

Review

A Survey of LSTM Pedestrian Intention Prediction and Lightweight Methods for Intelligent Guide Sticks

Yijia Cai ¹, Fang Jing ^{1,*}, Huafeng Qu ^{1,*} , Yuxi Xie ¹  and Shafrida Sahrani ² 

¹ School of Information and Intelligent Engineering, Yunnan College of Business Management, Kunming 650300, China; caiyijia@stu.ynjgy.edu.cn (Y.C.); yuxi.xie.14@alumni.ucl.ac.uk (Y.X.)

² Institute of Visual Informatics, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; shafrida@ukm.edu.my

* Correspondence: 20192145@ynjgy.edu.cn (F.J.); p132874@siswa.ukm.edu.my (H.Q.)

Abstract

The travel problem of visually impaired people is a worldwide issue that needs urgent attention. Although intelligent guide sticks provide obstacle detection and early warning through multi-sensor fusion and embedded algorithms, existing systems generally cannot model the temporal movement patterns of dynamic obstacles, such as pedestrians and vehicles, thereby hindering intention prediction and active obstacle avoidance. Long short-term memory (LSTM), with its gating mechanism, effectively captures long-term dependencies in trajectories and offers a promising solution. This review compares and analyzes LSTM against other mainstream temporal models under the resource constraints of intelligent guide sticks and finds that LSTM demonstrates a favorable combination in temporal modeling capability, lightweight maturity, and edge deployment feasibility. We categorize five lightweight techniques—architecture simplification, low-rank decomposition, structured pruning, quantization, and knowledge distillation—and examine their compression effectiveness, accuracy preservation, and hardware applicability across typical platforms. Furthermore, this review surveys application cases in speech guidance, trajectory prediction-based obstacle avoidance, positioning and navigation, edge computing, and Internet collaboration, exploring the diverse potential of LSTM in intelligent guide stick scenarios. The findings indicate that, after lightweight processing, LSTM models can meet the deployment requirements of resource-constrained edge devices, suggesting their potential feasibility on resource-constrained hardware platforms. However, existing applications still face challenges in balancing real-time performance and accuracy, meeting stringent resource constraints, and the absence of end-to-end validation on real intelligent guide stick prototypes. The reviewed evidence suggests that LSTM-based prediction represents a promising and practically valuable pathway for transitioning intelligent guide sticks from passive response to active prediction. Future research should prioritize real-world deployment validation, domain-specific data collection, and hardware-software co-design to realize its potential fully.

Keywords: intelligent guide stick; LSTM; pedestrian intention prediction; lightweight; edge computing



Academic Editors: Mohamed Lahby and Salim El Khediri

Received: 8 June 2026

Revised: 11 July 2026

Accepted: 13 July 2026

Published: 15 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The travel problem of visually impaired people is a worldwide issue that needs urgent attention. The World Health Organization estimates that at least 2.2 billion people

worldwide have near-vision or far-vision impairment, of whom at least 1 billion cases could have been prevented or remain untreated [1]. With population aging and changing lifestyles, the number of visually impaired people continues to grow, making travel difficulties increasingly prominent. Traditional white canes, while capable of detecting ground-level obstacles through physical contact, cannot detect hanging obstacles at waist or head height, and require systematic orientation and mobility training—which some visually impaired individuals forgo due to the high training threshold or psychological resistance. Intelligent guide sticks, which achieve obstacle detection and early warning through multi-sensor fusion and embedded algorithms, have become an important research direction in assistive mobility. However, existing typical systems—such as those based on LiDAR and RGB-D cameras [2], YOLO-Fastest [3], SenseStride [4], and more recently emerging edge-cloud collaborative platforms [5–8]—although capable of target recognition and distance warning, cannot generally model the temporal movement patterns of dynamic obstacles such as pedestrians and vehicles, making intention prediction and active obstacle avoidance difficult.

In this context, Long Short-Term Memory (LSTM) networks [9] offer a viable technical pathway to address these limitations. Through the cooperative control of the forget gate, input gate, and output gate, LSTM effectively alleviates the vanishing and exploding gradient problems of traditional recurrent networks and can stably capture long-term dependencies in the trajectories of dynamic obstacles. This characteristic gives LSTM significant advantages in trajectory prediction and intention recognition tasks, and provides a theoretical foundation for its introduction into intelligent guide stick systems. In recent years, LSTM has been extensively validated in pedestrian-dense scenarios and autonomous driving applications; however, its suitability for deployment in intelligent guide sticks—a resource-constrained, unstable-field-of-view scenario—remains to be evaluated.

Although independent reviews exist on intelligent guide stick hardware, general deep learning compression, and pedestrian trajectory prediction, studies that integrate these three domains from a task–hardware–algorithm co-design perspective remain scarce. Specifically: (1) existing intelligent guide stick-oriented reviews primarily focus on sensor configuration and feedback modalities, without thoroughly analyzing the deployment suitability of temporal prediction models on embedded platforms; (2) existing lightweight surveys largely concentrate on general model compression techniques or optimization schemes for CNNs/Transformers, and provide insufficient classification for the gated structure of LSTMs; and (3) existing trajectory prediction studies are mostly based on autonomous driving or surveillance scenarios, overlooking the unique constraints of sensors mounted on intelligent guide sticks, such as low mounting height, unstable field of view, and warning horizons of less than 2 s.

To address these gaps, this review aims to evaluate the applicability of LSTM in intelligent guide stick scenarios, synthesize lightweight deployment methods, and identify the critical gaps between algorithmic validation and engineering implementation. Specifically, we first construct an applicability analysis framework that compares LSTM against other mainstream temporal models under the resource constraints of intelligent guide sticks. Second, we propose a hierarchical taxonomy of lightweight methods tailored to the LSTM gated structure, and analyze the applicability and trade-offs of each technique across typical hardware platforms. Third, we find that although numerous studies have validated LSTM-based prediction in robotic or simulation environments, and preliminary explorations of Internet-integrated assistive systems have emerged, end-to-end deployment on actual intelligent guide stick prototypes with system-level validation involving visually impaired users has yet to be achieved, thereby drawing a clear boundary between algorithmic potential and engineering reality. This review provides a framework for researchers

and engineers transitioning from passive obstacle detection to active intention prediction in assistive navigation devices.

To guide this review, the following research questions were formulated:

RQ1: What are the mainstream LSTM-based methods for pedestrian trajectory and intention prediction, and what are their performance characteristics and applicable scenarios?

RQ2: What lightweight techniques are currently available for deploying LSTM models on resource-constrained edge platforms, and what are their compression performance, accuracy preservation, and hardware applicability?

RQ3: What are the key technical bottlenecks in integrating LSTM-based trajectory and intention prediction capabilities into intelligent guide stick systems, and how can they be addressed in future research?

This review focuses on deploying LSTM-based pedestrian intention prediction on MCU-level embedded platforms, with particular emphasis on lightweight techniques for STM32-series microcontrollers. Such platforms offer a favorable balance between power consumption and performance required for wearable assistive devices (<500 mW, >8 h battery life). While FPGA-based approaches are also discussed where relevant, the core analysis centers on MCU-level deployment as the main thread.

Section 2 first describes the scope and selection criteria of the literature search for this review, followed by an overview of intelligent guide stick systems and their existing limitations in Section 3. Sections 4–6 then address the three research questions respectively, with discussion and conclusions presented in Sections 7 and 8.

2. Methods

This review aims to survey the technical landscape and research progress of LSTM-based pedestrian intention prediction and lightweight deployment for intelligent guide sticks. To ensure transparency and reproducibility of the literature coverage, this section defines the database sources, time window, search terms, and inclusion principles.

Databases searched include IEEE Xplore, Web of Science, Scopus, PubMed, and China National Knowledge Infrastructure (CNKI), covering both international frontier research and domestic engineering-oriented contributions.

The search period spans from January 1997 to June 2026. The starting year is set to 1997, which corresponds to the publication of the foundational LSTM paper by Hochreiter and Schmidhuber [9]. This seminal work established the theoretical basis for all recurrent temporal prediction models discussed in this review, making it the most appropriate starting point for tracing the evolution of LSTM-based methods. Furthermore, the representative pioneering work on pedestrian trajectory prediction, Social LSTM [10], was published in 2016, and falls well within this retrieval interval. The end date covers all publicly available publications up to the manuscript submission date in June 2026.

Search keyword combinations included English terms such as “LSTM”, “long short-term memory”, “pedestrian trajectory”, “intention prediction”, “guide stick”, and “smart cane”.

Literature inclusion follows these principles: (1) studies involving LSTM in pedestrian trajectory or intention prediction; (2) studies addressing model compression, quantization, pruning, or other lightweight techniques, as well as embedded or edge deployment; (3) studies related to assistive navigation devices, intelligent guide sticks, or visually impaired assistance systems. Preference is given to peer-reviewed journal articles and conference papers, while representative preprints and technical reports are also considered where they provide significant technical contributions.

It should be noted that this article is structured as a narrative review, aiming to organize the technical evolution, mainstream approaches, and engineering challenges in a coherent manner, rather than following the PRISMA guidelines for a systematic review.

3. Overview of Intelligent Guide Stick System

Intelligent guide sticks achieve environmental perception and interactive feedback through multi-sensor fusion and embedded algorithms, serving as important assistive devices for the travel of visually impaired individuals. Existing systems generally adopt a “perception-decision-feedback” three-tier hardware architecture and have made certain progress in obstacle detection and distance warning, yet remain significantly limited in intention prediction and active obstacle avoidance for dynamic obstacles. This section first introduces the composition and functions of typical hardware architectures, then outlines the key functional requirements and technical specifications, and finally provides a comparative analysis of three representative systems.

3.1. Typical Hardware Architecture

The sensing layer integrates multiple complementary sensors for comprehensive environmental perception. Ultrasonic sensors provide basic ranging from 2 cm to 450 cm with centimeter-level accuracy, typically arranged in a multi-probe layout to eliminate blind spots [3]. Time-of-Flight (ToF) sensors offer higher precision (± 5 mm) for mid-range detection up to 4 m, making them particularly suitable for detecting hanging obstacles at head height [4]. 2D LiDAR enables 360° omnidirectional scanning for environmental contour mapping and SLAM-based localization, though at higher cost and power consumption. RGB-D cameras simultaneously capture color images and depth information, supporting 86-class object recognition including pedestrians, vehicles, and traffic lights [2]. However, the use of such visual sensors in public spaces inevitably captures bystander information, raising privacy concerns that should be addressed through on-device processing and data minimization. Inertial Measurement Units (IMUs) monitor the stick’s attitude and the user’s motion state through three-axis accelerometers, gyroscopes, and magnetometers, enabling fall detection and assisting positioning during GPS loss [2]. This multi-sensor configuration effectively eliminates the detection blind spots inherent to single-sensor solutions. Figure 1 illustrates the overall architecture of the environmental sensing and time-series preprocessing module, including the data acquisition, multi-sensor fusion, and the generation of temporal input sequences for subsequent prediction.

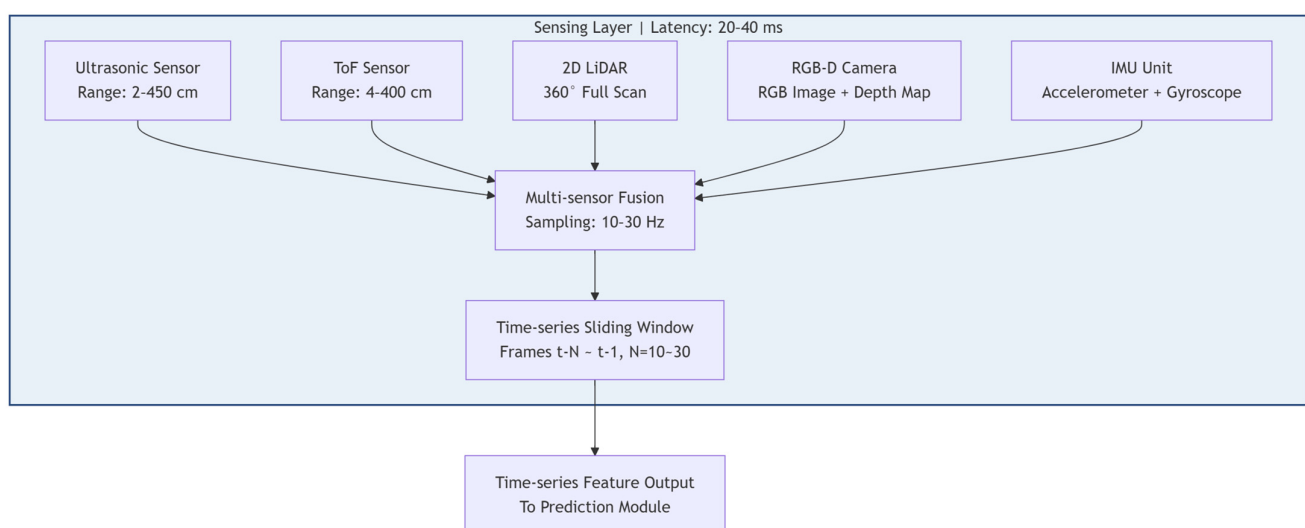


Figure 1. Architecture of Environmental Sensing and Timeries Preprocessing Module.

The decision layer serves as the computational core, responsible for sensor data fusion and algorithm execution. Embedded platforms are selected based on the trade-off between computing power and power consumption. High-performance platforms such as the NVIDIA Jetson Nano deliver up to 472 GFLOPS at 5–10 W, enabling end-to-end deep learning models like YOLOv5 for target detection and tracking [2], though their power demands limit endurance. Mid-range platforms like Raspberry Pi 4 operate at 3–5 W, suitable for lightweight models such as YOLO-Fastest with inference delays within 120 ms per frame [3]. Ultra-low-power platforms such as STM32 series microcontrollers consume only 0.1–0.5 W, running RTOS-based simple algorithms like distance thresholding with response delays as low as 50 ms, making them suitable for battery-powered wearable devices [4]. Field-Programmable Gate Arrays offer nanosecond-level precision timing and hardware-level image processing acceleration, with demonstrated response times at the millisecond level for obstacle detection [11,12]. Figure 2 depicts the internal structure of the lightweight LSTM prediction and risk grading module within the decision layer, illustrating the complete computational pipeline from temporal feature extraction to risk-level output.

The feedback layer conveys processed environmental information to the user through multimodal interaction. Voice feedback via bone-conduction headphones or small speakers delivers complex information such as obstacle type, distance, and direction, supporting offline wake-up and context-aware dialog. Vibration feedback through eccentric rotating mass motors provides tactile signals that do not occupy auditory channels, allowing users to simultaneously perceive environmental sounds. Vibration frequency and intensity are graded according to obstacle distance: safe distance (>150 cm) triggers no feedback; caution distance (80–150 cm) produces low-frequency vibration; warning distance (30–80 cm) produces medium-frequency vibration; and danger distance (<30 cm) triggers high-frequency vibration. Multi-directional tactile feedback is achieved by arranging vibration motors at different handle positions to indicate obstacle direction. In practice, voice and vibration are used cooperatively—vibration for routine navigation to maintain environmental awareness, and voice for complex or hazardous situations requiring detailed description [4]. Figure 3 presents the overall architecture of multi-modal human–machine feedback and closed-loop iteration, covering the cooperative working mechanisms of voice, vibration, and other feedback modalities.

3.2. Functional Requirements and Technical Indicators

The functional requirements of the intelligent guide stick span multiple dimensions, including obstacle detection, target recognition, real-time warning, positioning and navigation, endurance, and hardware resource constraints. Table 1 summarizes the above functional requirements, specific technical indicators, and their constraint significance for LSTM deployment, providing the basic boundary conditions for subsequent model selection and lightweight design.

In terms of obstacle detection, the system needs to detect obstacles in front, above, and on the ground in real time. Typical schemes use ultrasonic or ToF sensors. For example, the HC-SR04 ultrasonic sensor (Shenzhen Enfu Electronic Technology Co., Ltd. Shenzhen, China) has a detection range of 2 cm to 400 cm, a resolution of up to 3 mm, and a recommended measurement cycle of over 60 ms between trigger events [3]. The detection period should be kept within the sensor's recommended cycle to meet real-time requirements, and the sensor's recommended measurement cycle of 60 ms corresponds to approximately 16 Hz [13]. Some schemes also integrate infrared sensors for close-range blind compensation and inertial measurement units for stick posture monitoring and fall detection, thereby expanding the range of sensing data.

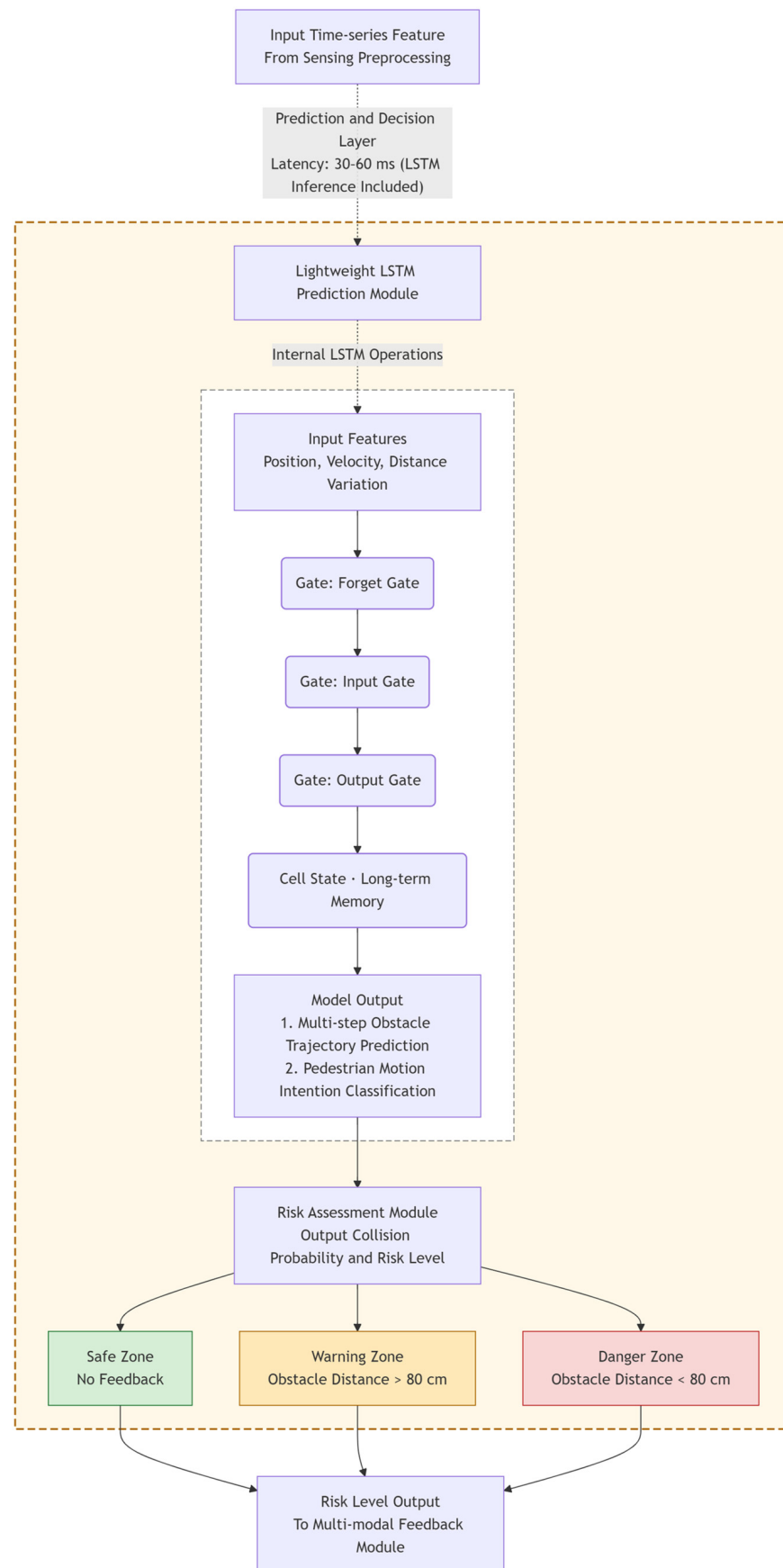


Figure 2. Architecture of Lightweight LSTM Prediction and Risk Grading Module.

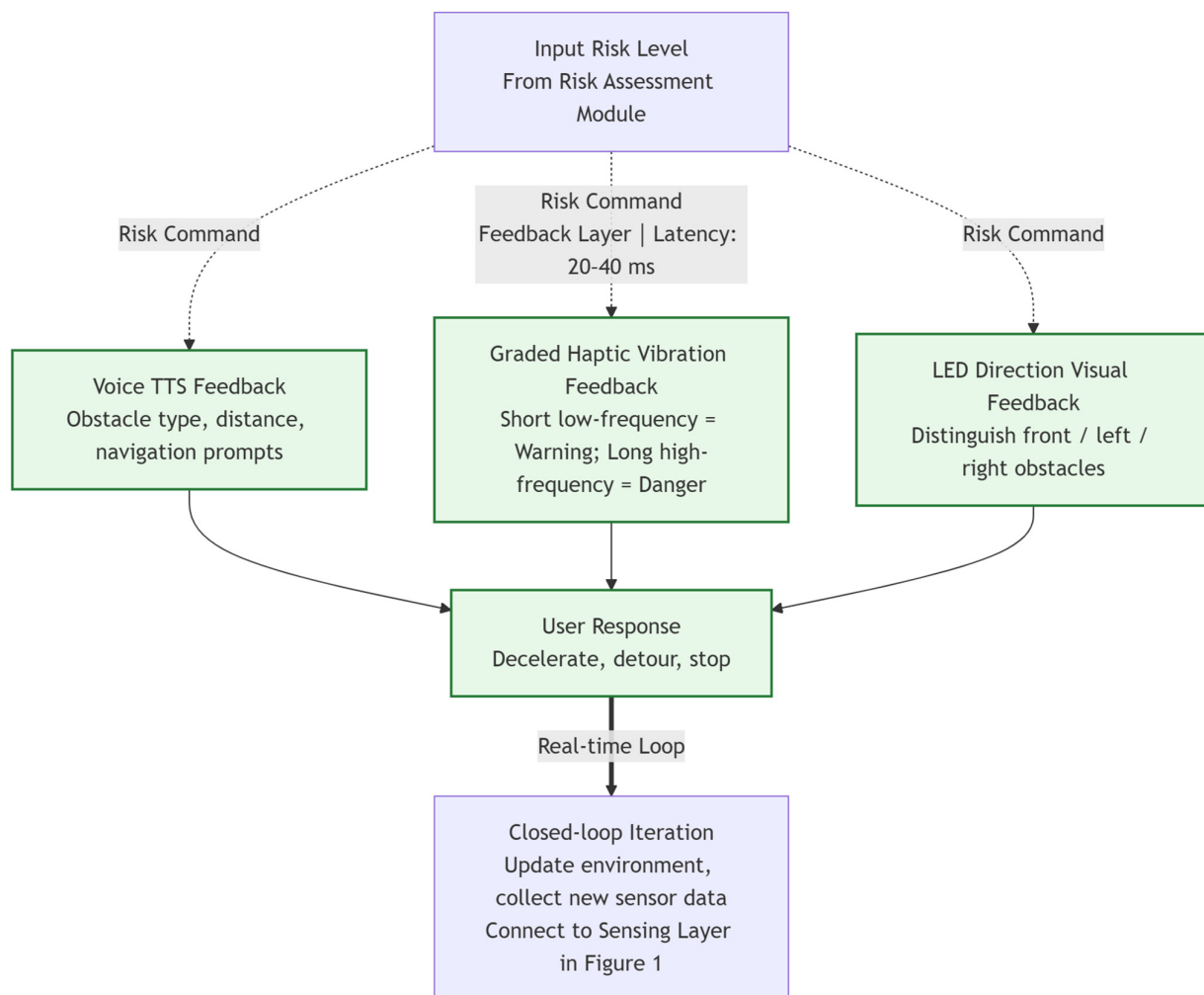


Figure 3. Architecture of Multi-modal Human–Machine Feedback and Closed-loop Iteration.

For pedestrian and vehicle recognition, the system needs to distinguish between obstacle types to assist decision-making. The intelligent guide stick, based on the improved YOLOv5 algorithm [2,14], can recognize 86 object types in the benchmark dataset [2]. The recognition rate of the crosswalk is 84.6%, the vehicle recognition rate is 71.8%, and the system processing frame rate is maintained at 25–26 fps [2]. The YOLO-Fastest-based YF-SLAM system achieves 10.12 ms per frame for segmentation and tracking on embedded platforms [15]. In recent years, researchers have further explored the fusion strategy of the ReLU and SiLU activation functions, combined with an improved coupling head detection module and the SloU bounding box loss function, to improve detection accuracy and model robustness for small obstacles in complex backgrounds [2].

In terms of real-time early-warning feedback, the total end-to-end delay of the system is usually required to be kept between 50 ms and 150 ms. The typical tactile feedback scheme adopts the vibration motor array, and sets different vibration pulse durations according to the distance of the obstacle: the long distance is the safe range, and the system has no feedback; middle distance produces a short vibration pulse to prompt attention; produces a medium vibration pulse for early warning at close range; and dangerous distance triggers a long vibration pulse as alarm [4]. In some commercial products, such as the WeWALK Smart Cane 2 [16], obstacle detection sensors capable of detecting obstacles above waist level are added to detect suspended obstacles, such as branches and lintels, and to warn users via audio or tactile vibration feedback. In addition, for example, the AI guide device of commercial products such as Glide [17] uses a stereo depth camera

combined with voice and vibration prompts to achieve a lead function similar to a ‘guide dog’ (i.e., the device provides active directional guidance through wheel control, not an actual canine) through active wheel control. It is important to note that the performance claims for these commercial products are based on manufacturer documentation and marketing materials, and have not been independently verified in peer-reviewed academic studies. These feedback methods do not occupy the auditory channel, allowing users to perceive environmental sounds simultaneously and achieve response delays as low as 50 ms.

In terms of positioning and navigation, the system usually integrates GPS modules for outdoor positioning, and the horizontal positioning accuracy must be better than 5 m [2]. The typical scheme uses two-in-one modules, such as the SIM808, to support A-GPS-assisted positioning and shorten the first positioning time. For indoor and other GPS-signal-missing scenarios, 2D LiDAR combined with the Cartographer algorithm [18] can be used to perform SLAM mapping and positioning. The mapping accuracy can reach $1\text{ m} \pm 7\text{ cm}$, and the SLAM mapping speed is maintained at 25–31 fps [2]. The scheme achieves autonomous guidance through omnidirectional wheel drive, enabling the intelligent guide stick to navigate actively.

In terms of endurance and hardware resources, the system needs to support continuous use for more than 8 h. A low-power platform such as an STM32 series microcontroller can have power consumption controlled to 0.1–0.5 W; standby power consumption can be as low as 0.3 μA [19], with the FreeRTOS real-time operating system enabling multi-task scheduling and power optimization. High-power platforms such as Jetson Nano have a power consumption of 10–15 W [2] is suitable for applications that require end-to-end deep learning. For low-power MCU-based solutions such as the STM32L series, Flash resources are usually limited to 512 KB, and SRAM to 256 KB [19]. By optimizing the handle design and the layout of electronic components, the WeWALK Smart Cane 2 (Wewalk Teknoloji A.S., İstanbul, Türkiye) achieves a weight close to that of a traditional white cane, making it easy to carry and use for extended periods.

In terms of weight and portability, the system’s overall weight should be kept below 400 g [4], which is close to that of the traditional white stick. This limitation indirectly constrains battery capacity and heat-dissipation design, which in turn affects processor selection. In terms of environmental adaptability, the system needs to support various scenarios, such as day and night and indoor and outdoor settings, and the visual sensor needs to be adaptable to low illumination.

Table 1. The functional requirements and technical indicators of intelligent guide stick and its constraint significance for LSTM deployment.

Functional Classification	Concrete Demand	Technology Index	Constraint Significance for LSTM Deployment
Obstacle detection	Real-time detection of front, upper and ground obstacles	The ultrasonic/ToF detection range is 0.02–4.5 m [3]	Perceptual data provides time series input for LSTM and requires data frame rate $\geq 10\text{ Hz}$.
Pedestrian/vehicle identification	Distinguish the types of obstacles (people, cars, traffic lights, etc.)	Recognition rate approximately 61% [2]; detection period approximately 10.12 ms [15]	LSTM needs to be combined with CNN or accept the test result sequence for intention prediction.

Table 1. Cont.

Functional Classification	Concrete Demand	Technology Index	Constraint Significance for LSTM Deployment
Intention anticipation	Predict pedestrian crossing, stopping and other behaviors	High accuracy intention classification	LSTM needs to complete a single inference in tens of milliseconds, and the model needs to be lightweight.
Real-time early warning feedback	Voice/Vibration grading early warning	The total end-to-end delay is less than 120 ms [3], and the typical system is 50–150 ms [3,4]	LSTM inference delay needs to meet the real-time requirements of the system.
Positioning and navigation	GPS/Beidou Positioning, Path Planning	Horizontal positioning accuracy ≤ 5 m (outdoor), auxiliary inertial navigation [2]	LSTM can be used for position prediction and trajectory smoothing, but it is not a real-time core.
Endurance function	Continuous use of more than 8 h	The power consumption of the whole machine needs to meet the requirements of continuous 8 h endurance [4]	LSTM inference power consumption needs to be optimized, and gap wake-up or quantitative inference can be used.
Hardware resource	Run on MCU or low-end FPGA	Flash ≤ 512 KB, SRAM ≤ 256 KB (typical STM32 level) [4]	The number of LSTM weight parameters needs to be compressed to hundreds of KB (e.g., 3.4 k parameters after Slim LSTM compression [20]).
Weight and portability	Close to the traditional white stick	Overall weight ≤ 400 g, usually around 300 g [4]	Restrict battery capacity and heat dissipation, and indirectly restrict the selection of computer chips.
Acclimatization	Day/Night, Indoor/Outdoor	The camera should support low illumination (such as infrared light), and the sensor should resist environmental interference	LSTM needs to be robust to sensor noise and may require multi-modal fusion.

3.3. Existing System Analysis

To comprehensively understand the current technical status and deficiencies of intelligent guide sticks, this section selects four representative systems for comparative analysis. Table 2 provides an intuitive comparison of these four systems across multiple dimensions, including core sensors, main algorithms, key performance metrics, Internet technology integration, and temporal modeling capabilities.

1. Intelligent guide stick based on 2D LiDAR and RGBD camera

This system was proposed by Mai et al. [2], utilizing 2D LiDAR and an RGB-D camera as core sensors, with a Jetson Nano B01 and STM32 dual-processor architecture. The system employs the Cartographer algorithm [18] for laser SLAM mapping and localization, achieving mapping accuracy of $1 \text{ m} \pm 7 \text{ cm}$ at a speed of 25–31 fps, enabling short-range obstacle avoidance and navigation in both indoor and outdoor environments. For target recognition, the system uses an improved YOLOv5 algorithm [2], recognizing 86 object categories on benchmark datasets [2], with a crosswalk recognition rate of 84.6%, vehicle recognition rate of 71.8%, overall recognition rate of 61.2%, and a processing frame rate maintained at 25–26 fps. The most distinctive feature of this system is its autonomous guidance capability

through an omnidirectional wheel drive at the base, endowing the intelligent guide stick with active guidance functionality similar to a “guide dog” (i.e., the stick itself provides active directional steering, rather than an actual canine companion), effectively guiding visually impaired users to avoid obstacles and reach their intended destinations.

2. Predictive intelligent guide stick based on YOLO-Fastest

This study [3], from California State University, aims to address the limitation of traditional intelligent guide stick devices that can only provide reactive obstacle detection. The system uses a Raspberry Pi 4 as the computing platform, integrating ultrasonic sensors and an Arducam OV5647 wide-angle camera, (Nanjing Arducam Electronics Technology Co., Ltd., Nanjing, China) running the lightweight YOLO-Fastest [15] deep learning model to enable real-time obstacle detection and tracking. Unlike the previous scheme, this system predicts short-term obstacle trajectories by analyzing positional changes in moving objects between consecutive video frames, thereby achieving the leap from “passive response” to “active warning.” Experimental results show that the system achieves high spatial localization accuracy in static obstacle scenarios, with an average Euclidean error of only 2.05–5.17 pixels; however, in dynamic challenge scenarios such as fast-moving obstacles and sudden obstacles, prediction performance degrades significantly, with average Euclidean errors of 70.37–72.06 pixels and 64.85–65.41 pixels, respectively [3]. The system’s end-to-end latency is approximately 120 milliseconds per frame [3], primarily presented through visual warnings such as directional arrows and trajectory markers. These visual cues are used only for test verification and are not directly intended for visually impaired users.

3. Sense Stride guide stick

Sense Stride is an intelligent guide stick system designed by Megharsh et al. [4] based on the STM32H563ZI microcontroller and ToF sensors (STMicroelectronics, Geneva, Switzerland). The system employs multiple VL53L0X ToF sensors for multi-directional obstacle distance detection, combined with a PIR sensor to detect moving humans, providing feedback to users through tactile vibration. Its main advantages are low power consumption and low latency: the STM32H5 series consumes approximately 0.1–0.5 W, with response time controllable within 50 milliseconds. However, the system’s functionality is relatively simple, relying solely on distance threshold detection, lacking target recognition capability and the ability to predict dynamic movement trends of obstacles.

4. Edge Computing and Cloud Collaborative Intelligent Guide Stick

Gao et al. [21] proposed a vision-based intelligent guide stick system that uses an STM32F103 as the sensor data acquisition controller to collect multi-modal data from ultrasonic, GPS, and inertial sensors, transmitting the data via serial communication to a Jetson Xavier NX edge computing platform, where a YOLO model is deployed for visual perception and obstacle recognition, achieving an average detection speed of 20.61 FPS. This system realizes the decoupling and collaboration between low-level sensor acquisition and high-level deep learning inference, balancing real-time performance with power consumption and computational capability. Building on this foundation, an AIoT-based intelligent guide stick system [5] uses an STM32F407 as the main controller, integrating a K230 camera and BeiDou positioning module, achieving 94.2% recognition accuracy for traffic lights and 92.8% for crosswalks through an improved YOLOv5 algorithm, while uploading perception results to the cloud for remote monitoring. Veerasamy et al. [6] developed a low-cost intelligent assistive cane based on an edge-cloud collaborative platform, with a Raspberry Pi as the terminal device, implementing functions including overhead obstacle recognition, fall detection, and traffic signal recognition. Furthermore, a V2X-based solution [7] connects the intelligent guide stick with transportation infrastructure to

obtain traffic light phase information for beyond-line-of-sight warning; Shireesha et al. [8] proposed a cloud-integrated smart cane solution that captures images via ESP32-CAM and uploads them to the Firebase cloud platform for object recognition, achieving 94.7% average recognition accuracy with 310 ms cloud inference latency within an IoT framework. However, the core algorithms of the above systems remain primarily CNN-based, without incorporating LSTM or other temporal prediction models. This means these systems can recognize current obstacles but cannot predict their future trajectories—the technological leap from “perceiving the present” to “predicting the future” remains a critical gap to be filled.

The comparative analysis of the four intelligent guide stick systems reveals several common limitations across current designs. Several review studies have indicated that existing electronic travel assistance devices generally face challenges such as limited processor performance and insufficient environmental perception, and that visually impaired users continue to experience considerable difficulties in dynamic obstacle avoidance and complex scene understanding [22,23]. From a technological evolution perspective, the first three systems focus respectively on current-frame perception, reactive warning, and local decision-making, while the fourth category, despite incorporating edge computing and cloud collaboration capabilities, still operates at the CNN-based current-frame recognition level. Notably, none of the systems reviewed have incorporated temporal motion modeling of dynamic targets into their decision-making pipelines, which limits their ability to anticipate short-term trajectory changes in pedestrians and vehicles. This observation suggests that introducing temporal modeling capabilities, such as LSTM, into intelligent guide stick architectures may offer a promising direction for transitioning from reactive warning to proactive assistance. However, the practical feasibility and effectiveness of such integration still require further investigation through systematic deployment and validation on actual intelligent guide stick prototypes.

Table 2. Comparison of four typical intelligent guide stick systems.

System	Based on 2D LiDAR + RGB-D Camera	Predictive Intelligent Guide Stick Based on YOLO-Fastest	Sense Stride	Edge Computing and Cloud Collaboration
Core sensor and main control	2D LiDAR, RGB-D camera; Jetson Nano + STM32 [2]	Monocular camera, Ultrasonic; Raspberry Pi 4 [3]	ToF sensor (VL53L1X), PIR; STM32H563ZI [4]	Camera, ultrasonic, GPS, IMU; STM32F103 + Jetson Xavier NX [21]
Core algorithm/function	SLAM mapping [18] YOLOv5 target recognition (86 categories) [2,24]	YOLO-Fastest target detection and trajectory prediction [3,15]	Multi-direction obstacle threshold detection [4]	YOLO object detection, cloud data upload, V2X communication [5–8]
Key performance	The accuracy of SLAM is $1\text{ m} \pm 7\text{ cm}$ and the recognition rate of pedestrian crosswalk is 84.6% [2].	The delay is about 120 ms/frame; dynamic obstacle prediction error 70–72 pixels [3]	Response delay 50 ms; the accuracy of obstacle detection is 95% [4]	Detection speed: 20.61 FPS [21]; traffic light recognition: 94.2%, crosswalk recognition: 92.8% [5]
Feedback	Omni-directional wheel autonomous guidance [2]	Visual warning (for testing) [3]	Five-direction tactile vibration [4]	Voice feedback, remote monitoring [5,21]
Power consumption and computing power requirements	High (Jetson Nano Edge calculation, about 5–10 W) [2]	Medium (Raspberry Pi 4, approx. 3–5 W) [3]	Low (STM32H5 series, approx. 0.1–0.5 W) [4]	Medium-High (Jetson Xavier NX, ~10–15 W) [21]

Table 2. Cont.

System	Based on 2D LiDAR + RGB-D Camera	Predictive Intelligent Guide Stick Based on YOLO-Fastest	Sense Stride	Edge Computing and Cloud Collaboration
Internet Technology Integration	No explicit integration, local inference [2]	No cloud collaboration, local processing [3]	No network connectivity, pure MCU decision-making [4]	Yes (edge computing + cloud platform + V2X exploration) [5–8]
Prediction and time series modeling capabilities	No (only the current frame recognition and mapping, no future trajectory prediction) [2]	Yes (simple linear extrapolation based on the detection box, without using the time series model) [3]	No (only the current distance threshold judgment) [4]	None (current-frame CNN detection only, without LSTM) [5–8]

Note: The “high, medium, and low” power classification is based on the intelligent guide stick scenario, where total system power must remain below 500 mW for >8 h of continuous use. Accordingly, Jetson Nano (5–10 W), Raspberry Pi 4 (3–5 W), STM32H5 (0.1–0.5 W), and Jetson Xavier NX (10–15 W) are classified as high, medium, low, and medium-high, respectively. This classification is for relative reference within this context only. For WeWALK Smart Cane 2 [16] and Glide [17], specifications are from product documentation, not peer-reviewed studies.

4. Role of LSTM in Pedestrian Intention Prediction

Pedestrian intention prediction infers a pedestrian’s future movement trends and behavioral intentions by analyzing interactions between their historical trajectory and the surrounding environment. It represents the key technology for upgrading intelligent guide sticks from “passive response” to “active prediction.” Among the numerous temporal modeling approaches, Long Short-Term Memory (LSTM) networks [9], with their gating mechanism that excels at capturing long-term dependencies in sequences, have established a mainstream position in pedestrian trajectory prediction [22,23]. This chapter first introduces the fundamental principles of LSTM, then reviews mainstream methods and representative improved variants for trajectory prediction and finally analyzes the suitability of deploying LSTM in the resource-constrained intelligent guide stick scenario.

4.1. The Basic Principle of LSTM

Long Short-Term Memory (LSTM) was proposed by Hochreiter and Schmidhuber in 1997 [9] to address the problem that the error signal in traditional Recurrent Neural Networks (RNNs) is prone to exploding or vanishing in long-time-lag tasks. In a standard RNN, the gradient is repeatedly multiplied by the weight matrix during backpropagation. If the matrix norm is less than 1, the gradient will decay exponentially to zero. If it is greater than 1, the gradient will grow exponentially, rendering the network unable to learn long-distance dependencies.

The core innovation of LSTM lies in the introduction of a memory unit and a gating mechanism, with the retention and forgetting of information controlled by the input, forget, and output gates. The LSTM unit contains an information conveyor belt called ‘Cell State’ that runs through the entire time series so that the gradient can propagate stably without attenuation. This mechanism is called ‘Constant Error Carousel’. The cell state is updated by iteration, thereby avoiding the exponential growth that would result from multiplying the time step over.

For the cell state, the forward propagation of the LSTM unit can be fully described by the following set of formulas [9]. Assuming that the input at time step t is, the hidden state at the previous moment is, and the cell state is, the calculation of each gating unit is as follows:

The Forget Gate determines how much information in the cell state should be discarded from the previous moment. The output value is between 0 and 1; 0 means complete forgetting, and 1 means complete retention:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

The input gate controls how much new information in the current input will be updated to the cell state. It calculates the update ratio through the Sigmoid function, and combines the candidate memory generated by the Tanh function to complete the information writing:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

The cell state update is achieved by the weighted combination of the forgetting gate and the previous state, the input gate, and the candidate memory:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (4)$$

The output gate determines which information in the cell state will be output as a hidden state at the current moment:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

In the above formulas, σ denotes the Sigmoid activation function, whose output ranges in $[0, 1]$; $\tanh$ is the hyperbolic tangent activation function with an output range of $[-1, 1]$; $\odot$ denotes element-wise multiplication; W and b are the weight matrices and bias terms of each gating unit, which are the parameters to be learned during network training. The in Formula (3) represents the candidate cell state, which is distinct from the final cell state in Formula (4).

This design enables LSTM to dynamically determine when to remember, when to forget, and when to output information, effectively capturing long-term dependencies of more than a thousand discrete time steps. LSTM is a local operation in space and time, and the computational complexity and the number of weights in each time step are constant. It has high computational efficiency while ensuring modeling ability. Previous studies have shown that LSTM achieves better performance than traditional RNN methods for long-time-lag tasks and learns faster, thereby solving complex problems that previous recursive network algorithms struggle to address [9].

4.2. Mainstream Position of LSTM in Trajectory Prediction and Pedestrian Crossing Intention Prediction

Pedestrian motion prediction is a core technological foundation for enabling proactive warning in intelligent guide sticks, encompassing two primary tasks—trajectory prediction and crossing intention prediction—which differ fundamentally in their output targets and modeling approaches. Trajectory prediction aims to output continuous spatial coordinates of a pedestrian's future positions over several time steps based on their historical motion trajectory, constituting a regression problem. Crossing intention prediction, in contrast, aims to determine whether a pedestrian will cross the road and is typically formulated as a binary classification problem. These two tasks play complementary roles in the safety warning system of intelligent guide sticks, and LSTM, owing to its temporal modeling capabilities, has demonstrated certain application potential in both tasks.

In the field of pedestrian trajectory prediction, methods based on recurrent neural networks, particularly Long Short-Term Memory networks, have become one of the mainstream directions of current research. Huang et al. [22] reviewed the application of neural network methods in pedestrian dynamics research, noting that most existing mainstream

pedestrian trajectory prediction methods are based on LSTM, which achieves sufficient accuracy for short-term prediction. The review points out that neural network models exhibit considerable performance differences across crowd scenarios, and LSTM has received considerable attention in pedestrian trajectory prediction tasks due to its adaptability to sequence data. Similarly, Yin et al. [23], in their narrative review of deep learning applications in vehicle trajectory prediction, noted that RNNs, LSTMs, and Transformers have been widely adopted in this domain and suggested that appropriate integration of interaction perception and scene encoding can effectively improve prediction accuracy.

The influence of LSTM as a foundational architecture for numerous improved variants has continued to grow in recent years. Alahi et al. [10] proposed Social LSTM, which introduces a social pooling mechanism that enables each pedestrian's LSTM unit to accept hidden-state information from neighboring pedestrians, thereby capturing interactions between individual motion patterns and those of surrounding pedestrians. Building on this, Zhang et al. [25] proposed SR-LSTM, which aims to improve the robustness of path prediction in crowded environments by encoding spatial relationships among pedestrians. Alia et al. [26] proposed an enhanced Social LSTM model that introduces a dynamic occupancy-space loss function to guide learning and avoid real collisions; experimental results show that across various crowd scenarios ranging from low to ultra-high density, the collision rate was reduced by 31% compared with the baseline scheme, while average and final displacement errors were reduced by 5% and 6%, respectively. The above works all employ LSTM as their core encoder, providing evidence to a certain extent for its foundational role in complex dynamic environments.

Beyond pedestrian-dense scenarios, LSTM has also been explored extensively in autonomous driving and vehicle trajectory prediction. Yan et al. [27] proposed an algorithm that combines LSTM with attention mechanisms to simultaneously model vehicles' short-term target points and predict multiple possible driving destinations, with the aim of improving trajectory prediction accuracy. Chen et al. [28] proposed Att-LSTM for pedestrian trajectory prediction, combining it with an improved social force model to construct a data-driven stacking fusion framework, achieving results that outperformed some comparison methods in unsignalized intersection scenarios. These cross-scenario applications provide further reference evidence for the generality of LSTM as a baseline architecture for trajectory prediction.

In the task of crossing intention prediction, LSTM has also demonstrated certain applicability. Dasgupta et al. [29] showed that LSTM and its variants can effectively capture dynamic changes in pedestrian actions for pose-sequence-based crossing intention classification, providing a reference basis for early prediction of crossing behavior. In recent years, lightweight GNN-based models have also emerged in this domain, such as PedGNN proposed by Riaz et al. [30], which achieves a GPU memory footprint of approximately 27 KB and millisecond-level inference latency. However, for the specific scenario of intelligent guide sticks targeting MCU deployment, end-to-end deployment of GNN models on STM32-class platforms currently lacks research validation, whereas LSTM already has empirical data for lightweight deployment on MCUs, with an INT8-quantized model size of 136.5 KB and inference latency of 21 ms [31]. Furthermore, guide sticks in actual use need to continuously perceive the motion trends of dynamic obstacles, which requires the model to possess robust long-term dependency modeling capabilities—precisely the original design motivation of the LSTM gating mechanism [9]. Therefore, under the MCU deployment constraints of intelligent guide sticks, LSTM currently has relatively stronger evidence supporting its applicability compared with alternative approaches such as GNNs.

In summary, the research foundation of LSTM in trajectory prediction and crossing intention prediction can be characterized from the following two aspects. First, its gating

mechanism provides certain advantages in capturing trajectory motion patterns ranging from short-term to long-term dependencies, making it suitable for both regression prediction of continuous trajectory coordinates and providing temporal feature representations for discrete intention classification, while alleviating the vanishing gradient problem that plagues traditional RNNs to some extent. Second, its architectural flexibility enables integration with modules such as attention mechanisms, graph networks, and generative models, as explored in representative works such as Social LSTM [10] and SR-LSTM [25]. These characteristics provide a corresponding theoretical foundation for deploying LSTM on resource-constrained devices such as intelligent guide sticks.

4.3. Applicability Analysis of LSTM for Intelligent Guide Sticks

In the embedded deployment scenario for intelligent guide sticks, model selection must strike a balance between temporal modeling capability and hardware resource constraints. As a wearable assistive device for visually impaired individuals, intelligent guide sticks impose stringent requirements on real-time performance, power consumption, memory footprint, and portability: end-to-end latency is typically required to be controlled within 120 ms, total system power consumption must remain below 500 mW to support continuous use for more than 8 h, and Flash and SRAM resources at the microcontroller level are often only a few hundred kilobytes. Consequently, the adopted temporal prediction model must not only adequately capture the movement trends of dynamic obstacles such as pedestrians and vehicles but also support real-time inference on low-power embedded platforms.

In terms of temporal modeling capability, LSTM [9], through its gating mechanism, demonstrates relatively strong performance in capturing long-term dependencies in sequence data, a capability that has been reasonably well validated in trajectory prediction and intention recognition tasks [22,23]. Among alternative recurrent architectures, GRU [32] performs competitively in short-term prediction [29], yet its accuracy degrades as the prediction horizon increases [33]—a limitation worth noting for guide sticks that require 1–2 s of advance warning. Temporal Convolutional Networks [34], while capable of parallelized inference, have fixed receptive fields that limit flexibility in adapting to variable-length historical dependencies, and their parallel computation advantage is difficult to fully realize in MCU-level single-core execution environments. Graph-based social models [35] effectively capture pedestrian interactions, but their computational cost grows quadratically with crowd density, offering relatively limited benefits in the low- to medium-density scenarios typical of guide stick operation. Overall, LSTM demonstrates a favorable balance in modeling flexibility, long-term dependency capture, and interaction adaptability.

Regarding embedded deployment feasibility, it is necessary to distinguish between theoretical architectural advantages and empirical deployment evidence on MCU-class hardware. LSTM relies on linear gated computations, avoiding the quadratic complexity of self-attention, which provides a favorable theoretical foundation for deployment on resource-constrained devices. More importantly, concrete deployment cases are available for reference: Zhou et al. [31] deployed a DeepConv LSTM model on the Arduino Nano 33 BLE Sense (ARM Cortex-M4) using full integer quantization, achieving a compressed model size of 136.5 KB and a measured inference latency of 21 ms on a human activity recognition (HAR) task. Although this task differs from trajectory prediction, the hardware platform (ARM Cortex-M4) and resource constraints (sub-MB Flash and SRAM) are comparable to the guide stick deployment environment, making this result a valuable empirical reference for the guide stick scenario.

In contrast, the empirical evidence for other lightweight temporal models is relatively weaker. GRU [32], despite having fewer parameters, shares a similar recurrent compu-

tational paradigm with LSTM and offers no significant advantage in MCU deployment. Lightweight Transformers (such as Longformer [36] and Reformer [37]) reduce theoretical complexity through sparse attention; TransLSTM [38], as a hybrid architecture, has been validated as lightweight in trajectory prediction (78K parameters, 4.9 ms inference latency); EdgeVTP [39] and other edge-oriented trajectory predictors have also explored the combination of Transformers with graph modeling. However, benchmark reports on Transformer-based models for trajectory prediction tasks on ARM Cortex-M class MCUs remain relatively limited in the current literature. The memory bandwidth requirements and inference latency of attention mechanisms on such platforms have not been fully characterized, and mature inference libraries for ARM Cortex-M are not yet widely available. Emerging approaches combining Transformers with graph neural networks (e.g., PedGT [40]) have demonstrated potential for efficient spatio-temporal encoding on embedded platforms, but their MCU-level deployment under the latency and memory constraints of intelligent guide sticks remains to be validated. Attention-enhanced RNNs [41] and Kalman filter-RNN hybrid models [42] introduce additional computational overhead, such as the $O(T)$ complexity of attention mechanisms and the matrix inversion required in Kalman filtering, which may impose considerable pressure on MCU resources within millisecond-level inference budgets. TinyML-targeted time-series models [43,44] offer high energy efficiency but have not yet been validated on pedestrian trajectory or intention prediction tasks, nor do they possess the capability to model social interactions and scene semantics. Overall, in terms of documented MCU deployment cases with quantifiable hardware metrics, LSTM currently has relatively stronger empirical support.

Beyond modeling capability and deployment considerations, the complex dynamic environments encountered by intelligent guide sticks require models to possess scene adaptability—the ability to integrate multi-modal perception information and make early judgments about pedestrian intentions. LSTM’s architectural flexibility provides a solid foundation for scene adaptability: Social LSTM [10] and SR-LSTM [25] capture pedestrian interactions through social pooling and spatial relationship encoding; Social GAN [45] generates multi-modal trajectory predictions to address pedestrian uncertainty; T-Pose-LSTM [46] performs pedestrian trajectory prediction based on 3DOF pose trajectories in service robot scenarios; GazeX-LSTM [47] combines gaze information with contextual cues for pedestrian trajectory prediction; CNN-BiLSTM [48] demonstrates the synergistic effect of CNN and BiLSTM feature extraction in sensor-based pedestrian activity recognition; and context-aware LSTM [49] models multi-agent motion through dynamic and static contextual information. These works demonstrate that LSTM, as a foundational backbone, can integrate attention mechanisms, graph networks, generative models, and multi-modal inputs to address diverse interaction and scene understanding requirements. This extensively adapted algorithmic ecosystem provides a relatively rich repository of algorithms for guide stick development—a level of maturity that other MCU-targeted temporal models have not yet fully achieved in the pedestrian trajectory domain.

In summary, although GRU [32], TCN [34], lightweight Transformers [36,37,39], attention-enhanced RNNs [41], KF-RNN hybrid models [42], graph-based models [35], and TinyML models [43,44] each have advantages in specific scenarios, they each exhibit certain limitations—whether in long-term modeling accuracy, empirical deployment validation on MCU hardware, or scene adaptability—that constrain their direct applicability to intelligent guide sticks. LSTM’s strengths lie in its combination of: (1) validated long-term temporal modeling capability in trajectory and intention prediction tasks; (2) documented MCU deployment evidence—achieving 136.5 KB model size and 21 ms inference latency on ARM Cortex-M4 hardware through full integer quantization [31]; and (3) architectural extensibility demonstrated across numerous task-specific variants. These characteristics

position LSTM as one of the temporal models with relatively stronger empirical support for intelligent guide stick development. A comprehensive comparison of the mainstream temporal prediction models discussed above is summarized in Table 3. Nevertheless, its practical effectiveness in this specific deployment scenario remains to be further validated through end-to-end implementation on real prototype hardware.

Table 3. Comparison of the comprehensive applicability of mainstream temporal prediction models for intelligent guide stick deployment.

Model	Temporal Modeling Capability	MCU Deployment Feasibility	Scene Adaptability	Overall Suitability
LSTM	Strong, gated mechanism captures long-term dependencies [9]	High: validated on ARM Cortex-M4 (21 ms inference, 136.5 KB model, 97% accuracy) [31]	High: validated in pedestrian trajectory prediction, supporting social interaction and multimodal fusion [10,25,45–49]	Highly suitable (for intelligent guide stick constraints)
GRU	Moderate; competitive in short-term prediction; weaker in long-term dependencies [32]	Moderate: fewer parameters than LSTM, but similar recurrent architecture offers no substantial deployment advantage [32]	Limited: fewer variants and extensions for complex scene integration	Moderate
TCN	Moderate; fixed receptive field constrains long-range dependency capture [34]	Low: parallel computation advantage lost on single-core MCUs [34]	Limited: no inherent mechanism for modeling social interactions	Moderate-Low
Lightweight Transformer	Strong in theory via self-attention; memory bandwidth remains high [36,37]	Moderate: TransLSTM shows 4.9 ms inference potential [38], but documented deployment on STM32-class MCUs for trajectory prediction remains limited [36,37]	Moderate: attention can model long-range dependencies but lacks proven scene integration	Low (emerging research shows potential, but MCU deployment under intelligent guide stick constraints remains unvalidated)
Attention-enhanced RNN	Strong; but extra O(T) complexity overhead	Low: additional attention computation may violate real-time constraints on MCUs	Moderate: selective attention benefits context awareness, but cost is high	Moderate-Low
KF-RNN Hybrid	Moderate; handles sensor noise robustly, but matrix inversion scales quadratically [42]	Very Low: matrix operations too computationally heavy for MCUs [42]	Moderate: principled uncertainty handling, but limited scene semantics	Low
Graph-based Model	Strong in crowded scenarios; message passing scales $O(N^2)$ with pedestrian count [35]	Low: high computational cost for limited benefit in low-to-medium density scenes [35]	Moderate: specialized for interaction modeling, but scene-dependent	Moderate (scene-dependent)
TinyML Model	Not validated for trajectory or intention prediction tasks [43,44]	High: KB-level memory footprint, but for general time-series forecasting [43,44]	None: lacks social interaction and scene semantics modeling	Unvalidated

Note: Assessments are relative comparisons based on the reviewed literature, not absolute performance guarantees. “High” indicates proven feasibility with measured hardware metrics on MCU-level platforms; “Low” indicates significant barriers preventing practical deployment; “Moderate” indicates partial suitability under specific conditions. Actual performance depends on specific model configurations and hardware implementations.

While MCU-only deployment is feasible for highly compressed LSTM models—as demonstrated by the DeepConv LSTM deployment [31] achieving 21 ms inference latency with a 136.5 KB model size and 97% accuracy on ARM Cortex-M4, and Slim LSTM [20] with only 3.4K parameters—the additional computational demands of multi-sensor fusion and real-time object detection may favor FPGA acceleration to meet the sub-120 ms end-to-end latency requirement in more demanding scenarios. Recent advances in FPGA-based LSTM acceleration, including C-LSTM [11] and FINN-GL [12], provide complementary solutions for such cases. However, given the power and cost constraints of wearable assistive devices,

MCU-first deployment remains the primary focus of this review, with FPGA-based acceleration discussed as a supplementary approach for specific high-performance requirements.

5. LSTM Lightweight Method

To enable real-time deployment of LSTM on resource-constrained embedded platforms such as intelligent guide sticks, researchers have explored a variety of lightweight approaches from different perspectives. These methods can be categorized into five main types according to their compression strategies: architecture simplification, which reduces trainable parameters by fixing parts of the gating signals; low-rank decomposition, which approximates weight matrices as products of low-rank factors through techniques such as SVD; structured pruning, which introduces regular matrix structures (e.g., block-circulant matrices) to facilitate hardware acceleration; quantization, which reduces numerical precision to shrink model size and accelerate inference; and knowledge distillation, which transfers knowledge from a large teacher model to a lightweight student model. The following sections review each category in detail, with their compression effectiveness, accuracy preservation, and hardware applicability.

5.1. Streamlined Architecture: Slim LSTM

The architecture reduction method aims to reduce the number of trainable parameters in the model structure, with Slim LSTM [20] as a representative work in this direction. The standard LSTM [9] controls the information flow through three gating signals. Each gate is driven by a function that includes an input weight matrix (processing external inputs), a hidden-state weight matrix (processing historical information), and a bias term. Although this structure achieves strong performance, it also exhibits obvious parameter redundancy and over-parameterization, leading to large model size and high computational complexity, and making it difficult to deploy directly on resource-constrained edge devices.

In response to this problem, Akandeh and Salem [20] proposed a lightweight variant, Slim LSTM, which saves computational resources by significantly reducing the number of parameters in the gating signals. The core idea of these variants is that each gated signal is itself a simple recurrent neural network, and the generation of gated signals need not rely on full matrix multiplication of the input, historical state, and bias. By identifying and removing redundant parts, the model can be significantly compressed while maintaining performance.

In this kind of variant, the authors focus on two of the simplest structures: LSTM_6 and LSTM_C6 [20].

1. LSTM_6: LSTM_6 retains the three gating signals of the standard LSTM, but its core simplification is to remove the matrix multiplication part of the external input in all gating signals and replace it with a trainable constant or identity matrix. It means that the gating signal is no longer dynamically adjusted to real-time changes in the input sequence but instead relies mainly on feedback from the previous hidden state. This structure replaces the input matrix of each gate with a constant, fundamentally eliminating the need for full matrix multiplication in the input, forget, and output gates, thereby greatly reducing the total number of parameters.
2. LSTM_C6: As a further simplification of LSTM_6, LSTM_C6 not only inherits the clipping of gating signals from LSTM_6 but also simplifies the memory-cell input block of the cell body. It removes the full matrix multiplication in the input block and retains only the linear scaling operation. This design calculates the entire LSTM unit, primarily focusing on updating the state matrix, thereby further reducing the model's complexity.

The experimental results show that under the standard LSTM configuration with an input dimension of 32 and a state dimension of 100, the parameter-count compression rates of LSTM_6 and LSTM_C6 are extremely significant. Taking LSTM_C6 as an example, its parameter count is reduced from about 53,200 to 3400 under the same configuration; the compression rate exceeds 93%, and the computational complexity is significantly reduced. Multiple benchmark tests [20,50] show that these parameter-reduced Slim LSTM variants are broadly comparable to standard LSTM in training and inference performance on datasets such as IMDB review sentiment [50] and 20 Newsgroups multi-classification. It is also noted that by reasonably adjusting hyperparameters such as the learning rate, the number of hidden units, and the gating constant, these minimalist lightweight models can achieve results similar to those of standard LSTM.

Due to its low parameter count and computational overhead, Slim LSTM is suitable for online training and real-time inference on portable devices with limited computing resources, provided that retraining from scratch is acceptable. This method of streamlining the structure itself provides a good starting point for further compression operations, such as quantization and pruning. It serves as a suitable basis for lightweight models for embedded devices, such as intelligent guide sticks. It should be noted, however, that the architectural simplification of Slim LSTM comes at the cost of requiring a redesigned training pipeline and cannot be directly applied to compress existing pre-trained models; furthermore, its performance depends on appropriate hyperparameter tuning.

5.2. Low-Rank Decomposition: SVD/PCA/MPO

Low-rank decomposition is an important technique in model compression, whose core idea is to approximate the original high-dimensional weight matrix as the product of several low-rank submatrices. By truncating non-critical singular value information, redundant parameters and storage requirements can be reduced in a principled manner from a mathematical perspective without significantly affecting model accuracy. Owing to its mathematical rigor and accuracy-preserving capability, this method is particularly suitable for safety-critical edge deployment scenarios such as intelligent guide sticks.

Based on different decomposition strategies, low-rank decomposition has derived the following representative technical approaches:

1. Singular Value Decomposition (SVD) is a fundamental and widely used method in low-rank decomposition. By decomposing the weight matrices of LSTM layers and fully connected layers into products of multiple low-rank matrices, SVD selectively preserves the largest singular values and their corresponding singular vectors, thereby replacing the original high-dimensional matrices with low-rank approximations. Liu et al. [51] proposed a lightweight WMD-LSTM method that applies SVD to the weight matrices of each LSTM gating unit and combines it with a dynamic learning-rate adjustment strategy to optimize training. This approach ultimately reduced model parameters by 45% and compressed the model volume to 45% of its original size, maintaining prediction accuracy without decline. In another study, Dai et al. [52] proposed the LSTM-SVD method, which accelerated training speed to 3.9 times that of the original model while reducing the parameter count of gated RNNs by more than 10 times, without compromising accuracy.
2. General low-rank matrix factorization is a method suitable for compressing a wider range of LSTM structures, flexibly employing SVD, LU, QR, and other low-rank approximation strategies. Prabhavalkar et al. [53] applied low-rank decomposition to LSTM acoustic models in embedded speech recognition, demonstrating that even when the model is compressed to one-fifth of its original size, the decomposed model maintains performance comparable to the original. Liu et al. [51] analyzed the impact

of low-rank decomposition on different LSTM recurrent structures. They found that additive rather than multiplicative recurrence is more important for compression effectiveness, thereby validating low-rank decomposition as an effective post-processing compression method for LSTM edge deployment.

3. Matrix Product Operator (MPO) represents another frontier direction in low-rank decomposition. MPO is a tensor network representation method originating in quantum many-body physics that decomposes the original weight matrix into a product chain of multiple low-dimensional tensors, while discarding irrelevant components during the decomposition to achieve systematic compression of large models. Experiments show that MPO-LSTM achieves superior post-compression performance compared to traditional pruning methods in sequence classification and prediction tasks [54]. Similarly, in action recognition tasks, Pan et al. [55] proposed the TR-LSTM model, which employs low-rank tensor ring decomposition to reparameterize the input-hidden transformation, demonstrating good numerical stability and compression efficiency.
4. Principal Component Analysis (PCA), as a classical dimensionality reduction algorithm, is applied in LSTM lightweighting primarily through input-side data-level dimensionality reduction—an “indirect compression” strategy to achieve overall model efficiency. PCA performs a linear orthogonal transformation on the input features to extract the most discriminative principal components, which replace the original high-dimensional inputs, thereby reducing the number of parameters to be learned in the model’s first layer. In multivariate time series prediction tasks, the QPCA-LSTM method proposed by Boddu et al. [56] introduced a quartile PCA mechanism based on traditional PCA, increasing the explained variance by 10% and confirming the statistical superiority of its predictive performance through the Diebold-Mariano hypothesis test (at a 1% significance level).

From an applicability perspective, SVD [51,52], with its progressive truncation strategy, achieves efficient compression while maintaining accuracy, making it particularly suitable for scenarios requiring “lossless” compression. PCA [56] achieves feature dimensionality reduction with extremely low computational overhead, demonstrating unique advantages in high-input-dimension scenarios. Tensor decomposition methods such as MPO [54] and tensor rings [55], with their high flexibility and extremely high upper limits on compression, offer clear advantages for extreme scenarios demanding high compression ratios. It should be noted that low-rank decomposition is essentially a linear approximation method; when the target weight matrix lacks a low-rank structure, an excessively high compression ratio may significantly degrade accuracy. This theoretical limitation must be considered alongside specific tasks in practical engineering.

5.3. Structured Pruning: C-LSTM

Structured pruning addresses the irregularity problem caused by traditional pruning methods by introducing a regular matrix structure, with C-LSTM [11] serving as a structured compression and acceleration framework for FPGA hardware platforms. In acoustic recognition systems, although accuracy can be significantly improved by increasing the size of the LSTM model, the resulting model size consumes more FPGA resources and reduces design efficiency. Existing work uses pruning-based compression techniques to reduce model size and accelerate inference. However, the randomness of pruning techniques transforms the original dense weight matrix into a highly unstructured sparse matrix, resulting in an unbalanced computational load and irregular memory access, which seriously affect the system’s overall performance and energy efficiency.

To overcome the above problems, C-LSTM [11] adopts structured compression using a block-circulant matrix rather than a sparse matrix. In mathematical terms, the weight

matrix dimension is set to $k \times k$, which is divided into sub-blocks of size $b \times b$. Each sub-block satisfies the properties of a circulant matrix (each row is obtained by shifting one bit to the right from the previous row). Then each sub-block performs circular convolution independently, and the fast Fourier transform (FFT) can efficiently accelerate the multiplication operation by exploiting the Fourier diagonalization properties of the circulant matrix. The specific calculation process for the block-circulant matrix is as follows: First, the product form of the block-circulant matrix C and the vector x is constructed, and the diagonalization is converted into element-by-element multiplication using the discrete Fourier transform. Finally, the result is restored by an inverse transformation. This design reduces the storage requirement from $O(k^2)$ to $O(k)$ and the computational complexity from $O(k^2)$ to $O(k \log k)$ by using the fast Fourier transform algorithm. The experimental results show that C-LSTM achieves up to 18.8- and 33.5-fold performance improvements, respectively, with a very small loss in accuracy [11].

With a very small loss in accuracy, the data path and activation function are quantized to 16 bits, and hardware optimization and deployment are automatically performed via the integrated framework. More importantly, C-LSTM proposes an end-to-end integrated framework that automatically optimizes and deploys multiple LSTM variants on FPGAs. At the programming-optimization level, asynchronous pipeline processing between the CPU and FPGA overlaps data transmission and inter-core computation, further reducing data access delays. At the same time, subsequent E-RNN work [57] uses the alternating direction method of multipliers (ADMM) for more accurate block-cycle training. At the same accuracy, the energy efficiency is more than 2 times higher than that of C-LSTM and 37.4 times higher than that of the early ESE design [58]. In addition, the design idea of C-LSTM is inspired by systems theory in the field of structured matrices. Relevant reviews point out that structured weight-matrix representation methods, such as circulant and block-circulant matrices, are becoming an important paradigm for building efficient LSTM hardware accelerators. However, this scheme has limitations in versatility: its highly FFT-dependent acceleration mechanism makes it difficult to realize its advantages on general-purpose processors or lightweight MCUs without hardware-level FFT support, and the migration cost is relatively high. At the same time, for intelligent guide-stick scenarios that require end-to-end inference on MCUs, hardware-software co-design still entails additional workload.

5.4. Quantization Method: FINN-GL

FINN-GL [12] is an FPGA-oriented LSTM hybrid-precision quantization deployment framework, which is based on the Xilinx open-source FINN framework [59] for general expansion. The original FINN [59] is primarily oriented toward feedforward and convolutional neural networks and lacks support for recurrent structures such as LSTMs. To address this shortcoming, FINN-GL [12] uses the Scan operator in the ONNX specification to accurately model the recurrent computing characteristics of LSTM—the Scan operator can iteratively process sequence input and update state variables at each time step, which is highly consistent with the hidden state and cell state update mechanism of LSTM.

In terms of implementation, FINN-GL [12] employs a comprehensive process that combines quantization-aware training and compilation optimization. Developers first use the Brevitas library to perform quantization-aware training on the LSTM and derive a QONNX model that supports arbitrary-bit-width mixed-precision quantization. FINN-GL then identifies the Scan operator structure in the computation graph and maps it to the HLS kernel module in the FINN hardware library, thereby significantly reducing hardware resource usage by retaining the loop structure. The framework also uses the QCDQ quantization method, allowing the bit widths of 11 independent quantizers in the

LSTM to be configured separately, making it convenient for designers to fine-tune the trade-off between inference accuracy and hardware resources. Finally, the framework integrates the Vitis HLS toolchain to generate a deployable hardware IP core automatically.

Experiments show that when the weight and activation quantization are 8 bits, the quantized LSTM accelerator achieves extremely low inference latency while maintaining high accuracy, meeting the needs of edge real-time processing [12]. The framework supports end-to-end automated deployment: users only need to provide trained models and quantization parameters; the remaining steps are handled automatically by the toolchain. FINN-GL provides an efficient, automated, flexible, and controllable technical pathway for practical applications—such as intelligent guide sticks—that require deploying LSTM-based intention prediction on low-power edge devices. However, the deployment scheme of FINN-GL is highly dependent on FPGA platforms and the Xilinx development toolchain, with a high hardware design barrier and relatively limited generality. For intelligent guide-stick scenarios where MCUs serve as the primary computing core, their applicability requires careful evaluation based on specific hardware conditions.

5.5. Knowledge Distillation

Knowledge distillation is a model compression technique that enables knowledge transfer via a teacher-student architecture, first proposed by Hinton et al. in 2015 [60]. The core idea is to compress the generalization ability of the large-scale, high-performance teacher model into the lightweight student model, thereby significantly reducing model complexity while maintaining high accuracy. The framework contains three key elements: the teacher model, which is usually a deep neural network with a large parameter count and high precision but slow inference speed; the student model, which has a simple structure and a small number of parameters, to imitate the teacher model's output behavior; and the distillation loss function, which measures the difference between the student model's output and the teacher model's output. Unlike traditional hard-label training, knowledge distillation introduces the category probability distribution from the soft target—the teacher model. The soft target contains rich inter-class similarity information, and these relative probabilities encode the teacher model's understanding of the relationships between categories. The student model can learn structured knowledge beyond a single label by simultaneously fitting hard labels (true categories) and soft targets (teacher output), thereby maintaining high accuracy while greatly reducing the number of parameters.

In the field of spatio-temporal prediction, the frequency-aligned knowledge distillation framework (SDKD) proposed by Li et al. [61] is a representative work. This study extracts multi-scale spectral features (including high-frequency details and low-frequency trends) from the teacher model's latent space as migration targets to guide student model learning. In the equations dataset, the lightweight student model after SDKD distillation achieves a significant performance improvement, with an 81.3% reduction in root mean square error and a 52.3% reduction in mean absolute error. In terms of cross-architecture knowledge transfer, the adaptive dual distillation framework proposed by Cheng et al. [62] transfers the knowledge of the large LSTM teacher model to the lightweight bidirectional GRU student model, achieving a compression rate of 5.34 times on the NASA C-MAPSS dataset, with the prediction accuracy of the distilled student model even exceeding that of the original LSTM teacher model. Compared with pruning and quantization, knowledge distillation has three advantages: supporting cross-architecture knowledge transfer (e.g., LSTM to GRU), being combinable with other lightweight methods, and preserving the teacher model's representational ability for complex temporal patterns. Its limitation is that it typically requires pretraining a large teacher model, which is costly.

5.6. Methods Comparative Summary

On the whole, each method demonstrates its core strengths in different aspects. Slim LSTM [20] achieves a parameter compression rate exceeding 93% through architectural simplification, with essentially no loss in accuracy. SVD [51,52], as a post-processing method, can directly compress pre-trained models without requiring retraining. INT8 quantization achieves 21 ms inference latency and 97% accuracy on ARM Cortex-M4 [31], with complete hardware measurement data reported on an MCU-class device, making it one of the well-validated schemes for MCU deployment among the methods surveyed in this review. C-LSTM [11] reduces storage and computational complexity from $O(k^2)$ to $O(k)$ and $O(k \log k)$, respectively, through block-circulant matrices. Knowledge distillation [60–62] supports cross-architecture transfer (e.g., LSTM to GRU) and can be combined with other lightweight methods.

Nevertheless, each method has inherent limitations. Slim LSTM [20] must be retrained from scratch, cannot directly compress existing pre-trained models, and lacks publicly reported hardware validation metrics on MCUs. SVD [51,52] has a compression ratio ceiling of approximately 50%, and its linear approximation nature may fail for tasks involving highly nonlinear trajectory patterns. INT8 quantization requires hardware support for INT8 or mixed-precision instructions and incurs approximately 1–2% loss in accuracy [31]. C-LSTM [11] and FINN-GL [12] are primarily oriented toward FPGA platforms, heavily relying on FFT hardware acceleration units and the Xilinx development toolchain, offering limited advantages on general-purpose MCUs while demanding hardware-software co-design expertise. The applicability of knowledge distillation hardware [60–62] depends entirely on the chosen student architecture; it typically requires pre-training a large teacher model, incurs high training costs, and carries the risk of a teacher-student mismatch. Hybrid strategies—such as applying Slim LSTM followed by quantization, or using a quantized model as the student for knowledge distillation—may combine the strengths of multiple approaches but also potentially compound their respective limitations, with their actual effectiveness requiring task-specific validation.

For intelligent guide stick deployment, no single method is universally optimal across all hardware conditions and accuracy requirements. Practical selection should consider the target hardware platform, accuracy tolerance, development cycle, and the availability of pre-trained models. When accuracy requirements are strict and retraining is feasible, Slim LSTM [20] offers a suitable option. When a pre-trained model must be compressed without accuracy loss, SVD [51,52] offers a suitable post-processing pathway. In latency-sensitive MCU applications, INT8 quantization [31] is one of the well-validated approaches among those reviewed, with complete hardware measurement data reported on an MCU-class device. When the target platform is an FPGA, and the team possesses hardware-software co-design capabilities, C-LSTM [11] or FINN-GL [12] are more appropriate. Notably, DeepConv LSTM [31] is one of the few studies that provide complete, publicly reported hardware measurement data on an MCU-class device, underscoring the critical need for future research to report standardized hardware metrics—including model size, memory footprint, inference latency, and power consumption—when proposing lightweight LSTM models, thereby facilitating the effective translation of algorithmic innovations into practical assistive devices. Table 4 provides a quantitative comparison of these five methods in terms of compression effectiveness, accuracy preservation, hardware applicability, and key hardware metrics.

Table 4. Quantitative comparison of LSTM lightweight methods.

Method Category	Compression Effect	Accuracy Preservation	Hardware Applicability	Model Size	Memory Footprint	Inference Latency	Power Consumption	Critical Limitation for Intelligent Guide Sticks
Slim LSTM	>93% (53.2 k→3.4 k) [20]	≈Flat [20]	General-purpose MCU (theoretical) [20]	3.4 k parameters [20]	Not reported	Not reported	Not reported	Must retrain from scratch; cannot compress existing models [20]
SVD	45% volume reduction [51,52]	Lossless under specific conditions [51,52]	Primarily FPGA [51,52]	55% of original model [51,52]	Not reported	3.9× training speedup [51,52]	Not reported	Compression ceiling ≤ 50%; linear approximation may fail for nonlinear patterns [51,52]
C-LSTM	$O(k^2) \rightarrow O(k)$, $O(k^2) \rightarrow O(k \log k)$ [11]	Minimal loss [11]	FPGA (FFT hard IP required) [11]	$O(k)$ storage [11]	Not reported	18.8× performance improvement [11]	33.5× energy efficiency improvement [11]	FPGA-only; requires HW-SW co-design [11]
Quantization (INT8)	74% (513 KB→136 KB) [31]	Minor loss (98.24%→97%) [31]	ARM Cortex-M4 (vali-dated)/FPGA [31]	136.5 KB [31]	29.1 KB SRAM [31]	21 ms [31]	Not reported	Requires INT8/mixed-precision HW support [31]
Knowledge Distillation	MSE↓81.3%, MAE↓52.3% (SDKD) [60–62]	Controllable (>95% of teacher) [60–62]	Depends on student model [60–62]	Depends on student model	Not reported	Not reported	Not reported	High training cost; teacher-student mismatch risk [60–62]

Note: “Not reported” indicates that the specific metric was not provided in the cited reference. As the experimental platforms, task types, and evaluation protocols vary across studies, caution should be exercised when making direct cross-method comparisons. The arrows (→ and ↓) indicate that the values are reduced after compression or optimization compared to the original values.

6. The Application Case of LSTM in Intelligent Guide Stick

The application of LSTM to intelligent guide sticks and related assistive devices spans multiple dimensions, including speech guidance, trajectory prediction, positioning and navigation, edge computing, and Internet technology collaboration. Existing studies have preliminarily validated the feasibility of LSTM-based methods across these scenarios, covering both algorithmic innovation and hardware implementation. Representative works in each category are reviewed below, with a focus on their methodological approaches, performance outcomes, and implications for intelligent guide stick deployment.

6.1. Speech Guidance System Based on CNN-LSTM

In view of the various activities and scene restrictions faced by visually impaired people in their daily lives, Kambham et al. [63] proposed a multi-modal deep learning system that combines the DepthNet module and the convolutional LSTM layer to detect obstacles such as vehicles and pedestrians in the travel path and provide early warning and guidance through voice. The system integrates two functions of real-time obstacle detection and voice guidance. Compared with the traditional single-function auxiliary technology, it tries to build a more comprehensive support scheme. CNN is used to extract spatial features, and LSTM is used to capture temporal features. The fusion of the two enables the model to identify obstacle types and track their motion patterns in continuous video frames. The DepthNet module focuses on extracting depth information from monocular input and outperforms traditional methods in spatial context awareness. The system performs pre-training and fine-tuning on two autonomous driving benchmark datasets, MS COCO [64] and BDD100K [65], to ensure robustness in a variety of environments. To improve the perceptual accuracy for dynamic obstacles, the system also introduces an optical flow algorithm. By accurately estimating pixel motion between consecutive frames, the system enhances real-time tracking of dynamic obstacles, such as vehicles and pedestrians. It ultimately provides users with environmental information and early warnings via voice descriptions.

Furthermore, some studies have used YOLO [14,66] combined with optical flow motion estimation to achieve full-scene, accurate perception of static and dynamic obstacles. In the field of intelligent guide sticks, Gao Shunqiang et al. [21] used an STM32F103 as the sensor data acquisition controller. They combined it with the YOLO model deployed on the Jetson Xavier NX edge computing platform to realize visual perception and obstacle recognition, demonstrating the feasibility of deploying deep learning models on heterogeneous embedded platforms. Megharsh et al.'s Sense Stride system [4] is based on the STM32H563ZI microcontroller, integrates the VL53L0X ToF sensor and PIR motion sensor, and realizes five-direction obstacle detection and tactile vibration feedback. The average response time is only 50 milliseconds, and the obstacle detection accuracy is 95%, providing a hardware feasibility reference for deploying lightweight models such as CNN-LSTM on an MCU.

It is worth noting that although Kambham et al.'s system [63] is more comprehensive, it still focuses on obstacle recognition and voice description of the current frame and cannot predict the future motion trend of the dynamic target. This deficiency provides a direction for improvement in subsequent trajectory prediction and behavioral intention analysis when combined with LSTM. In general, this study verifies the engineering feasibility of the deep learning multi-modal fusion scheme for the intelligent guide stick scenario. It lays the foundation for embedding LSTM-based trajectory prediction capabilities into the speech guide system.

6.2. Obstacle Avoidance Based on LSTM Trajectory Prediction

The traditional deep reinforcement learning obstacle-avoidance method relies on instantaneous perceptual information to make obstacle-avoidance decisions. In essence, it still falls under reactive control and struggles to cope with emergencies, such as sudden changes in the trajectories of dynamic obstacles. Figure 4 outlines the overall logic of LSTM-based collision risk prediction and proactive obstacle avoidance, which predicts future positions from historical trajectories and adjusts avoidance strategies in advance accordingly. Poulet et al. [67] proposed a collision-risk prediction method based on LSTM, which achieved active obstacle avoidance through short-term position prediction. The core structure of this method is divided into three levels: the perception layer uses the LSTM model to learn the historical motion trajectory of the agent and predict its next-moment position; the evaluation layer dynamically modulates the reward function of the deep Q-learning network [68] with the predicted location information and imposes penalties in advance before potential collisions occur. The control layer performs obstacle avoidance actions based on the decision output of the deep Q-learning network. Specifically, the LSTM model accepts the position sequences of the robot and its surrounding agents over the past several time steps as input and outputs an estimate of the future position by learning the motion law. Then, this estimated result is used to adjust the reward function of the deep Q-learning network dynamically. If the predicted trajectory poses a collision risk, a negative reward is immediately given to encourage the agent to adjust its direction of travel in advance, thereby avoiding danger before a collision occurs and enabling the leap from reactive to predictive obstacle avoidance. The paper clearly shows that although the sampling frequency in the test environment is only 1 Hz, this method still significantly reduces collisions and improves motion stability; owing to its low computational overhead, it is particularly suitable for implementation on embedded systems.

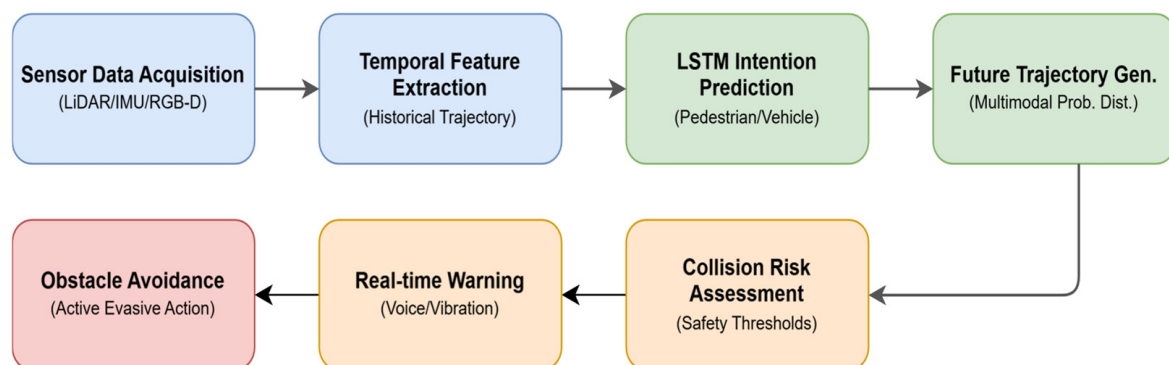


Figure 4. Pedestrian intention prediction and active obstacle avoidance flow chart based on LSTM.

The engineering feasibility of this method has been fully verified on the mobile robot platform. Mulás-Tejeda et al. [69] implemented a dynamic obstacle-avoidance method based on an LSTM on the TurtleBot3 robot. They used lidar to obtain environmental information, trained the model on user-controlled running trajectory data, and predicted the robot's linear and angular velocities. Physical experiments show that the model can successfully guide the robot to avoid dynamic obstacles and reach the target point, with a verification accuracy of 98.02%. Zhang et al. [70] introduced an LSTM-based approach, building on the Proximal Policy Optimization (PPO) algorithm, and improved dynamic obstacle-avoidance performance by optimizing the Actor-Critic network. The effectiveness of the algorithm was verified in both the Gazebo simulation environment and the TurtleBot3 real-world scene.

Although the above studies primarily use mobile robots as carriers, their LSTM-based trajectory prediction mechanisms are methodologically aligned with the pedestrian intention prediction task for intelligent guide sticks. Both application scenarios require the system to predict the future positions of moving pedestrians or vehicles and to issue warnings before collisions occur. The primary differences lie in the sensor platform and the warning modality: robots typically rely on onboard LiDAR and generate control commands, whereas intelligent guide sticks use handheld or cane-mounted sensors and deliver tactile or voice feedback to the user. These differences, however, do not fundamentally alter the LSTM prediction pipeline. Therefore, trajectory prediction methods validated on robots can serve as a valuable reference for intelligent guide-stick algorithm design, provided that appropriate adaptations are made to sensor characteristics and feedback mechanisms. It should be noted, however, that no existing study has yet validated LSTM-based prediction models on actual intelligent guide stick hardware, and this remains an open research gap.

6.3. Positioning and Navigation Based on LSTM

LSTM has been employed in positioning and navigation for visually impaired users in multiple forms, ranging from external infrastructure-based localization to onboard sensor-based pedestrian dead reckoning to GPS/INS-integrated navigation. Bakouri et al. [71] developed a wireless ultrasound-based localization prototype that uses LSTM to jointly model speech recognition and multi-sensor signals, achieving an indoor positioning RMSE of approximately 0.192 m. While this work demonstrates LSTM's capability in processing multi-source time-series signals for accurate positioning, its reliance on an external ultrasonic network limits its practicality as a portable, self-contained intelligent guide stick.

A more practical direction for intelligent guide stick deployment is onboard-sensor-based positioning without external infrastructure. Woyano et al. [72] proposed a CNN-LSTM hybrid model that extracts sequence features from MEMS inertial measurement units (IMU) mounted on wearable devices, achieving an RMSE of 0.52 m—significantly outperforming traditional step-and-heading systems and inertial navigation systems. Shin et al. [73] further demonstrated the robustness of LSTM-based positioning under GNSS outages: their dual-LSTM architecture, trained on IMU measurements and RTK-GNSS data, improved positioning accuracy by more than 90% compared to traditional Kalman filtering during GNSS interruptions lasting over 30 s.

In GPS/INS-integrated navigation, LSTM has been adopted to address positioning degradation during GPS signal outages. Alaeiyan et al. [74] proposed an Incremental Regularized LSTM (IncRLSTM) method that models GPS signals as a multi-objective regression task and corrects INS outputs using a Faded Memory Kalman filter, reducing positioning errors by an average of 72% on real-world datasets. Wang et al. [75] introduced a transfer-learning error-state Kalman filter (TL-ESKF) approach that cascades two LSTM networks with ESKF—the first provides initial correction of vehicle state deviation, while the second refines the correction using transfer learning to accelerate training—achieving higher navigation accuracy during GPS signal interruptions and better training efficiency than high-precision machine learning methods. These studies demonstrate that LSTM can effectively compensate for positioning drift during GPS signal loss. However, their computational overhead and model complexity remain challenging for real-time deployment on intelligent guide stick hardware.

While the above studies validate LSTM's potential for accurate positioning in assistive contexts, several limitations persist. First, the positioning accuracy of LSTM-based systems remains highly sensitive to IMU sensor drift and noise, particularly during prolonged use or in environments with complex motion patterns. Second, most existing evaluations are conducted in controlled environments, with limited validation in real-world urban settings

that involve diverse terrain, crowded sidewalks, and dynamic GPS signal occlusion. Third, the computational overhead of hybrid CNN-LSTM or cascaded LSTM architectures may exceed the real-time inference budget of low-power MCUs, posing a deployment challenge for intelligent guide stick hardware. Among the reviewed studies, Woyano et al. [72] and Shin et al. [73] demonstrate that LSTM-based positioning can achieve high accuracy using only onboard sensors, without relying on external infrastructure—a desirable characteristic for deploying portable intelligent guide sticks.

6.4. Edge Computing for Intelligent Guide Stick

Edge computing offloads the deep learning model's inference task to the terminal device, providing a feasible path for the intelligent guide stick to integrate complex sensing capabilities while meeting low-power-consumption and low-latency constraints. Using an STM32F103 microcontroller and the YOLO [14] model deployed on the Jetson Xavier NX edge computing platform, Gao Shunqiang et al. [21] achieved visual perception and obstacle recognition and constructed a complete technical framework that covers key dimensions such as image processing, multi-sensor fusion, and edge computing. The system collects environmental images via the camera, uses the YOLO algorithm to detect obstacles such as pedestrians and vehicles in real time, and feeds the recognition results back to the user as voice output, demonstrating the engineering feasibility of deploying deep learning models on resource-constrained MCU-level platforms. At the more basic single-chip microcomputer level, Chen Yuhai et al. [76] built a multi-modal sensing system based on an STM32 and YOLOv5n [24], using modules such as ultrasound, GPS, camera, and inertial sensors to verify the accuracy and reliability of the visual deep learning algorithm on the embedded end. The Sense Stride system, developed by Megharsh et al. [4], further validates the hardware foundation of MCU-side intelligent sensing. The system uses the STM32H563ZI microcontroller as the core controller, integrates three VL53L0X time-of-flight ranging sensors and a pyroelectric infrared sensor, and provides hierarchical feedback through five-direction tactile vibration. The average response time is only 50 milliseconds, and the obstacle detection accuracy is 95%. The scheme is based on the independent architecture of a low-power microcontroller, enabling it to operate in real time without relying on network connections, thereby demonstrating the hardware feasibility of a low-power MCU for real-time intelligent sensing tasks. In addition, Majoni et al. [77] designed an artificial intelligent guide stick that integrates ultrasound and visual perception. They used four HC-SR04 ultrasonic sensors in conjunction with the YOLOv4-tiny model [66] on a Raspberry Pi 4B, thereby enriching the sensor fusion strategy for edge deployment.

Although the above research mainly uses the CNN model [14,66] as the core of the algorithm, their common findings indicate that STM32-class microcontrollers can handle complex deep learning inference workloads, providing a valuable reference for the hardware capabilities required for LSTM deployment on MCUs. More importantly, Section 5 has reviewed direct deployment cases of LSTM on MCU-level platforms—for instance, DeepConv LSTM [31] achieved a 21 ms inference latency and 97% accuracy on ARM Cortex-M4. It demonstrates that real-time inference of LSTM models on MCUs is supported by traceable empirical evidence, rather than merely theoretical speculation. Combined with the MCU hardware capabilities demonstrated by the edge computing studies reviewed above, the edge deployment of LSTM-based pedestrian intention prediction models on intelligent guide sticks has established preliminary hardware feasibility. Of course, hardware feasibility is only the starting point for system integration. Practical deployment still requires coordinating the LSTM prediction pipeline with existing perception, detection, and feedback modules, as well as software-level optimization for real-time data scheduling, resource management, and system architecture design.

6.5. Internet-Enabled LSTM Applications for Intelligent Guide Sticks

The Internet of Things enables intelligent guide sticks to evolve from isolated devices into connected terminals capable of remote monitoring and data upload. In this architecture, LSTM models can perform real-time inference on edge devices while uploading critical data—such as predicted collision risks, user location trajectories, and abnormal events—to cloud platforms via lightweight IoT protocols like MQTT or CoAP, enabling remote monitoring. Shireesha et al. [8] demonstrated the feasibility of this approach through a cloud-integrated smart cane system that captures images via an ESP32-CAM and uploads them to the Firebase cloud platform for object recognition, achieving 94.7% average recognition accuracy with 310 ms cloud inference latency. Building on this foundation, incorporating LSTM-based temporal prediction would allow the cloud to not only recognize current obstacles but also forecast their future trajectories, thereby providing more proactive remote warnings. In edge-cloud collaborative architectures, STM32-class MCUs can perform lightweight LSTM inference—as demonstrated by DeepConv LSTM [31], which achieves 21 ms inference latency with a 136.5 KB model size on ARM Cortex-M4—while offloading critical prediction results and anomaly events to the cloud via 4G/5G networks for model iteration and multi-user data aggregation. Furthermore, cloud platforms can support federated learning across multiple users [78,79]: cloud-aggregated and anonymized trajectory data from numerous users could enable the training of more generalizable LSTM-based pedestrian behavior prediction models, thereby alleviating the data scarcity bottleneck in intelligent guide stick scenarios.

LSTM models that rely solely on historical trajectory data cannot perceive environmental information beyond sensor range, while V2X technology can address this limitation through communication with transportation infrastructure. V2X-based intelligent guide stick solutions leverage wireless communication to enable the stick to obtain traffic light status and crosswalk information [7]. C-V2X-based approaches further enable bidirectional information exchange among traffic signals, vehicles, and intelligent guide sticks. Incorporating external information such as traffic signal phases and vehicle positions at intersections as additional input features to the LSTM model enables more accurate predictions of pedestrian crossing intentions by combining historical trajectory data with real-time traffic signal states. For example, when the LSTM model detects a pedestrian trajectory approaching an intersection and the traffic signal is about to turn red, an early warning can be triggered; when V2X signals indicate an approaching vehicle, the LSTM model can adjust collision risk assessments in advance. This integrated architecture combining LSTM-based temporal prediction with V2X-based environmental awareness has the potential to significantly enhance warning capabilities in high-risk scenarios such as intersections.

The high bandwidth and low latency characteristics of 5G networks provide communication guarantees for cloud-edge collaborative inference in intelligent guide sticks. Existing pilot projects combining 5G with edge cloud for assistive devices have demonstrated the feasibility of offloading computational tasks to edge clouds to reduce terminal burdens [80]. This approach is equally applicable to LSTM models: computationally intensive LSTM variants—including Social LSTM [10], attention-enhanced LSTM, and Social GAN [45] for multi-modal prediction—can be deployed on cloud or edge cloud platforms, with the intelligent guide stick terminal responsible only for lightweight feature extraction and real-time warnings, coordinated through 5G networks. This collaborative mode shows promise in alleviating the tension between limited terminal computational capacity and increasing model complexity. Additionally, cloud platforms support model hot-updating: when improved LSTM weights are trained on the cloud, they can be pushed to terminal devices via over-the-air (OTA) updates, enabling continuous evolution of the intelligent guide stick's predictive capabilities without requiring manual device upgrades by users.

In summary, Internet technologies are expanding the application boundaries of LSTM in intelligent guide sticks across three dimensions: IoT-cloud collaboration, V2X communication, and 5G cloud platforms. The integration of LSTM-based temporal prediction with Internet technologies has demonstrated technical potential across multiple dimensions—IOT-cloud collaboration enables remote monitoring and data backhaul, V2X introduces external environmental information to LSTM predictions, and 5G with cloud platforms supports large-scale model inference on the cloud and continuous model evolution. However, most existing studies have addressed only one of these dimensions in isolation, and systematic validation that deeply integrates LSTM-based temporal prediction with multi-level Internet technologies remains absent from the publicly available literature. Practical deployment still needs to address the tension between communication latency and real-time requirements, system robustness under discontinuous network coverage, and energy consumption optimization for cloud-edge collaborative inference—these challenges provide clear technical directions for future research.

6.6. Datasets

Since this review synthesizes findings from the literature rather than presenting new experimental results, it is necessary to examine the datasets used in the studies reviewed. The choice of dataset directly affects model performance, generalizability, and the validity of cross-study comparisons. This section provides an overview of the representative datasets used for pedestrian trajectory prediction, intention classification, and intelligent guide stick-related perception tasks, with particular attention to their sensor modalities, scale, and relevance to intelligent guide stick deployment.

For pedestrian crossing intention prediction, several publicly available datasets have become de facto benchmarks in the field. JAAD and PIE are two representative real-world in-vehicle datasets: the former comprises 346 high-resolution video clips from a dashboard-mounted camera (totaling approximately 240 h of driving footage) [81], while the latter provides over 6 h of footage with nearly 910,000 frames and 1842 annotated pedestrian samples from a first-person driving perspective [82]. Both are widely used for training and evaluating pedestrian crossing intention prediction models. To alleviate the scarcity of rare crossing events in real-world data, researchers have further constructed synthetic data resources. PedSynth, generated programmatically using the ARCANE framework on top of the CARLA simulator, consists of 947 crossing/non-crossing video clips with a total duration of 5 h and 26 min, serving as complementary material for model pre-training and data augmentation [30]. In addition, recent work has explored using PedSynth together with pre-trained intention models to automatically generate crossing pseudo-labels for real-world videos (e.g., the S2R-UDA-CP framework), showing promise in reducing manual annotation costs—further extending the utility of PedSynth as an auxiliary resource.

Beyond the aforementioned intention-oriented datasets, this review also refers to other publicly available resources for related tasks. ETH/UCY is one of the most widely used benchmarks for trajectory prediction, containing about 1500 trajectories across five scenes; its bird's-eye view differs fundamentally from the forward low-height perspective of a guide stick. BDD100K is a large-scale autonomous driving perception dataset providing 100,000 driving videos with diverse annotations including object detection and segmentation; its scene diversity offers reference value for assistive navigation, but its vehicle-mounted forward-view setting is not aligned with the guide-stick scenario. MS COCO is a commonly used image benchmark for general object detection, containing 330,000 static images with 80 object categories; it can provide pre-training material for the obstacle recognition module of a guide stick, yet its lack of temporal information makes it only partially suitable for applications that rely on continuous perception. Table 5 provides

an overview of these datasets, summarizing their key characteristics and highlighting the critical gap in publicly available data captured from the intelligent guide stick perspective.

Table 5. Representative datasets used in reviewed studies.

Dataset	Task	Sensor Modality	Scale	Key Differences from Intelligent Guide Stick Scenario
JAAD	Pedestrian crossing intention prediction	Video (forward-facing, dashboard-mounted)	346 clips, ~240 h driving footage [81]	Vehicle-mounted forward-view; higher camera height (~1.5 m); stable platform; different motion characteristics
PIE	Pedestrian intention estimation and trajectory prediction	Video (first-person forward-facing)	>6 h, 909,480 frames, 1842 pedestrian samples [82]	Vehicle-mounted forward-view; higher camera height; stable platform; larger scale
PedSynth	Pedestrian C/NC intention prediction	Synthetic RGB video (CARLA-based)	947 clips, 5 h 26 min [30]	CARLA simulator viewpoint (vehicle-mounted); synthetic domain; lacks real sensor noise, occlusion, and handheld jitter
ETH/UCY	Pedestrian trajectory prediction	Video (top-down view)	5 scenes, ~1500 trajectories [83,84]	Top-down view; no ego-motion or handheld jitter
BDD100K	Autonomous driving perception	Video (forward-facing)	100,000 videos [65]	Vehicle-mounted, stable platform, higher sensor height
MS COCO	Object detection	RGB images	330,000 images [64]	Static images only; no temporal information
In-house guide-stick data	Obstacle detection	ToF, ultrasound, camera	Small-scale prototypes [2–4,21,76,77]	Directly relevant but small scale and different annotation tasks

All the above publicly available datasets originate from non-handheld low-height perspectives—vehicle-mounted forward-view, top-down, or static images—whose sensor settings differ from the actual operating conditions of a guide stick. During use, a guide stick undergoes natural swaying and irregular swinging accompanying human gait, and the low height results in more frequent occlusions near the ground. The spatiotemporal interaction patterns learned by models trained on vehicle-mounted views may not generalize well to the unique handheld jitter, low-height occlusions, and motion irregularities characteristic of a guide stick. It should be noted that while synthetic data such as PedSynth can partially compensate for the limited number of training samples, the gap between virtual environments and real streets in terms of illumination changes, sensor noise, and dynamic occlusions remains non-negligible; moreover, since its viewpoint still follows the vehicle-mounted setting, it cannot fully replace real data collected from the low handheld perspective of a guide stick. As for automatic labeling methods like S2R-UDA-CP, their effectiveness has mainly been validated in autonomous driving scenarios, and the reliability of pseudo-labels may be constrained by the source model’s training bias toward vehicle-mounted views. Transferring these methods to the guide-stick scenario requires additional human quality inspection and domain adaptation, and they cannot yet fully substitute for systematic collection of real-world guide-stick-perspective data.

Overall, there is currently a lack of large-scale publicly available pedestrian crossing intention datasets specifically captured from the handheld low-height perspective of an intelligent guide stick. Although a few prototype studies have collected small-scale in-house guide-stick perception data, these existing data mostly focus on obstacle detection and other local perception tasks, lacking unified annotation standards and benchmark divisions for pedestrian crossing intention, and thus are not yet sufficient to support thorough algorithm training and fair comparison for this task. Bridging this gap calls for systematic efforts in real-world data collection, the formulation of intention annotation standards, and the development of simulation tools tailored to the guide-stick perspective.

6.7. Summary and Discussion

Synthesizing the five application cases reviewed above, LSTM-based prediction methods have demonstrated technical potential highly relevant to intelligent guide stick scenarios across multiple assistive dimensions. In speech guidance, CNN-LSTM fusion architectures [63] have proven effective for identifying and tracking dynamic obstacles, while hardware validation on STM32-class MCUs [4,21] confirms the technical feasibility of deploying lightweight models. In trajectory prediction and obstacle avoidance, LSTM-based collision risk prediction methods [67] have achieved validation accuracy exceeding 98% on mobile robot platforms [69]. Their methodological logic—“perceiving historical trajectories → predicting future positions → issuing early warnings”—aligns closely with the core requirement of active obstacle avoidance for intelligent guide sticks. In positioning and navigation, CNN-LSTM hybrid models [72] achieve 0.52 m positioning accuracy using only onboard IMU sensors, and maintain stable output during prolonged GPS outages [73–75]. In edge computing, STM32-series MCUs have been confirmed capable of handling deep learning inference tasks [4,21,76,77], meeting both real-time and accuracy requirements. Collectively, this evidence indicates that LSTM possesses the fundamental prerequisites for intelligent guide stick deployment in terms of algorithmic capability, lightweight potential, and engineering transferability. In terms of Internet technology collaboration, the integration of IoT-Cloud, V2X, and 5G technologies provides new capability dimensions for LSTM-based prediction, including remote monitoring, environmental information supplementation, and cloud-edge collaborative inference, thereby expanding the potential pathway for intelligent guide sticks to evolve from standalone intelligence to system-level collaboration.

However, several critical gaps remain between existing research and the deployment of practical intelligent guide stick products. At the algorithmic level, speech guidance systems [63] still focus primarily on current-frame recognition and have not yet integrated LSTM-based prediction capabilities into their warning workflows; trajectory prediction methods [21,69,70], though mature on robotic platforms, have not been adapted or tested for the specific characteristics of handheld intelligent guide sticks—sensor jitter, low mounting height, and unstable field of view; positioning and navigation systems [71–75] impose computational overhead that challenges MCU-level real-time inference, and most evaluations are conducted in controlled environments without validation in complex urban terrains; edge computing validations [21,76,77] have primarily targeted CNN models, leaving it unverified whether LSTM’s recurrent temporal prediction can coexist and co-operate with perception, detection, and feedback modules on the same resource-constrained hardware. More fundamentally, none of the existing studies have demonstrated complete end-to-end system integration—from perception through LSTM-based prediction to graded warning—on an actual intelligent guide-stick prototype with validation by visually impaired users. Future user-centered evaluations should assess not only algorithmic accuracy but also practical usability, cognitive load, and user trust, as these factors determine whether a technically accurate device is actually adopted in daily use. The scarcity of domain-specific data (as discussed in Section 6.6, the lack of large-scale public datasets captured from the intelligent guide stick perspective) further constrains model training and evaluation. Moreover, the impact of real-world factors—illumination variations, weather conditions, sensor noise—on system robustness has not been adequately tested.

7. Discussion

The analysis presented in the preceding sections indicates that deploying LSTM-based pedestrian intention prediction on intelligent guide sticks is supported by multidimensional evidence, yet several constraints remain.

In terms of constraints, hardware resource limitations are the primary challenge. A typical MCU offers only 512 KB of Flash and 256 KB of SRAM, and even after lightweight processing, the LSTM model must coexist with sensor drivers, communication stacks, and feedback modules within this limited resource budget. At the data level, there is currently no publicly available large-scale dataset captured from the intelligent guide stick perspective for pedestrian intention prediction; most existing datasets originate from autonomous driving or surveillance applications, which differ substantially in viewing angle, sensor height, and motion characteristics. At the validation level, existing studies are predominantly conducted in simulation environments or on general-purpose robotics platforms, and no study has yet achieved both complete end-to-end deployment of an LSTM-based prediction model on a real intelligent guide stick prototype and validation with visually impaired users. Future studies involving visually impaired participants must be conducted under ethically approved protocols with informed consent. These challenges are interconnected and collectively constitute the primary barrier to translating algorithmic research into practical assistive devices.

Despite these challenges, the evidence reviewed in this paper supports the feasibility of LSTM-based deployment on intelligent guide sticks. In terms of temporal modeling capability, Social LSTM [10] effectively captures pedestrian interactions through its social pooling mechanism; SR-LSTM [25] improves prediction robustness in crowded environments through spatial relationship encoding; and the enhanced Social LSTM [26] achieves a 31% reduction in collision rate across various crowd densities. In terms of lightweight deployment, DeepConv LSTM [31] achieves 21 ms inference latency, a 136.5 KB model size, and 97% accuracy on ARM Cortex-M4 after quantization; Slim LSTM [20] achieves over 93% parameter compression through architectural simplification; and the INT8 quantization scheme provides a direct reference path for MCU deployment, with its hardware feasibility confirmed by Sense Stride [4] on STM32H563ZI. Collectively, this evidence demonstrates that LSTM possesses both the temporal modeling capability required for intelligent guide stick tasks and the potential, after lightweight processing, to meet hardware resource constraints.

Based on the above analysis, future research may proceed along the following directions. First, building on the five lightweight methods reviewed in this paper, hybrid strategies that combine architectural simplification and quantization warrant further exploration to improve compression efficiency and reduce deployment risk. Second, to address the data scarcity identified above, real-world data collection from the intelligent guide stick perspective and annotation of pedestrian behavioral intentions should be prioritized, complemented by simulation tools for large-scale synthetic data generation. Third, a standardized evaluation framework specifically designed for intelligent guide stick deployment should be established to facilitate meaningful cross-study comparisons and algorithmic iteration. In addition, hardware-software co-design remains a worthwhile direction to pursue. How to effectively integrate the LSTM prediction pipeline with existing perception, detection, and feedback modules on the same resource-constrained hardware remains to be validated in practice.

Several limitations of this review should be acknowledged. The literature search is restricted to English and Chinese publications, potentially overlooking important contributions in other languages. The performance metrics reported for different lightweight methods are drawn from heterogeneous datasets, hardware platforms, and experimental configurations, which limits direct quantitative comparisons. In the absence of a standardized benchmark platform for LSTM-based prediction on intelligent guide sticks, assessments of method superiority rely on reported results rather than replicated experiments. Furthermore, given the diversity of evaluation metrics and experimental conditions, this

review does not perform a quantitative meta-analysis. These limitations do not undermine the qualitative synthesis of technical trends and mainstream approaches presented herein, but they should be considered when interpreting the comparative assessments.

8. Conclusions

This paper has provided a review of LSTM-based methods for pedestrian intention prediction and lightweight deployment for intelligent guide sticks. The analysis of existing intelligent guide stick systems [2–4,7–9] reveals a common deficiency: the lack of temporal modeling capability for dynamic obstacles, which prevents current systems from achieving proactive obstacle avoidance. The review demonstrates that LSTM [9], through its gating mechanism and architectural flexibility, effectively captures long-term dependencies in trajectory data and, compared with alternative temporal models including GRU [32], TCN [34], lightweight Transformers [36,37], and others, has relatively strong validated evidence in terms of temporal modeling capability, lightweight maturity, and MCU deployment feasibility. For resource-constrained edge deployment, five lightweight techniques—architecture simplification [20], low-rank decomposition [51–56], structured pruning [11,57,58], quantization [12,31,59], and knowledge distillation [60–62]—are examined, with hybrid strategies identified as promising pathways. Application cases across speech guidance [63], trajectory prediction-based obstacle avoidance [4,21,76,77], positioning and navigation, edge computing, and Internet collaboration [78–80] validate the diverse potential of LSTM in intelligent guide stick scenarios, yet a persistent gap remains: end-to-end deployment that integrates LSTM-based prediction with graded feedback on actual intelligent guide stick prototypes, validated with visually impaired users, has yet to be achieved. Future research should prioritize real-world deployment validation, domain-specific data collection, and hardware-software co-design to translate algorithmic advances into practical assistive devices. LSTM is expected to play a significant role in transitioning intelligent guide sticks from passive response to active prediction, ultimately improving travel safety and quality of life for visually impaired individuals.

Author Contributions: Conceptual framework construction: Y.C.; research method design: Y.C.; first draft writing: Y.C.; resource support: H.Q.; revision editing: Y.X., H.Q. and S.S.; project guidance: F.J.; project management: F.J. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the Scientific Research Foundation of Yunnan College of Business Management: Design of Intelligent Guide Stick System Based on ESP32 Chip (Project No. 2026XK23).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors would like to express their gratitude to the colleagues and researchers in the field for their valuable support during the preparation and revision of this paper. We also highly appreciate the technical support and insights on Internet of Things (IoT) technology provided by the expert team at the Institute of Visual Informatics (IVI), Universiti Kebangsaan Malaysia (UKM). Additionally, this study was supported in part by the Yunnan College of Business Management.

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

Abbreviations

The following abbreviations are used in this manuscript:

ADMM	Alternating Direction Method of Multipliers
AI	Artificial Intelligence
ARM	Advanced RISC Machine
CNN	Convolutional Neural Network
CPU	Central Processing Unit

DOI	Digital Object Identifier
FFT	Fast Fourier Transform
FPGA	Field-Programmable Gate Array
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HLS	High-Level Synthesis
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IoT	Internet of Things
LiDAR	Light Detection and Ranging
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCU	Microcontroller Unit
MEMS	Micro-Electro-Mechanical Systems
MSE	Mean Square Error
MPO	Matrix Product Operator
ONNX	Open Neural Network Exchange
PCA	Principal Component Analysis
PIR	Passive Infrared
PPO	Proximal Policy Optimization
QONNX	Quantized Open Neural Network
RGB-D	Red Green Blue-Depth
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RTK	Real-Time Kinematic
RTOS	Real-Time Operating System
SHS	Step and Heading System
SLAM	Simultaneous Localization and Mapping
SVD	Singular Value Decomposition
STM	STMicroelectronics
TCN	Temporal Convolutional Network
ToF	Time-of-Flight
TTS	Text-to-Speech
YOLO	You Only Look Once

References

1. World Health Organization. Ageing and Health Fact Sheet. Available online: <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health> (accessed on 3 June 2026).
2. 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. [CrossRef] [PubMed]
3. Gaikwad, K.N. AI Powered Camera-Based Smart Cane for Visually Impaired Navigation. Master's Thesis, California State University, Northridge, CA, USA, 2025.
4. Megharsh, L.; Uday, E.; Phalguna, K.R.; Rudragowda, K.S. Smart Walking Cane for Visually Impaired Individuals Using STM32H563ZI and Time-of-Flight Sensors with Haptic Feedback. *ITM Web. Conf.* **2025**, *79*, 01002. [CrossRef]
5. Liang, Z.H.; Gao, T.R.; Zhang, Y.R.; Ouyang, G.L.; Yang, S.Y.; Wu, Y.L. AIoT-Based Intelligent Guide Stick System. *Internet Things Technol.* **2026**, *16*, 75–77. (In Chinese) [CrossRef]
6. Veerasamy, B.; Reddy, A.S.K.; Chandu, A.; Reddy, K.S.S.; Manikanta, K.V.N.G. Edge Cloud Collaboration Intelligent Assistive Cane for Visually Impaired People. In Proceedings of the 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, 30–31 March 2023; pp. 135–139. [CrossRef]
7. Zhang, R.C.; Wang, L. Design and Implementation of Intelligent Navigation Cane Based on V2X Technology. *Comput. Knowl. Technol.* **2023**, *19*, 130–133. (In Chinese) [CrossRef]

8. Shireesha, B.; Kumar, P. Wearable Binocular Sensor System for Visual Guidance of the Blind People. *Int. J. Commun. Netw. Inf. Secur.* **2025**, *17*, 37–47.
9. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
10. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Li, F.; Savarese, S.; Fei-Fei, L. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 961–971. [CrossRef]
11. Wang, S.; Li, Z.; Ding, C.; Yuan, B.; Qiu, Q.; Wang, Y.; Liang, Y.C. C-LSTM: Enabling Efficient LSTM Using Structured Compression Techniques. In Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, CA, USA, 15–17 February 2018; pp. 11–20. [CrossRef]
12. Khandelwal, S.; Petri-Koenig, J.; Preußer, T.B.; Blott, M.; Shreejith, S. FINN-GL: Generalized Mixed-Precision Extensions for FPGA-Accelerated LSTMs. In Proceedings of the 2025 35th International Conference on Field-Programmable Logic and Applications (FPL), Antwerp, Belgium, 1–5 September 2025; pp. 46–54. [CrossRef]
13. HC-SR04 Ultrasonic Sensor Datasheet. Available online: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> (accessed on 3 June 2026).
14. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:1506.02640. [CrossRef]
15. Song, Z.; Su, W.; Chen, H.; Feng, M.; Peng, J.; Zhang, A. VSLAM Optimization Method in Dynamic Scenes Based on YOLO-Fastest. *Electronics* **2023**, *12*, 3538. [CrossRef]
16. WeWALK Smart Cane 2 User Manual, Document No. 230917-V0.1-V2. Available online: <https://support.wewalk.io/en/article/wewalk-smart-cane-2-users-manual> (accessed on 3 June 2026).
17. Glide Smart Guide Official FAQ Page. Available online: <https://glidance.io/frequently-asked-questions> (accessed on 3 June 2026).
18. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-Time Loop Closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278. [CrossRef]
19. STM32L562CE Microcontroller Datasheet. Available online: <https://www.st.com/resource/en/datasheet/stm32l562ce.pdf> (accessed on 3 June 2026).
20. Akandeh, A.; Salem, F.M. Slim LSTM Networks: LSTM_6 and LSTM_C6. In Proceedings of the 62nd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, USA, 4–7 August 2019; pp. 630–633. [CrossRef]
21. Gao, S.Q.; Wang, Z.W.; Zhao, W.Z.; Li, N. Application Research of Intelligent Blind Guide Cane System Based on Visual Perception. *Internet Things Technol.* **2025**, *15*, 147–150. (In Chinese) [CrossRef]
22. Huang, S.; Wei, R.; Lian, L.; Lo, S.; Lu, S. Review of the Application of Neural Network Approaches in Pedestrian Dynamics Studies. *Heliyon* **2024**, *10*, e30659. [CrossRef] [PubMed]
23. Yin, C.; Cecotti, M.; Auger, D.J.; Fotouhi, A.; Jiang, H. Deep-Learning-Based Vehicle Trajectory Prediction: A Review. *IET Intell. Transp. Syst.* **2025**, *19*, e70001. [CrossRef]
24. Jocher, G. YOLOv5 by Ultralytics, Version 7.0. 2022. Available online: <https://github.com/ultralytics/yolov5> (accessed on 10 July 2026).
25. Zhang, P.; Ouyang, W.; Zhang, P.; Xue, J.; Zheng, N. SR-LSTM: State Refinement for LSTM towards Pedestrian Trajectory Prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12085–12094. [CrossRef]
26. Alia, A.; Chraïbi, M.; Seyfried, A. Social LSTM with Dynamic Occupancy Modeling for Realistic Pedestrian Trajectory Prediction. *arXiv* **2025**, arXiv:2511.09735. [CrossRef]
27. Yan, Y.; Zhang, P.; Du, C.; Wang, H.; Wang, H.; Pi, D.; Chen, Y.H. Vehicle Trajectory Prediction for Autonomous Driving Applications: State-of-the-Art Review, Research Challenges, and Future Directions. *Automot. Innov.* **2026**, *9*, 263–295. [CrossRef]
28. Chen, H.; Zhang, X.; Yang, W.; Lin, Y. A Data-Driven Stacking Fusion Approach for Pedestrian Trajectory Prediction. *Transp. B Transp. Dyn.* **2023**, *11*, 548–571. [CrossRef]
29. Dasgupta, S.; Saha, P.; Roy, A.; Sen, J. Predicting Road Crossing Behaviour Using Pose Detection with Sequence Modelling. In Proceedings of the 2025 IEEE 6th India Council International Subsections Conference (INDISCON), Rourkela, India, 21–23 August 2025; pp. 1–6. [CrossRef]
30. Riaz, M.N.; Wielgosz, M.; Romera, A.G.; López, A.M. Synthetic Data Generation Framework, Dataset, and Efficient Deep Model for Pedestrian Intention Prediction. In Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2023; pp. 2742–2749. [CrossRef]
31. Zhou, H.; Zhang, X.; Feng, Y.; Zhang, T.; Xiong, L. Efficient Human Activity Recognition on Edge Devices Using DeepConv LSTM Architectures. *Sci. Rep.* **2025**, *15*, 13830. [CrossRef] [PubMed]
32. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734. [CrossRef]

33. Anargyarahman, N.B.; Timoteo, A.; Benyamin, F.R.; Iskandar; Yuliana, H. Analysis of Accuracy and Efficiency Comparison Between LSTM and GRU Models for 5G Network User Mobility Prediction in Jabodetabek. In Proceedings of the 2025 11th International Conference on Wireless and Telematics (ICWT), Lampung, Indonesia, 3–4 July 2025; pp. 1–6. [\[CrossRef\]](#)
34. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271. [\[CrossRef\]](#)
35. Mohamed, A.; Qian, K.; Elhoseiny, M.; Claudel, C. Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 14424–14432. [\[CrossRef\]](#)
36. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:2004.05150. [\[CrossRef\]](#)
37. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:2001.04451. [\[CrossRef\]](#)
38. Zhang, F.; Leong, L.V.; Yen, K.S. A lightweight pedestrian multi-proposal trajectory prediction method based on dynamic scene information module. *Measurement* **2026**, *258*, 119076. [\[CrossRef\]](#)
39. Kim, S.; Jafarpourmarzouni, R.; Neff, C.; Tabkhi, H.; Katariya, V.A. EdgeVTP: Exploration of Latency-Efficient Trajectory Prediction for Edge-Based Embedded Vision Applications. *arXiv* **2026**, arXiv:2604.16783. [\[CrossRef\]](#)
40. Riaz, M.N.; Wielgosz, M.; Xie, C.; López, A.M. PedGT: Enhancing Pedestrian Intention Prediction Using a Skeleton-Based Graph-Transformer. In Proceedings of the 2025 IEEE Intelligent Vehicles Symposium (IV), Cluj-Napoca, Romania, 22–25 June 2025; pp. 293–300. [\[CrossRef\]](#)
41. Yang, B.; Liu, X.; Wang, Z.; Yu, S.; Rao, Y. Attention-Enhanced CNN-LSTM for Ship Intention Prediction Using AIS Data. In Proceedings of the 2025 IEEE 8th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Guiyang, China, 15–17 August 2025; pp. 685–691. [\[CrossRef\]](#)
42. Li, Z.; Sun, H.; Xiao, D.; Xie, H. Hybrid Kalman Recurrent Neural Network for Vehicle Trajectory Prediction. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 2519814. [\[CrossRef\]](#)
43. Ekambaram, V.; Jati, A.; Dayama, P.; Mukherjee, S.; Nguyen, N.H.; Gifford, W.M.; Reddy, C.; Kalagnanam, J. Tiny Time Mixers (TTMs): Fast Pre-Trained Models for Enhanced Zero/Few-Shot Forecasting of Multivariate Time Series. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 74147–74181. [\[CrossRef\]](#)
44. Ma, A.; Luo, D.; Sha, M. MixLinear: Extreme Low Resource Multivariate Time Series Forecasting with 0.1K Parameters. *arXiv* **2024**, arXiv:2410.02081. [\[CrossRef\]](#)
45. Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; Alahi, A. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. *arXiv* **2018**, arXiv:1803.10892. [\[CrossRef\]](#)
46. Sun, L.; Yan, Z.; Mellado, S.M.; Hanheide, M.; Duckett, T. 3DOF Pedestrian Trajectory Prediction Learned from Long-Term Autonomous Mobile Robot Deployment Data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1–8. [\[CrossRef\]](#)
47. Li, D.; Feng, Y.; Krueger, R. Eye Gaze-Informed and Context-Aware Pedestrian Trajectory Prediction in Shared Spaces with Automated Shuttles: A Virtual Reality Study. *arXiv* **2026**, arXiv:2603.19812. [\[CrossRef\]](#)
48. Sabah, R.; Lam, M.C.; Qamar, F.; Zaidan, B.B. A BiLSTM-Based Feature Fusion with CNN Model: Integrating Smartphone Sensor Data for Pedestrian Activity Recognition. *IEEE Access* **2024**, *12*, 142957–142978. [\[CrossRef\]](#)
49. Tao, C.; Jiang, Q.; Duan, L.; Luo, P. Dynamic and Static Context-Aware LSTM for Multi-Agent Motion Prediction. In *Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 12366, pp. 547–563. [\[CrossRef\]](#)
50. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150. Available online: <https://aclanthology.org/P11-1015/> (accessed on 3 June 2026).
51. Liu, C.; Ren, H.; Li, G.; Ren, H.; Liang, X.; Yang, C.; Gui, W. Singular Value Decomposition-Based Lightweight LSTM for Time Series Forecasting. *Future Gener. Comput. Syst.* **2026**, *174*, 107910. [\[CrossRef\]](#)
52. Dai, R.; Li, L.; Yu, W. Fast Training and Model Compression of Gated RNNs via Singular Value Decomposition. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–7. [\[CrossRef\]](#)
53. Prabhavalkar, R.; Alsharif, O.; Bruguier, A.; McGraw, L. On the Compression of Recurrent Neural Networks with an Application to LVCSR Acoustic Modeling for Embedded Speech Recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 5970–5974. [\[CrossRef\]](#)
54. Gao, Z.F.; Sun, X.; Gao, L.; Li, J.; Lu, Z.Y. Compressing LSTM Networks by Matrix Product Operators. *arXiv* **2020**, arXiv:2012.11943. [\[CrossRef\]](#)
55. Pan, Y.; Xu, J.; Wang, M.; Ye, J.; Wang, F.; Bai, K.; Xu, Z. Compressing Recurrent Neural Networks with Tensor Ring for Action Recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4683–4690. [\[CrossRef\]](#)

56. Boddu, Y.; Manimaran, A. Maximizing Forecasting Precision: Empowering Multivariate Time Series Prediction with QPCA-LSTM. *Comput. Econ.* **2025**, *66*, 2755–2790. [\[CrossRef\]](#)
57. Li, Z.; Ding, C.; Wang, S.; Wen, W.; Zhuo, Y.; Liu, C.; Qiu, Q.; Xu, W.; Lin, X.; Qian, X.; et al. E-RNN: Design Optimization for Efficient Recurrent Neural Networks in FPGAs. In Proceedings of the 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), Washington, DC, USA, 16–20 January 2019; pp. 69–80. [\[CrossRef\]](#)
58. Han, S.; Kang, J.; Mao, H.; Hu, Y.; Li, X.; Li, Y.; Xie, D.; Luo, H.; Yao, S.; Wang, Y.; et al. ESE: Efficient Speech Recognition Engine with Sparse LSTM. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, CA, USA, 22–24 February 2017; pp. 75–84. [\[CrossRef\]](#)
59. Umuroglu, Y.; Fraser, N.J.; Gambardella, G.; Blott, M.; Leong, P.; Jahre, M.; Vissers, K. FINN: A Framework for Fast, Scalable Binarized Neural Network. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, CA, USA, 22–24 February 2017; pp. 65–74. [\[CrossRef\]](#)
60. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531. [\[CrossRef\]](#)
61. Li, Y.; Yang, C.; Zeng, H.; Dong, Z.; An, Z.; Xu, Y.; Tian, Y.; Wu, H. Frequency-Aligned Knowledge Distillation for Lightweight Spatiotemporal Forecasting. *arXiv* **2025**, arXiv:2507.02939. [\[CrossRef\]](#)
62. Cheng, X.; Chaw, J.K.; Sahrani, S.; Ang, M.C.; Gunasekaran, S.S.; Mahmoud, M.A.; Zaman, H.B.; Zhao, Y.; Ren, F. An Adaptive Dual Distillation Framework for Efficient Remaining Useful Life Prediction. *Complex Intell. Syst.* **2025**, *11*, 253. [\[CrossRef\]](#)
63. Kambham, S.P.; Varman, R.; Ramya, T.S.V.; Eniyan, M.K.; Tangavel, S.K.; Somasundaram, K.; Shanmugam, S.K. Voice Assistance to Blind People by Detection of Surrounding Objects, Activities, Humans. In Proceedings of the 2024 International Conference on Computing and Intelligent Reality Technologies (ICCIRT), Coimbatore, India, 5–6 December 2024; pp. 16–21. [\[CrossRef\]](#)
64. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8693, pp. 740–755. [\[CrossRef\]](#)
65. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2636–2645. [\[CrossRef\]](#)
66. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
67. Poulet, O.; Guinand, F.; Guérin, F. Deep Reinforcement Learning with Anticipatory Reward in LSTM for Collision Avoidance. *arXiv* **2025**, arXiv:2508.07941. [\[CrossRef\]](#)
68. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. *arXiv* **2015**, arXiv:1509.02971. [\[CrossRef\]](#)
69. Mulás-Tejeda, E.; Gómez-Espinosa, A.; Escobedo Cabello, J.A.; Cantoral-Ceballos, J.A.; Molina-Leal, A. Implementation of a Long Short-Term Memory Neural Network-Based Algorithm for Dynamic Obstacle Avoidance. *Sensors* **2024**, *24*, 3004. [\[CrossRef\]](#) [\[PubMed\]](#)
70. Zhang, Q.; Ma, W.; Zheng, Q.; Zhai, X.; Zhang, W.; Zhang, T.; Wang, S. Path Planning of Mobile Robot in Dynamic Obstacle Avoidance Environment Based on Deep Reinforcement Learning. *IEEE Access* **2024**, *12*, 189136–189152. [\[CrossRef\]](#)
71. Bakouri, M.; Alyami, N.; Alassaf, A.; Waly, M.; Alqahtani, T.; AlMohimeed, I.; Alqahtani, A.; Samsuzzaman, M.; Ismail, H.F.; Alharbi, Y. Sound-Based Localization Using LSTM Networks for Visually Impaired Navigation. *Sensors* **2023**, *23*, 4033. [\[CrossRef\]](#) [\[PubMed\]](#)
72. Woyano, F.; Park, S.; Blagovest Iordanov, V.; Lee, S. A Hybrid CNN-LSTM-Based Approach for Pedestrian Dead Reckoning Using Multi-Sensor-Equipped Backpack. *Electronics* **2023**, *12*, 2957. [\[CrossRef\]](#)
73. Shin, Y.; Lee, C.; Jung, D.; Kim, E. Long Short-Term Memory Network for INS Positioning During GNSS Outages: A Preliminary Study on Simple Trajectories. *J. Position. Navig. Timing* **2024**, *13*, 137–147. [\[CrossRef\]](#)
74. Alaeiyan, H.; Mosavi, M.R.; Ayatollahi, A. Improving the Performance of GPS/INS Integration during GPS Outage with Incremental Regularized LSTM Learning. *Alex. Eng. J.* **2024**, *105*, 137–155. [\[CrossRef\]](#)
75. Wang, Y.; Luo, D.; Wang, J.; Qi, C.; Chen, Z.; Yan, X. TL-ESKF: An Information Fusion Method for INS/GPS Integrated Navigation Considering Driving State Deviation. *Expert Syst. Appl.* **2025**, *287*, 128168. [\[CrossRef\]](#)
76. Chen, Y.H.; Wen, A.H.; Zhuang, X.X.; Zheng, X.B. Research on Intelligent Obstacle Detection Algorithm Based on Deep Learning and Microcontroller. *Robot. Ind.* **2025**, 90–99. (In Chinese) [\[CrossRef\]](#)
77. Majoni, L.; van Aardt, S.; Smith, F.; Hatefi, S. Camera and Ultrasonic Sensor Fusion for Electronic Travel Aid. In *MATEC Web of Conferences*; EDP Sciences: Les Ulis, France, 2025; Volume 417, p. 10005. [\[CrossRef\]](#)
78. Atlam, M.; Attiya, G.M.; Elrashidy, M. Federated Learning: A Comprehensive Survey of Applications, Challenges, and Emerging Research Frontiers. In Proceedings of the 2025 4th International Conference on Electronic Engineering (ICEEM), Menouf, Egypt, 4–5 October 2025; pp. 1–8. [\[CrossRef\]](#)

79. Karaali, O.; Farag, H.M.; Dosen, S.; Stefanovic, C. Enabling Next-Generation Cloud-Connected Bionic Limbs Through 5G Connectivity. *IEEE Internet Things Mag.* **2025**, *8*, 45–51. [[CrossRef](#)]
80. Akbarian, F.; Vlaeminck, R.; Verheijen, J.; Aminifar, A.; Aminifar, A. Federated Learning for Obstacle Detection to Assist the Visually-Impaired Using Augmented Reality. In Proceedings of the 2025 28th Euromicro Conference on Digital System Design (DSD), Salerno, Italy, 10–12 September 2025; pp. 486–490. [[CrossRef](#)]
81. Rasouli, A.; Kotseruba, I.; Tsotsos, J.K. Are They Going to Cross? A Benchmark Dataset and Baseline for Pedestrian Crosswalk Behavior. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 206–213. [[CrossRef](#)]
82. Rasouli, A.; Kotseruba, I.; Kunic, T.; Tsotsos, J. PIE: A Large-Scale Dataset and Models for Pedestrian Intention Estimation and Trajectory Prediction. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6261–6270. [[CrossRef](#)]
83. Pellegrini, S.; Ess, A.; Schindler, K.; Van Gool, L. You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking. In Proceedings of the 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009; pp. 261–268. [[CrossRef](#)]
84. Lerner, A.; Chrysanthou, Y.; Lischinski, D. Crowds by Example. *Comput. Graph. Forum* **2007**, *26*, 655–664. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.