluation.

5.3. *Feature-Group and Contextual-Input Analysis*

The six model inputs represent different sensing modalities and temporal characteristics. A dedicated ablation analysis was therefore conducted to determine whether individual variables or reduced feature groups could reproduce the performance of the complete DNN. The same compact architecture, feature normalization procedure, data-partitioning strategy, optimization settings, and evaluation metrics were applied across the tested configurations.

The evaluated configurations included continuous acceleration only, HRV and SpO_2, blood glucose only, obstacle distance only, stick pressure only, all inputs except blood glucose, all inputs except stick pressure, and the complete six-input vector.

Figure 7 visualizes the principal ablation metrics. Accuracy, macro F1-score, critical-class recall, and macro ROC-AUC are shown for every input configuration.

The complete six-input configuration achieved the strongest aggregate performance, with 95.24% accuracy, 92.47% balanced accuracy, a macro F1-score of 93.87%, and a macro ROC-AUC of 97.46%. No individual input or two-feature physiological configuration approached this combination of accuracy, class balance, and class separability.

Obstacle distance was the strongest individual variable, achieving 82.52% accuracy and a macro ROC-AUC of 92.84%. However, its balanced accuracy was 67.50%, and its critical-class recall was only 30.10%. The distance-only model therefore captured substantial label-related structure but was insufficient for balanced three-class classification.

The continuous-acceleration-only configuration achieved 65.51% accuracy, 56.73% balanced accuracy, and 66.84% critical-class recall. These results show that the continuous scalar acceleration feature contains useful movement-related information but does not reproduce the complete multimodal performance. Preserving timestamped three-axis acceleration and gyroscope sequences in subsequent data collection would permit direct analysis of gait transitions, orientation changes, and temporal motion windows.

The HRV and SpO_2 configuration achieved 61.07% accuracy, 45.57% balanced accuracy, and 36.73% critical-class recall. Stick pressure alone achieved 61.07% accuracy, 46.08% balanced accuracy, and 40.82% critical-class recall. These values show that the physiological and interaction-related inputs are not sufficiently discriminative when used independently, although they can contribute complementary context when combined with environmental and motion-related variables.

Blood glucose alone achieved 58.43% accuracy, 33.33% balanced accuracy, a macro F1-score of 24.59%, and zero recall for the critical/fall class. The glucose-only result therefore does not support blood glucose as an independent fall-risk classifier.

Removing blood glucose from the six-input vector reduced accuracy from 95.24% to 92.76%, reduced macro F1-score from 93.87% to 90.91%, and increased the critical-class false-negative rate from 13.27% to 16.84%. Within the transformed development dataset, the externally supplied glucose variable therefore contributes incremental statistical information when combined with the remaining inputs. This result does not demonstrate that glucose directly causes, temporally precedes, or independently predicts real-world falls.

Removing stick pressure reduced accuracy from 95.24% to 93.85% and macro F1-score from 93.87% to 92.82%. Critical-class recall increased slightly from 86.73% to 88.27%, indicating that stick pressure contributes to overall three-class performance but is not the dominant source of critical-class discrimination in the current test partition.

The ablation results support statistical complementarity among the six input variables within the expanded tabular dataset. Their combined performance is substantially stronger than the performance of individual variables or small feature groups. Because the records do not preserve synchronized timestamps, participant identifiers, session boundaries, or original signal windows, this complementarity should be interpreted as dataset-level statistical evidence rather than as a validated spatiotemporal association with independently observed fall events.

5.4. Embedded Obstacle-Module Results

The Arduino Nicla Vision obstacle-awareness module was evaluated through two separate stages: Edge Impulse model validation and controlled physical prototype testing. The two result sets are reported independently because they characterize different evaluation environments.

The deployed obstacle method combines four principal operations. Obstacle presence is obtained from the locally deployed INT8 Edge Impulse model, direction is assigned from the horizontal centroid of the detected region, distance is measured independently using the time-of-flight proximity sensor, and the resulting information is transmitted to the Arduino UNO WiFi Rev2 through serial communication.

The Edge Impulse model-validation stage produced a precision of 0.73, recall of 0.67, and F1-score of 0.70, as summarized in Table 5. These values characterize the model-development environment and are not combined with the physical-test accuracy.

The controlled physical experiment included 108 obstacle-present trials and 36 clear-path trials. The tested obstacles were a person, chair, and cardboard box positioned at 50, 100, and 150 cm. Left-, center-, and right-direction conditions, normal and dim indoor lighting, and stationary- and moving-camera operation were included.

A total of 108 of the 144 controlled trials were classified correctly. Equation (19) therefore gives an obstacle-presence accuracy of 75.00%. The embedded Edge Impulse pipeline operated within an observed range of approximately 19–20 FPS.

These results demonstrate that the Arduino Nicla Vision can execute quantized obstacle inference locally while also supporting centroid-based direction assignment, ToF distance acquisition, serial communication, and immediate display and buzzer response. The controlled experiment extends the module evaluation beyond frame-rate reporting by including varied obstacle types, distances, directions, lighting conditions, and camera-motion states.

Complete trial-level confusion-matrix, direction-label, distance-error, and repeated frame-level timing records were not retained in a form suitable for reproducible aggregate calculation. Consequently, physical-test precision, recall, F1-score, false-positive rate, false-negative rate, direction accuracy, distance-estimation MAE, distance-estimation RMSE, mean frame rate, and frame-rate standard deviation are not numerically claimed.

No older-adult participants or unrestricted walking routes were included in the controlled evaluation. The reported 75.00% accuracy therefore establishes component-level embedded feasibility rather than complete assistive-navigation effectiveness.

5.5. Integrated Interpretation of the cStick 2.0 Evaluation

The combined results show that cStick 2.0 brings together strong multimodal classification, compact deployment-oriented models, embedded obstacle awareness, local actuator response, GPS acquisition, display output, and CSV-based record accumulation within one layered assistive architecture.

The neural model comparison demonstrates that compact DNN and TinyML configurations can achieve approximately 95.4–95.8% accuracy with macro F1-scores approaching 95% on the expanded development dataset. The comparison with conventional statistical, ensemble, accelerometer-only, and majority baselines provides a broader evaluation than reliance on a single DNN result.

The feature-ablation analysis further demonstrates that the complete multimodal configuration provides the strongest overall performance. Obstacle distance was the most informative individual feature, but its low critical-class recall prevented it from serving as an adequate stand-alone classifier. Continuous acceleration, physiological variables, stick pressure, and blood glucose were also insufficient independently. Their combined use produced stronger accuracy, class balance, and ROC-AUC than any isolated feature configuration.

The glucose findings require a particularly careful interpretation. Blood glucose alone failed to identify the critical class, but removing it from the complete feature vector reduced aggregate performance. Within the transformed development dataset, glucose therefore functions as supplementary contextual information rather than as a stand-alone fall marker. The result does not establish a causal or synchronized physiological relationship with real-world fall events.

The controlled Arduino Nicla Vision evaluation complements the offline classifier analysis by addressing environmental hazards through embedded obstacle-presence processing. The measured 75.00% obstacle-presence accuracy and observed 19–20 FPS execution demonstrate practical component-level operation across multiple indoor testing conditions. Direction labeling and distance reporting further provide a structured basis for user-oriented warning logic, although their aggregate accuracy and error values require retained trial-level records in subsequent testing.

The externally generated DNN prediction was connected to the Arduino UNO WiFi Rev2 response logic, allowing automatic buzzer activation and active display output for warning and critical states. GPS acquisition and CSV sensor-record updates were also functionally demonstrated. The companion application remains a standalone interface prototype, while visual, haptic, spoken-audio, caregiver-notification, secure communication, and externally validated personalization functions remain structured extensions.

5.6. Current Evaluation Boundaries and Next Validation Stage

The current evidence is appropriately interpreted as prototype-level technical validation. The fall-risk results were obtained from a transformed and expanded tabular dataset containing 2,039 source records and 7,631 augmented records. The dataset does not preserve complete participant identifiers, session identifiers, synchronized timestamps, sensor sampling intervals, or original physiological and motion windows.

Future model evaluation will also retain complete three-class confusion matrices and report confidence intervals or repeated-run variability for accuracy, balanced accuracy, macro F1-score, critical-class recall, false-negative rate, and ROC-AUC. This analysis will quantify model stability and determine whether small performance differences among the compact DNN and TinyML configurations are statistically meaningful.

Acceleration, stick pressure, and obstacle distance can vary rapidly, whereas HRV and SpO_2 are derived over physiological observation intervals. Blood glucose is externally supplied and may have been acquired at a substantially different time from the remaining variables. The current records therefore do not demonstrate that all six inputs occurred within the same clinically meaningful interval.

The reported model performance establishes statistical complementarity and comparative classification capability within the expanded dataset. It does not establish a causal relationship between the input variables and falls, participant-independent clinical generalization, or prediction of naturally occurring falls in older adults.

Clinical arterial blood pressure is not included in the current model because the prototype does not contain a validated blood-pressure sensor and the development dataset does not provide synchronized arterial blood-pressure measurements. Consequently, a scientifically valid blood-pressure ablation could not be performed. Future participant-level studies will consider integrating a validated blood-pressure sensing interface or securely synchronized measurements from an external medical device, followed by quantitative feature-ablation analysis.

The next validation stage will build directly upon the present architecture through synchronized timestamped acquisition, participant and session identifiers, sensor-specific observation windows, stale-value rejection, missing-value handling, and participant- or session-separated evaluation. Temporal CNN, LSTM, or GRU models will become methodologically appropriate after genuine sequential sensor streams are available.

The embedded obstacle module will be extended through retained trial-level confusion matrices, direction-specific results, reference-distance errors, repeated frame-rate measurements, broader obstacle categories, dynamic obstacles, indoor walking routes, and outdoor illumination conditions. Older-adult testing will require institutional approval, informed consent, participant-level safety procedures, and accessibility and usability assessment.

The hybrid personalization pathway will be evaluated through user-specific training and held-out testing before and after external model refinement. Equation (9) provides the basis for quantifying whether personalization improves accuracy, calibration, critical-class recall, or false-alert performance.

These extensions build upon the current classification, embedded vision, sensing, response, GPS, display, and CSV-update capabilities. The present results therefore establish a technically integrated foundation for subsequent synchronized, longitudinal, and participant-level validation.

6. Conclusion and Future Work

cStick 2.0 presents a vision-enabled, IoMT-edge based smart walking-stick architecture that combines multimodal fall-risk classification, embedded obstacle awareness, physiological and motion-related sensing, stick-pressure monitoring, GPS acquisition, display output, buzzer response, and CSV-based sensor-record accumulation. The architecture further provides a structured pathway for external model refinement, wireless communication, application synchronization, multimodal accessibility support, and caregiver-oriented services.

The current evaluation establishes prototype-level evidence through complementary experimental pathways. Fall-risk classification was evaluated externally using a 9,670-record development dataset derived from 2,039 source records from the earlier cStick dataset and 7,631 cleaned, transformed, and augmented records. The deployment-oriented comparison showed that compact neural configurations achieved the strongest overall performance. TinyML INT8 achieved 95.76% accuracy, 93.91% balanced accuracy, and a macro F1-score of 94.92%, while the cStick compact DNN achieved 95.40% accuracy, 92.35% balanced accuracy, a macro F1-score of 93.90%, and a ROC-AUC of 97.44%.

A separate feature-ablation experiment using the complete six-input cStick DNN achieved 95.24% accuracy, 92.47% balanced accuracy, a macro F1-score of 93.87%, and a macro ROC-AUC of 97.46%. The critical/fall class achieved 86.73% recall, corresponding to a false-negative rate of 13.27%. The small difference between the deployment-oriented DNN result and the ablation reference arises from their separate evaluation pipelines and does not alter the overall interpretation.

The ablation results support the statistical complementarity of obstacle distance, stick pressure, HRV, externally supplied blood glucose, SpO_2, and continuous acceleration within the expanded tabular dataset. Obstacle distance was the strongest individual input, achieving 82.52% accuracy and a macro ROC-AUC of 92.84%; however, its critical-class recall was only 30.10%. Continuous acceleration, physiological variables, stick pressure, and blood glucose also produced substantially weaker and less balanced performance when evaluated independently.

Blood glucose was evaluated as externally supplied contextual health information rather than as an onboard or continuously synchronized sensor signal. The glucose-only configuration achieved 33.33% balanced accuracy, a macro F1-score of 24.59%, and zero recall for the critical/fall class. Removing glucose from the complete feature vector reduced the macro F1-score from 93.87% to 90.91% and increased the critical-class false-negative rate from 13.27% to 16.84%. These findings indicate an incremental statistical contribution within the current dataset but do not establish glucose as an independent, causal, or clinically validated fall predictor.

The Arduino Nicla Vision obstacle-awareness module was evaluated separately through Edge Impulse model validation and a controlled physical experiment. The implemented pathway combines INT8 Edge Impulse obstacle inference, centroid-based left-, center-, or right-direction assignment, onboard time-of-flight distance measurement, serial communication with the Arduino UNO WiFi Rev2, and buzzer and display response. The Edge Impulse validation stage produced a precision of 0.73, recall of 0.67, and F1-score of 0.70.

The controlled physical evaluation included 108 obstacle-present trials and 36 clear-path trials across a person, chair, and cardboard box positioned at 50, 100, and 150 cm. Left-, center-, and right-direction conditions, normal and dim lighting, and stationary- and moving-camera operation were evaluated across six sessions. A total of 108 of the 144 physical trials were classified correctly, resulting in an obstacle-presence accuracy of 75.00%. The embedded pipeline operated at approximately 19–20 FPS. These results demonstrate component-level embedded obstacle awareness under varied controlled conditions while retaining a clear distinction from future route-level and older-adult evaluation.

Complete trial-level confusion-matrix, direction-classification, and distance-estimation records were not retained for reproducible aggregate calculation; therefore, additional physical-test precision, recall, F1-score, false-positive rate, false-negative rate, direction accuracy, MAE, and RMSE values are not claimed.

The Arduino UNO WiFi Rev2 was functionally demonstrated for sensor acquisition, CSV record generation, GPS acquisition, display control, and buzzer activation. Externally generated DNN decision outputs were supplied to the controller response logic, enabling automatic buzzer activation and active display output for warning and critical/fall states. The temperature and humidity sensor remained an auxiliary circuit component and was not used in the DNN, obstacle module, or reported evaluation.

The adaptive capability of cStick 2.0 follows a hybrid implementation. New sensor records can be appended to the CSV repository through an implemented data-accumulation pathway. The expanded repository can subsequently support external retraining and user-specific model refinement. Quantitative personalization benefits, including changes in accuracy, calibration, critical-class recall, or false-alert frequency, require longitudinal user-specific before-and-after evaluation and are therefore not claimed in the current prototype.

The cStick 2.0 application was developed as a standalone interface prototype for manual record entry, statistical-view selection, and visualization of representative sensor and decision information. It was not connected to the walking-stick hardware or evaluated through usability, accessibility, caregiver, or task-completion studies. Hardware-to-application synchronization, automatic caregiver notification, email transmission, haptic feedback, spoken-audio output, and remote emergency communication remain structured extensions of the architecture.

The present fall-risk evaluation is based on independent tabular records rather than synchronized multimodal streams collected from older adults using the complete prototype. Participant identifiers, session identifiers, original acquisition timestamps, sensor sampling intervals, and physiological and motion-signal windows are unavailable. Consequently, the reported results demonstrate dataset-level statistical complementarity and prototype-level classification performance rather than a causal or spatiotemporal relationship with independently observed real-world falls.

Future validation will extend cStick 2.0 through synchronized participant-level data acquisition. Each record will include a participant identifier, session identifier, sensor source, acquisition timestamp, sampling interval, and validity period. Acceleration, gyroscope, stick-pressure, and obstacle-distance signals will be retained as temporal windows, while HRV and SpO_2 will be calculated over their corresponding physiological observation intervals. Externally supplied glucose values will be subjected to freshness validation before inclusion as contextual information.

Normal walking, turning, stopping, support-seeking behavior, stumbling, and ethically designed fall-related activities will be independently annotated. Participant- and session-separated training, validation, and testing will then be used to compare motion-only, physiological-only, environmental, and synchronized multimodal configurations. Temporal CNN, LSTM, and GRU models will be evaluated after genuine sequential sensor streams become available.

The obstacle-awareness module will be extended through retained trial-level confusion matrices, direction-specific accuracy, distance-estimation MAE and RMSE, repeated frame-rate measurements, broader obstacle categories, dynamic objects, indoor walking routes, and outdoor illumination conditions. End-to-end latency, GPS error, actuator-response time, wireless communication performance, memory utilization, power consumption, and battery life will also be evaluated.

Future human-subject studies will be conducted after institutional approval and informed consent. These studies will assess comfort, usability, warning comprehension, false-alert burden, accessibility, caregiver response, long-term reliability, and personalized model refinement among older adults. Secure IoMT deployment will additionally incorporate authenticated communication, encrypted transmission and storage, role-based caregiver access, protected key management, audit logging, and event-limited sharing of physiological and GPS information.

Overall, cStick 2.0 establishes a technically integrated foundation for combining multimodal fall-risk classification with embedded obstacle awareness and immediate assistive response. The current offline model results, controlled Nicla Vision evaluation, functional sensor integration, automatic buzzer and display response, GPS acquisition, and CSV-update capability provide a strong basis for the next stage of synchronized, secure, longitudinal, and participant-level assistive-mobility validation.

Author Contributions: Conceptualization, L.R., S.S.H.C., S.P.M., and E.K.; methodology, L.R. and S.S.H.C.; software, S.S.H.C.; validation, L.R. and S.S.H.C.; formal analysis, L.R. and S.S.H.C.; investigation, L.R. and S.S.H.C.; resources, S.P.M. and E.K.; writing, original draft preparation, L.R. and S.S.H.C.; writing, review and editing, L.R., S.S.H.C., S.P.M. and E.K.; supervision, S.P.M. and E.K.; project administration, S.P.M. and E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. This study did not involve new human-subject experiments; the model validation used the cited prototype-level dataset.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used for model development is available in the cited Kaggle repository. Additional prototype data are available from the corresponding author upon reasonable request.

Acknowledgments: Figures 1, 2, and 5 include AI-generated or AI-assisted visual content created using ChatGPT (OpenAI). ChatGPT was also used to assist with English grammar, language refinement, and manuscript readability. All scientific content, research concepts, system design, methodological decisions, algorithmic logic, intellectual property, and novel contributions were independently conceived and developed by the human authors. The authors reviewed, edited, and approved all AI-assisted content and take full responsibility for the accuracy, originality, and final content of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. James, S.L.; et al. The Global Burden of Falls: Global, Regional and National Estimates of Morbidity and Mortality from the Global Burden of Disease Study 2017. *Injury Prevention* **2020**, *26*, i3–i11. https://doi.org/10.1136/injuryprev-2019-043286.

2. Montero-Odasso, M.; et al. World Guidelines for Falls Prevention and Management for Older Adults: A Global Initiative. *Age and Ageing* **2022**, *51*, afac205. https://doi.org/10.1093/ageing/afac205.
3. O'Loughlin, J.L.; Robitaille, Y.; Boivin, J.F.; Suissa, S. Incidence of and Risk Factors for Falls and Injurious Falls Among the Community-Dwelling Elderly. *American Journal of Epidemiology* **1993**, *137*, 342–354. https://doi.org/10.1093/oxfordjournals.aje.a116681.
4. Alshammari, S.A.; Alhassan, A.M.; Aldawsari, M.A.; Bazuhair, F.O.; Alotaibi, F.K.; Aldakhil, A.A.; Abdulfattah, F.W. Falls Among Elderly and Its Relation with Their Health Problems and Surrounding Environmental Factors in Riyadh. *Journal of Family and Community Medicine* **2018**, *25*, 29–34. https://doi.org/10.4103/jfcm.JFCM_48_17.
5. Lin, F.R.; Ferrucci, L. Hearing Loss and Falls Among Older Adults in the United States. *Archives of Internal Medicine* **2012**, *172*, 369–371. https://doi.org/10.1001/archinternmed.2011.728.
6. Razmara, J.; Zaboli, M.H.; Hassankhani, H. Elderly Fall Risk Prediction Based on a Physiological Profile Approach Using Artificial Neural Networks. *Health Informatics Journal* **2018**, *24*, 410–418. https://doi.org/10.1177/1460458216677841.
7. Denkinger, M.D.; Lukas, A.; Nikolaus, T.; Hauer, K. Factors Associated with Fear of Falling and Associated Activity Restriction in Community-Dwelling Older Adults: A Systematic Review. *The American Journal of Geriatric Psychiatry* **2015**, *23*, 72–86. https://doi.org/10.1016/j.jagp.2014.03.002.
8. Siripurapu, I.D.; Rachakonda, L.; Mohanty, S.P.; Kougianos, E. iLog 2.2: Volume and Nutrition Estimation for Mixed Foods via Mask R-CNN and Federated Learning. *Electronics* **2026**, *15*. https://doi.org/10.3390/electronics15071460.
9. Islam, S.M.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K.S. The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access* **2015**, *3*, 678–708. https://doi.org/10.1109/ACCESS.2015.2437951.
10. Siripurapu, I.D.; Mitra, A.; Mohanty, S.P.; Kougianos, E. iLog 3.0: Estimating Food Volume from 2D Images Using Mask R-CNN and Monocular Depth Estimation. In Proceedings of the Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2025, pp. 1–6. https://doi.org/10.1109/ISVLSI65124.2025.11130217.
11. Hamdan, S.; Ayyash, M.; Almajali, S. Edge-Computing Architectures for Internet of Things Applications: A Survey. *Sensors* **2020**, *20*, 6441. https://doi.org/10.3390/s20226441.
12. Arduino. Arduino UNO WiFi Rev2 Hardware Documentation. Arduino Documentation.
13. Arduino. Arduino Nicla Vision Hardware Documentation. Arduino Documentation.
14. Rachakonda, L.; Mohanty, S.P.; Kougianos, E. cStick: A Calm Stick for Fall Prediction, Detection and Control in the IoMT Framework. In *Internet of Things: Technology and Applications*; Springer: Cham, 2022; Vol. 641, *IFIP Advances in Information and Communication Technology*, pp. 129–145. https://doi.org/10.1007/978-3-030-96466-5_9.
15. Rachakonda, L. Elderly Fall Prediction and Detection. Kaggle dataset.
16. Rachakonda, L.; Sharma, A.; Mohanty, S.P.; Kougianos, E. Good-Eye: A Combined Computer-Vision and Physiological-Sensor Based Device for Full-Proof Prediction and Detection of Fall of Adults. In *Internet of Things: A Confluence of Many Disciplines*; Springer: Cham, 2020; Vol. 574, *IFIP Advances in Information and Communication Technology*, pp. 273–288. https://doi.org/10.1007/978-3-030-43605-6_16.
17. Wu, F.; Zhao, H.; Zhao, Y.; Zhong, H. Development of a Wearable-Sensor-Based Fall Detection System. *International Journal of Telemedicine and Applications* **2015**, *2015*, 576364. https://doi.org/10.1155/2015/576364.
18. Subramaniam, S.; Faisal, A.I.; Deen, M.J. Wearable Sensor Systems for Fall Risk Assessment: A Review. *Frontiers in Digital Health* **2022**, *4*, 921506. https://doi.org/10.3389/fdgth.2022.921506.
19. Usmani, S.; Saboor, A.; Haris, M.; Khan, M.A.; Park, H. Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. *Sensors* **2021**, *21*, 5134. https://doi.org/10.3390/s21155134.
20. Pech, M.; Sauzéon, H.; Yebda, T.; Benois-Pineau, J.; Amieva, H. Falls Detection and Prevention Systems in Home Care for Older Adults: Myth or Reality? *JMIR Aging* **2021**, *4*, e29744. https://doi.org/10.2196/29744.

21. Hu, X.; Duan, Q.; Tang, J.; Chen, G.; Zhao, Z.; Sun, Z.; Chen, C.; Qu, X. A Low-Cost Instrumented Shoe System for Gait Phase Detection Based on Foot Plantar Pressure Data. *IEEE Journal of Translational Engineering in Health and Medicine* **2023**, *12*, 84–96. https://doi.org/10.1109/JTEHM.2023.3319576.
22. Nho, Y.H.; Lim, J.G.; Kwon, D.S. Cluster-Analysis-Based User-Adaptive Fall Detection Using Fusion of Heart Rate Sensor and Accelerometer in a Wearable Device. *IEEE Access* **2020**, *8*, 40389–40401. https://doi.org/10.1109/ACCESS.2020.2969453.
23. Ngu, A.H.; Metsis, V.; Coyne, S.; Chung, B.; Pai, R.; Chang, J. Personalized Fall Detection System. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops. IEEE, 2020, pp. 1–7. https://doi.org/10.1109/PerComWorkshops48775.2020.9156172.
24. Ikram, S.; Bajwa, I.S.; Ikram, A.; de la Torre Díez, I.; Ríos, C.E.U.; Castilla, Á.K. Obstacle Detection and Warning System for Visually Impaired Using IoT Sensors. *IEEE Access* **2025**, *13*, 35309–35321. https://doi.org/10.1109/ACCESS.2025.3543299.
25. Farooq, M.S.; Shafi, I.; Khan, H.; De La Torre Díez, I.; Breñosa, J.; Martínez Espinosa, J.C.; Ashraf, I. IoT Enabled Intelligent Stick for Visually Impaired People for Obstacle Recognition. *Sensors* **2022**, *22*, 8914. https://doi.org/10.3390/s22228914.
26. Mai, C.; Chen, H.; Zeng, L.; Li, Z.; Liu, G.; Qiao, Z.; Qu, Y.; Li, L.; Li, L. A Smart Cane Based on 2D LiDAR and RGB-D Camera Sensor-Realizing Navigation and Obstacle Recognition. *Sensors* **2024**, *24*, 870. https://doi.org/10.3390/s24030870.
27. Apple Inc.. Apple Watch. https://support.apple.com/en-us/HT208944. [Accessed: Apr. 21, 2025].
28. Samsung. Use the Detect Fall Feature on Your Samsung Smart Watch. https://www.samsung.com/us/support/answer/ANS10003423/. [Accessed: Jan. 11, 2026].
29. Google. Get Help in an Emergency with Google Pixel Watch Safety Features. https://support.google.com/googlepixelwatch/answer/12663810. [Accessed: Dec. 08, 2025].
30. Garmin. Incident Detection and Assistance Requirements. https://support.garmin.com/en-US/?faq=hNLok26dmQ6Lntcw66lfX7. [Accessed: Jan. 12, 2026].
31. ADT. Medical Alert Systems. https://www.adt.com/health. [Accessed: Jan. 11, 2026].
32. Bay Alarm Medical. Automatic Fall Detection. https://www.bayalarmmedical.com/medical-alert-system/automatic-fall-detection/. [Accessed: Feb. 09, 2026].
33. Lively. Lively Mobile2 All-in-One Medical Alert. https://shop.lively.com/products/lively-mobile2-all-in-one-medical-alert. [Accessed: Jan. 11, 2026].
34. MobileHelp. Fall Button: A Fall Alert System with Automatic Fall Detection. https://www.mobilehelp.com/pages/fall-button. [Accessed: Jan. 11, 2026].
35. Lifeline. Medical Alert Systems for Home, Mobile, and Smartwatch Support. https://www.lifeline.com/. [Accessed: Feb. 11, 2026].
36. Medical Alert. Automatic Fall Detection. https://www.medicalalert.com/fall-detection/. [Accessed: Jan. 11, 2026].
37. UnaliWear. The Kanega Medical Alert Watch with Fall Detection. https://www.unaliwear.com/product/kanega-watch/. [Accessed: Jan. 11, 2026].