Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Customized Learning Paths: Machine Learning based Developing Intelligent Tutoring Systems for Elementary Students in India

P. Raja

rajsella.visys@gmail.com

Lovely Professional University

Gaganpreet Kaur Marwah

Lovely Professional University

Article

Keywords: Intelligent Tutoring Systems, Elementary Education, User-centered Learning, Personalized Education, Educational Technology

Posted Date: July 11th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6741072/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Customized Learning Paths: Machine Learning based Developing Intelligent Tutoring Systems for Elementary Students in India

P.Raja[1*], Dr Gaganpreet Kaur Marwah[2]

[1]rajsella.visys@gmail.com, [2]gaganpreet.marwah@lpu.com

[1,2]School of Electronics and Electrical Engineering, Lovely Professional Engineering, Phagwara, Punjab, India.

Abstract

In today's rapidly evolving educational landscape, the integration of artificial intelligence (AI) into learning environments has become increasingly vital—particularly in regions where access to personalized instruction is limited. Existing literature supports the effectiveness of learner-centered approaches, especially through the development of Intelligent Tutoring Systems (ITS), with a growing emphasis on enhancing education for children in the Indian context. Primary education continues to face significant challenges: rote memorization is often emphasized over conceptual understanding, classroom sizes are large, and opportunities for experiential learning remain limited. To address these issues, this study proposes the development of an interactive and accessible ITS designed to teach primary-level mathematics and Tamil to children aged 5 to 10 years in a more engaging and individualized manner. The system is implemented using a Raspberry Pi module and was tested with a group of 34 children. It aims to offer personalized learning experiences, foster motivation, and support measurable improvements in learning outcomes. The ITS allows children to progress at their own pace and incorporates adaptive learning strategies. Learners are presented with basic questions aligned to previously taught content; correct responses receive positive reinforcement, while incorrect answers prompt context-sensitive hints. Future work will focus on further validating the system through detailed assessment of student learning progress and capability development.

Keywords

Intelligent Tutoring Systems, Elementary Education, User-centered Learning, Personalized Education, Educational Technology.

1. Introduction:

India's elementary education system faces persistent challenges such as large class sizes, limited educational resources, and a lack of personalized instruction. Creating engaging, student-centered learning environments is critical to overcoming these gaps. Today, learner-focused educational settings require the integration of modern technologies and innovative pedagogical approaches.

Among these, **digital learning environments** have emerged as powerful tools for enhancing students' motivation and interest in learning.

The term *Intelligent Tutoring System* (ITS) broadly refers to computer-based programs that possess some degree of artificial intelligence and are designed to support learning. The ITS concept evolved from earlier models of *Computer-Assisted Instruction* (CAI), which typically used hardcoded, frame-based, or hypertext systems to deliver content. Unlike CAI, ITS platforms are capable of adapting instruction based on learners' needs and performance.

Traditional educational systems, particularly in India, often struggle to address the diverse learning styles and readiness levels of students. As noted in [2], a significant proportion of children are taught by a single teacher who may not be equipped to meet their individual learning needs. This mismatch can lead to disengagement and hinder academic achievement.

While India's conventional teaching methods have strengths, they are also marked by drawbacks such as rote memorization, overcrowded classrooms, limited access to resources, outdated curricula, and a lack of practical, hands-on learning opportunities. In rural areas, these challenges are further compounded by factors such as low literacy rates, inadequate infrastructure, unqualified educators, and poorly designed curricula. Encouraging student participation—an essential component of effective learning [1]—is often difficult under these circumstances.

One promising strategy to address these issues involves using **language-based and game-oriented instructional models**, which can sustain students' interest while delivering core educational content [5]. Despite this, critical questions remain about how to provide scalable, individualized instruction and instill in students a deeper understanding of the value of education.

Tutoring programs can play a significant role in developing the learning and study skills necessary for long-term academic success. The benefits of intelligent tutoring systems include one-on-one attention, improved academic performance, self-paced learning, the ability to target individual strengths and weaknesses, and a low-stress environment conducive to learning.

Drawing on both our own teaching experience and insights from the literature, we identify **key design principles** that contribute to the effectiveness of educational systems. Educators should consider these principles when developing new instructional content or courses [8]. According to a widely cited study [19], personalized, one-on-one instruction can be up to **98% more effective**

than traditional teaching approaches, enabling students to achieve performance gains of two standard deviations above the norm.

The design of an ITS can be informed by cognitive models that define the specific skills or competencies the system seeks to develop. Common types include **rule-based** and **constraint-based** models [13]. Understanding individual learner characteristics is also vital for designing effective web-based learning environments [14]. In addition to user-friendly interfaces, successful online instruction benefits from multimedia integration, adaptive learning strategies, and sophisticated reasoning techniques.

In this paper, we present our enhancement of the **Intelligent Tutoring System paradigm**, a well-established and widely studied framework for building intelligent educational technologies. The strength of the ITS model lies in its proven success across multiple domains, including but not limited to medicine [17]. ITS platforms employ various instructional strategies to deliver content in a personalized manner aligned with students' preferred learning styles, yielding promising and empirically supported outcomes [19].

Furthermore, we hypothesize that successful ITS design depends on the inclusion of flexible teaching strategies, learner models based on prior knowledge and abilities, and competency-specific instructional approaches. We also propose that instructors, like other domain experts, rely on cognitive models formed through experience. They draw upon past teaching instances, select those most relevant to current contexts, and adapt them accordingly—contributing to an evolving repository of instructional strategies.

The development of critical life skills in children, including problem-solving, measurement, and spatial awareness, is facilitated by early math education. Youngsters who grasp more difficult arithmetic ideas from an early age are more likely to succeed academically.

2. Scope of the research:

The scope of this research is to explore, design, and develop Intelligent Tutoring Systems (ITS) tailored for children aged 5 to 10 years within the Indian educational context. The study focuses on identifying existing tutoring system frameworks and adapting them to meet the diverse learning needs of young learners, particularly in underserved settings. Key areas of development include domain models for subjects such as Mathematics and Tamil, tutoring strategies suitable for early

education, and student models that reflect the cognitive and emotional profiles of primary-level learners. Special emphasis is placed on personalized learning experiences, with the systems designed to support junior school pupils, children in NGO-run schools, and those from underprivileged backgrounds such as urban slums.

3. Related work:

Over recent years, numerous Intelligent Tutoring Systems (ITS) have been developed by researchers, evolving to incorporate a wide range of features and applications. These systems span various educational domains and learning levels. For instance, ITS for ZOSMAT is a web-based interface designed to enhance learning [4], while problem-based learning (PBL) approaches have been employed to develop concept-driven instructional content [5]. The rise of conversational AI has led to the creation of intelligent tutoring chatbots [6], and in the field of language learning, ITS has been used to simulate personality-aware student interactions [7].

Other notable implementations include Smart Tutor, a web-based platform for adult education [8], and ITS applications in teaching mathematics [10] and fundamental physics concepts [12]. Systems using Bayesian networks and fuzzy logic have been employed to predict and enhance student academic performance [14], while subject-specific tutors such as those for circuit analysis [15] have also been developed.

Further advancements include ABITS, an ITS for distance learning [17], and AI4ALL, which focuses on AI education for K–12 students [18]. Dialogue-based systems like AutoTutor leverage mixed-initiative conversation techniques to guide learners [11], and ROBOMATH and robot-based tutoring systems have been developed for subjects such as multiplication and mathematics with small robots [20], [22], [23].

In the domain of reading and comprehension, ITS tools have been designed for K–12 students [24]. Several ITS platforms specifically target mathematics education at different levels—for example, tools for sixth-grade students [25], ITS for arithmetic and fractions [27], and web-based math education in German lower secondary schools [28]. Some systems focus on closing learning gaps, such as a math gap reducer for after-school programs [29], while others aim to assist high school students with chatbot-driven problem solving [30]. Additionally, efforts have been made to develop systems for personalized math instruction [30].

4. ITS structure for Mathematics and Tamil tutor

The structure of the Intelligent Tutoring System (ITS) is illustrated in Figure 1. A classic ITS framework typically comprises four key components: the **domain model**, **student model**, **teaching (or pedagogical) model**, and the **learning environment or user interface**. The degree of intelligence embedded within each component can vary significantly, resulting in considerable diversity among ITS implementations.

The primary objectives of an ITS include providing **personalized learning**, **real-time feedback**, and **dynamic knowledge tracing** to enhance student engagement and learning outcomes [7]. While the overall architecture of ITSs may vary depending on specific design requirements, most systems share a core structure built around three essential models: domain, student, and teaching [3].

- The **domain model** represents the instructional content, including the subject matter, the structure of the knowledge to be taught, and the relationships between various concepts.
- The **student model** maintains detailed information about individual learners, such as their prior knowledge, learning goals, preferences, progress, and misconceptions. This model plays a central role in enabling the system to adapt dynamically to each learner's unique needs.
- The **teaching model** (or pedagogical module) determines the instructional strategies and decision-making logic used to guide the learning process based on data from the domain and student models.

The **user interface** or **learning environment** serves as the medium through which learners interact with the system. It integrates the functionalities of the three core models, enabling the system to deliver adaptive content and track learner performance effectively. The **student model**, in particular, is instrumental in monitoring learning behavior and informing system responses, thereby contributing significantly to the overall intelligence of the ITS.

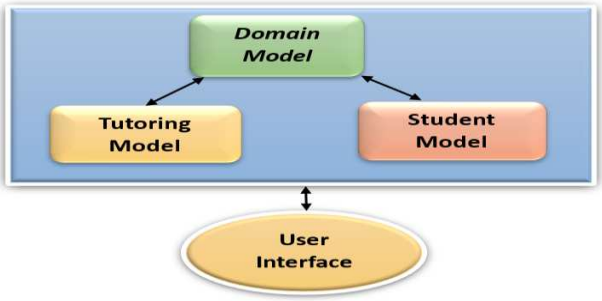

Figure 1: Structure of the ITS.

4.1 Domain model

The domain model in the proposed Intelligent Tutoring System (ITS) encompasses a wide range of essential and engaging topics related to early mathematics education and Tamil language learning. The system is designed to enhance foundational knowledge in young learners, specifically targeting children aged 5 to 10 years.

In the Mathematics domain, the ITS includes modules focused on:

- Counting and number recognition
- Shape and pattern recognition
- Basic arithmetic operations such as addition and subtraction
- Measurement concepts and comparative reasoning
- Introduction to money concepts and financial literacy
- Foundational understanding of fractions

These modules are structured to build a solid mathematical foundation through interactive, age-appropriate activities. The system employs visual aids, audio prompts, and interactive elements to support intuitive and engaging learning experiences.

In the Tamil language domain, the ITS offers modules aimed at:

- Learning Tamil Ezhuthugal (alphabets)
- Developing basic reading skills
- Practicing writing through structured exercises
- Enhancing vocabulary using an interactive word game called Varthai Vilaiyattu

This game-based learning activity is designed to make vocabulary acquisition both enjoyable and effective. The Tamil modules aim to foster phonemic awareness, orthographic understanding, and fluency through practice-based and interactive approaches.

The overall design of the ITS supports independent learning, instant feedback, and adaptive instruction based on the learner's progress. By aligning content with the cognitive and academic needs of young learners, the system promotes individualized learning pathways and enhances overall educational outcomes.

4.2 Student model

To access the course materials and complete the assigned activities, each student is required to create a personalized user profile within the Intelligent Tutoring System (ITS). The student profile stores critical data including the student's name, last session date, unique student ID, current score, total (cumulative) score, selected difficulty level, and problem number for each learning session.

- The current score reflects the student's performance at the currently selected difficulty level.
- The total score aggregates performance across all levels, providing a comprehensive view of the student's overall progress.
- The difficulty level and problem number fields help the system resume the session from the exact point where the student last left off.

All student-related data, including activity logs and progress metrics, are securely stored and managed using a SQLite3 database. This allows for efficient data retrieval and tracking of individual learning trajectories, thereby enabling personalized learning experiences and data-driven feedback.

4.3 Teaching model

The teaching model in the Intelligent Tutoring System (ITS) is designed as a multi-model framework, in alignment with the structure described in the domain model. Pupils are given the flexibility to choose the learning module based on their interests and needs, allowing the system to adapt to their preferred learning path and pace.

For the mathematics teaching model, the instructional design emphasizes student engagement, conceptual understanding, and the development of problem-solving skills. The teaching process is structured into the following stages:

4.3.1 Concept Introduction

Each session begins with a clear and concise explanation of the mathematical concept to be taught. To enhance comprehension and relevance, real-world examples and visual aids are incorporated to illustrate the importance and applicability of the concept.

4.3.2 Interactive Exploration

Students engage in interactive demonstrations or hands-on activities to explore the concept further. Collaborative discussions and group-based problem-solving tasks are facilitated to foster critical thinking, communication, and peer learning.

4.3.3 Application and Problem-Solving

Learners are presented with contextual problems or real-life scenarios that require applying the concept in diverse settings. During this phase, the focus is on guiding students through logical reasoning, structured problem-solving methods, and developing confidence in applying learned knowledge.This structured teaching model supports adaptive learning, ensuring that students not only acquire theoretical understanding but also build the skills necessary to apply mathematical concepts in meaningful ways.

4.4 User interface:

Designing a user interface (UI) for a mathematics tutoring system requires careful consideration to ensure that the interface is intuitive, engaging, and easy to navigate, especially for young learners. The UI is developed to serve as the frontend access point through which students interact with the tutoring modules.

The interface allows learners to seamlessly navigate between lessons, practice exercises, and interactive activities. It incorporates essential features such as:

- Hints and guided assistance to support learners when they encounter difficulties,
- Real-time feedback to reinforce correct answers or guide improvement,
- Motivational prompts (e.g., encouragement messages, badges, or progress animations),

- Score tracking to record the learner's performance for each session.

All user interactions and performance data are logged, allowing the system to provide adaptive learning experiences based on the student's progress and behavior. The UI is designed to be age-appropriate, with colorful visuals, audio cues, and minimal text to cater to early learners and maintain their engagement throughout the learning process.

5. Results and discussion

The integration of the Intelligent Tutoring System (ITS), as illustrated in *Figure 2*, is implemented using a Raspberry Pi as the primary processing unit. The system architecture comprises four core components: a processing unit, input unit, output unit, and the physical body of the device. This setup enables a compact, cost-effective, and efficient platform suitable for one-on-one personalized instruction. The ITS supports various educational modules tailored for early learners. In the Tamil language module, children can engage in learning alphabets (Ezhuthugal), reading, writing, and participate in memory-based word games. Similarly, the mathematics module includes interactive activities such as number recognition, arithmetic operations, and pattern identification. These modules aim to enhance the child's foundational knowledge through interactive, adaptive, and feedback-driven learning experiences.

The system records the scores and feedback for each child, allowing for individualized progress tracking. In this study, 34 children from GET Excellence Matriculation and CBSE School participated in the evaluation. Based on the scores obtained in different modules, the system provides valuable data to help tutors assess the extent of learning and conceptual understanding acquired by each pupil.



Figure 2: Intelligent Tutoring system with pupils

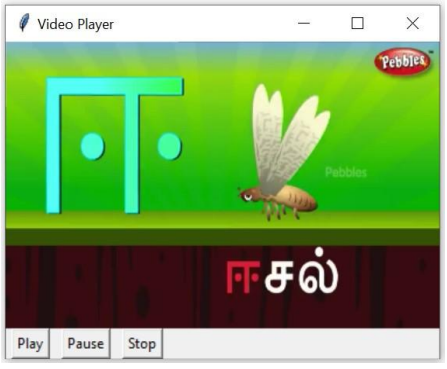

Figure 3: Tamil vowels learning module

The Tamil letters learning module is equipped with a user-friendly interface, enabling pupils to select lesson plans and engage in visual learning experiences. The design emphasizes simplicity and interactivity, allowing children to explore Tamil alphabets (Uyir, Mei, and Uyirmei Ezhuthugal) through colorful visual content and audio support. To enhance usability, the interface includes essential control features, such as pause and stop buttons, enabling pupils to manage their learning sessions at their own pace. These controls are clearly displayed on the main interface, as illustrated in Figure 3.

Following the same design principles, similar user interfaces have been developed for other modules in this research, including those for mathematics learning, reading and writing skills, and vocabulary games. Each module is structured to promote self-paced, interactive, and engaging learning experiences tailored to the developmental needs of children aged 5 to 10.



Figure 4: Interface to Guess the number

The Guess-the-Number interface, illustrated in Figure 4, is designed to help pupils identify and spell numbers correctly in an interactive and engaging manner. In this activity, a number is displayed visually, and pupils are prompted to type the correct spelling of the number using the input field provided.

Upon submission, the system evaluates the input. If the spelling matches the correct answer, the system responds with a positive reinforcement message, such as “Correct! Well done!”. If the answer is incorrect, the system provides constructive feedback, displaying the message “Sorry, wrong answer!”. This module aims to reinforce number recognition, spelling accuracy, and typing skills, while also keeping young learners motivated through instant feedback and encouragement.

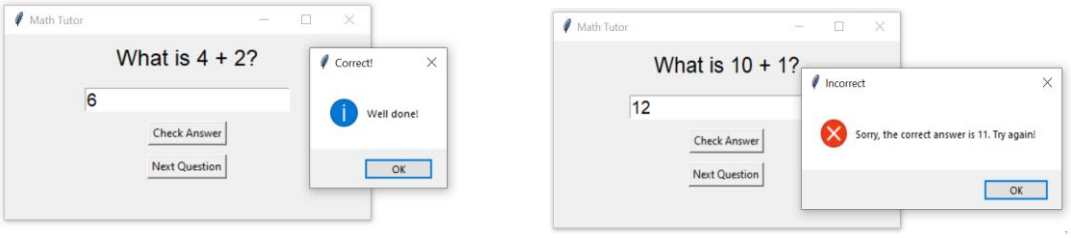

Figure 5: Interface to perform the addition (a) with the right answer and (b) with the wrong answer

As illustrated in Figure 5, the Addition Practice Interface allows pupils to solve basic addition problems interactively. A question is displayed on the screen, prompting the pupil to input the correct numerical answer using the provided input field.

Once the answer is submitted, the system compares the user's response with the correct result. If the answer is correct, the interface displays a positive reinforcement message, such as “Well done!”. In the case of an incorrect response, the system provides constructive feedback, displaying “Sorry” along with the correct answer to guide learning. This interactive feature is designed to enhance arithmetic skills, support self-paced practice, and encourage active learning through immediate feedback.

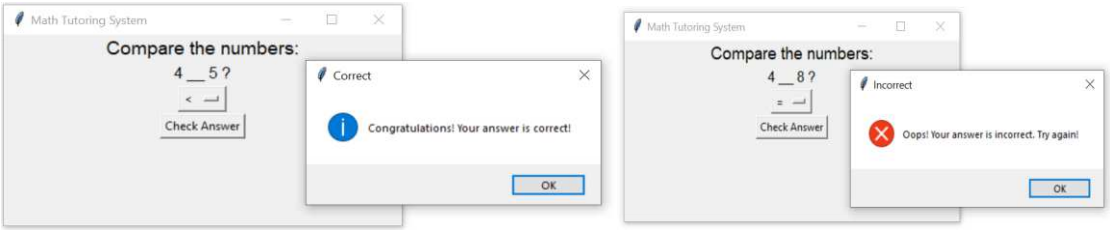

Figure 6: a) Comparison of two numbers with right answer b) Comparison of two numbers with wrong answer

As depicted in Figure 6, the Number Comparison Interface enables pupils to compare two randomly generated numbers using appropriate mathematical symbols. The system presents a pair of numbers as a question on the screen, and the pupil is required to select the correct comparison symbol (e.g., <, =, >) from a drop-down menu.

Once the selection is made, the pupil clicks on the "Check Answer" button. If the chosen symbol correctly represents the relationship between the two numbers, the system provides positive feedback with the message, “Congratulations! Your answer is correct.” Otherwise, the system displays the message, “Oops! Your answer is incorrect. Try again!” to encourage further attempts.

This interface is designed to develop number sense, reinforce comparative reasoning, and promote active engagement through immediate and formative feedback.

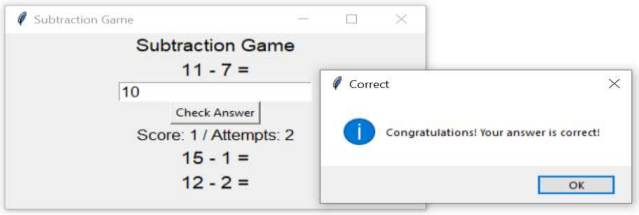

Figure 7: Interface to verify the subtraction

As shown in Figure 7, the Subtraction Interface is designed to help pupils practice basic subtraction problems in an interactive manner. A subtraction question is presented on the screen, and pupils are required to input their answer in the designated field and click on the "Check Answer" button.

If the entered answer is correct, the system automatically advances to the next question, while simultaneously updating the pupil's score and recording the number of attempts made. In the case of an incorrect answer, the system allows the pupil to try again, thereby reinforcing learning through repeated practice and feedback. This interface supports incremental learning, performance tracking, and self-paced engagement, making it a valuable component of the intelligent tutoring system for foundational mathematics.



Figure 8: Interface to verify the colors

As illustrated in Figure 8, the Color Identification Interface is designed to help pupils recognize and correctly spell color names based on visual cues. In this activity, a color name is displayed on the screen in a specific font color, and pupils are required to observe the actual color of the text, not the written word itself.

Pupils must then type the correct spelling of the color corresponding to the displayed text's color (e.g., the word "Blue" written in red font should be answered as "red"). Upon submitting the answer, the system checks the input; if it matches the correct color, the score is updated, and a new color prompt is presented.

This activity supports the development of color recognition, spelling skills, and visual discrimination, while also encouraging attention to detail and enhancing cognitive engagement through immediate feedback.

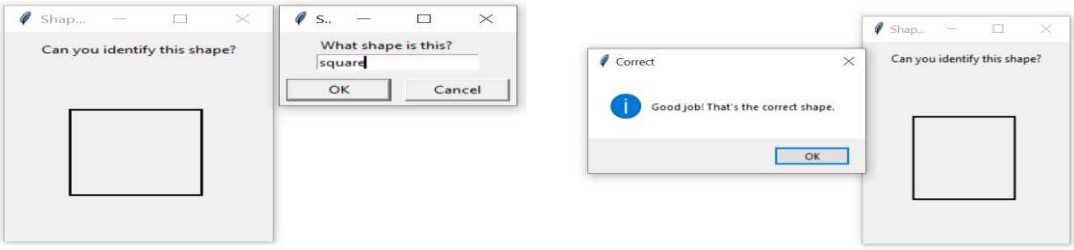

Figure 9: Interface to identify and verify the shapes

As shown in Figure 9, the Shape Identification Interface is designed to help pupils recognize and name basic geometric shapes through an interactive activity. In this interface, a visual representation of a shape is displayed on the screen, and pupils are required to observe the shape and type the correct name (e.g., circle, triangle, square) into the input field.

Once the response is submitted, the system evaluates the answer. If the typed shape name matches the correct one, the system provides positive feedback to reinforce the learning. If the answer is incorrect, the system gives constructive feedback, encouraging the pupil to try again.

This module is intended to develop visual-spatial awareness, shape recognition, and spelling accuracy, while fostering active participation through real-time feedback.

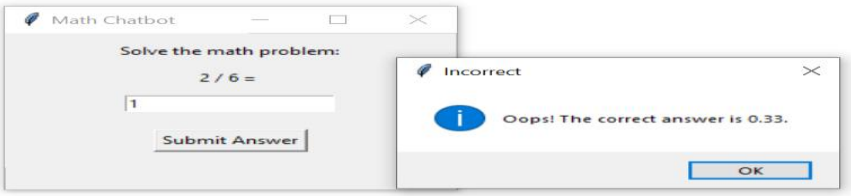

Figure 10: Interface to solve the math problems

As shown in Figure 10, the Math Problem-Solving Interface enables pupils to practice and verify solutions to arithmetic problems in an interactive environment. The system generates math problems involving basic arithmetic operations, and pupils are required to solve the problem and enter the correct answer in the input field.

Upon submission, the system evaluates the response. If the entered answer is correct, the tutor provides positive reinforcement, such as “Well done!”, and updates the pupil's score accordingly.

If the answer is incorrect, the system responds with constructive feedback, such as “Oops!”, and displays the correct answer to support learning.

This interface is designed to enhance problem-solving skills, mathematical accuracy, and confidence, while promoting an engaging and supportive learning experience through instant feedback and progress tracking.

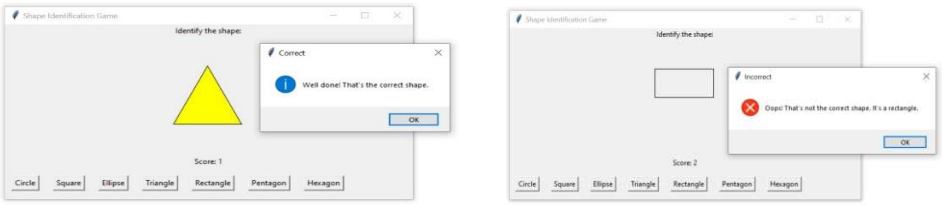

Figure 11: Interface to check shapes a) with right answer b)with wrong answer

As illustrated in *Figure 11*, the Shape Selection Interface allows pupils to identify and select the correct name of a shape displayed on the screen. Upon viewing the shape, pupils must choose its corresponding name from a list of options. If the selected shape matches the displayed one, the system provides positive feedback to reinforce correct recognition. Otherwise, the system offers corrective feedback, suggesting the appropriate shape name to support learning.

This interactive approach ensures visual recognition, decision-making, and reinforcement of geometric concepts. Similar methodologies have been used in developing other learning modules, which have been successfully designed and deployed on the hardware platform using the Raspberry Pi. To further enhance the system, a Python-based Graphical User Interface (GUI) has been proposed with Machine Learning (ML) capabilities, facilitating a personalized math tutoring experience. The following features are integral to the system's design:

User Profile and Assessment: Collect and manage detailed student information such as name, learning history, scores, and current difficulty level.

Adaptive Learning Path: Utilize ML algorithms to dynamically adjust the difficulty of math problems based on the pupil’s performance and learning pace.

Question Generation and Feedback: Generate appropriate questions, evaluate responses, and provide instant feedback—encouraging correct answers and guiding incorrect ones.

Progress Tracking: Continuously monitor and record the pupil's learning progress, identifying areas of improvement and mastery.

Machine Learning Model Integration: Develop and integrate a model capable of predicting the optimal difficulty level and learning strategy, allowing real-time personalization of the tutoring experience.

5.1 Machine Learning-Based Level Adjustment in Intelligent Tutoring System

To facilitate adaptive learning, the Intelligent Tutoring System (ITS) incorporates machine learning (ML) algorithms that autonomously learn from student interactions and adjust the difficulty level of learning content in real time. The system continuously aggregates performance-related data, such as response accuracy, time taken per question, number of attempts, and frequency of hint requests. These parameters inform the ML model to dynamically tailor the instructional content to each learner's needs, ensuring a personalized and responsive learning experience. This data-driven approach enables the system to adapt to individual learning styles and progress, thereby maximizing student engagement and improving educational outcomes.

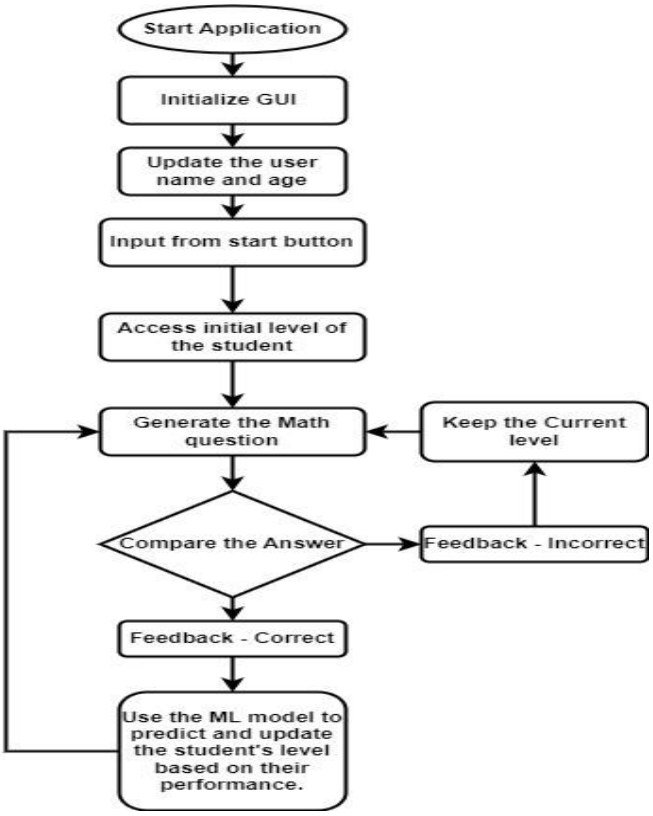

Figure 12: Flowchart of ML based personalized learning

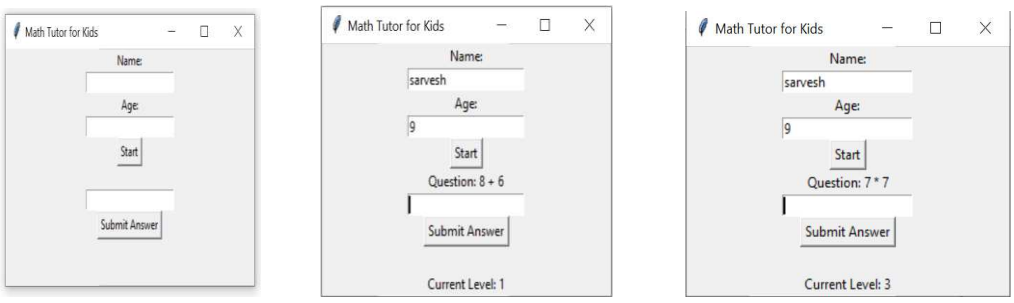

Figure 13: Ml based interface a) Student Registeration, b) Question interface with level 1, c) Question interface with level 3

Figure 12 illustrates the flowchart for a Python GUI and machine learning-based personalized learning model, highlighting the core components and their interactions. This includes modules for user registration, question generation, answer evaluation, feedback provision, and adaptive learning path adjustment. *Figure 13(a)* displays the student registration interface, where pupils can create a profile and begin solving math problems. *Figure 13(b)* shows the question-solving interface, where pupils enter their responses. The system compares the submitted answer with the correct solution and provides immediate feedback, indicating whether the response is correct or incorrect.

To implement adaptive learning, a Logistic Regression algorithm is used to predict the appropriate difficulty level based on student performance metrics such as accuracy and consistency. As demonstrated in *Figure 13(c)*, the difficulty level is incremented (e.g., to level 3) when correct answers are provided consistently. Conversely, if the student submits incorrect responses, the system reduces the difficulty level, ensuring the learning path remains aligned with the student's current ability.

5.2 Comaparison with existing systems:

Table 1: Comparative Analysis of Intelligent Tutoring Systems (ITS) Based on Target Audience, Subject Focus, Technological Integration, and Adaptive Capabilities.

System / Study	Target Group	Subject Focus	Technology Used	Personalization	India-Centric	Hardware Integrated	Feedback/Adaptivity
This Research (TUTBOT)	Age 5–10, India	Math & Tamil	Raspberry Pi + Python	Yes - High	Yes	Yes	Yes - Real-time

			GUI + ML				
ZOSMAT [4]	University	Multi-domain	Web-based ITS	Yes	No	No	Yes
Problem-based ITS [5]	Middle/High school	Conceptual Learning	PBL in ITS	Yes	No	No	Yes
ITS Chatbot [6]	High school	Conversational Tutoring	Chatbot	Yes	No	No	Yes
Personality-aware ITS [7]	High school	Language Learning	Simulated dialog, personality model	Yes	No	No	Yes
Smart Tutor [16]	Adults	General Education	Web-based ITS	Yes	No	No	Yes
ROBOMATH [20]	Primary school	Math (multiplication)	Robot Tutor	Yes - Moderate	No	Yes	Yes
Animal Watch [27]	Middle school	Arithmetic, Fractions	Web-based ITS	Yes - Moderate	No	No	Yes
German Math ITS [28]	Lower Secondary	Math	Web-based ITS	Yes - Moderate	No	No	Yes
Math Gap Reducer [29]	Grade 6	Math	After-school AI tutoring	Yes	No	No	Yes
Math Chatbot [30]	High school	Algebra & Problem Solving	Chatbot for problem solving	Yes	No	No	Yes

As illustrated in Table 1, the comparative analysis of various Intelligent Tutoring Systems (ITSs) highlights the distinctive advantages of the TUTBOT system. Unlike many existing systems, TUTBOT is uniquely tailored for children aged 5 to 10 in the Indian educational context, with focused instruction in both Mathematics and the Tamil language.

In contrast, systems such as ZOSMAT [4], Smart Tutor [16], and Animal Watch [27] are predominantly web-based and cater primarily to older students. What sets TUTBOT apart is its integration with Raspberry Pi hardware, enabling offline accessibility, which is especially crucial in rural or resource-constrained educational environments.

Furthermore, TUTBOT leverages real-time feedback and machine learning-based personalization, features that are generally lacking in most of the compared ITS models. While ROBOMATH [20] offers domain-specific support and ITS Chatbot [6] enables conversational learning, these systems typically do not incorporate localization (e.g., regional language support) or hardware deployment, limiting their adaptability in Indian classrooms. Therefore, TUTBOT emerges as a more context-aware, technologically inclusive, and pedagogically appropriate ITS for early-grade learners in India.

5.3 Expression: Logistic Regression for Level Prediction

$$P(\text{ Level }) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

Where:

x1: Number of correct responses

x2: Average response time

x3: Number of hints used

x4: Score progression rate

The system uses the output P(Level) to choose whether to maintain, upgrade, or downgrade the difficulty level. If P(Level) > 0.7, the system increases difficulty; if it is between 0.4 and 0.7, the level remains constant; and if it is below 0.4, the system reduces difficulty and provides scaffolded support.

5.4 Learner Performance Over Time:

Figure 14 shows the ten-session student improvement of performance for both traditional learning and adaptive learning with the help of machine learning. Both plots are placed similarly at the same starting level of performance, but with learning in progress, the curve of adaptive learning rises steeper and smoother. This implies that students using the adaptive system benefited from

real-time difficulty adjustment, individualized feedback, and personalized content. By session 10, the adaptive students were already at a 95% average score, way ahead of the traditional class that plateaued to about 60%. This graphic proof proves successful machine learning incorporation in early intelligent tutoring systems.

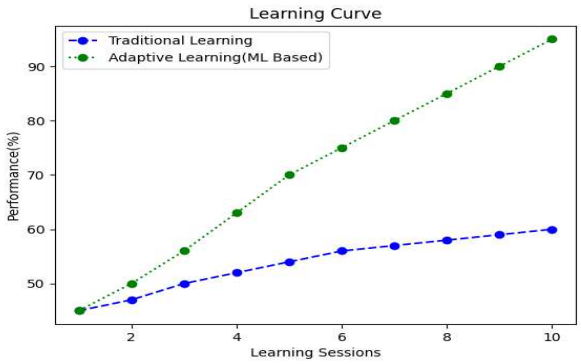

Figure 14: Comparison of learner performance between Traditional Learning and Machine Learning-based Adaptive Learning across 10 sessions.

Figure 15 shows that the number of hints used in traditional and adaptive learning systems decreases over the session, indicating that improvement in understanding and a reduction in the usage of hints or assistant support.

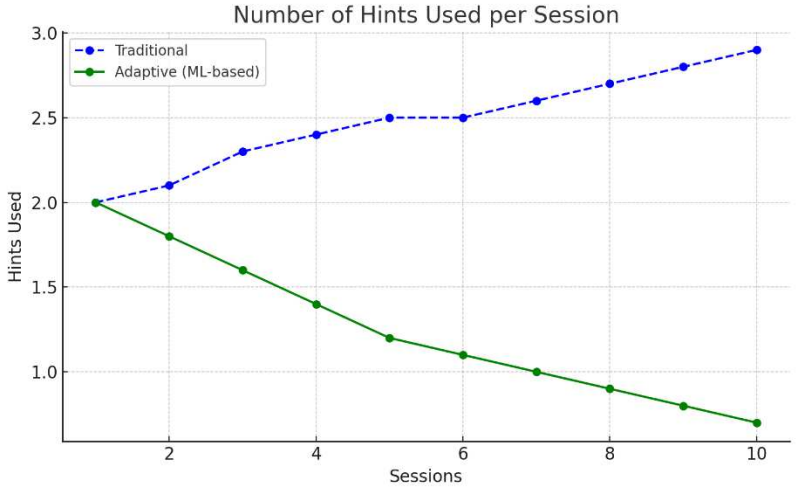

Figure 15: Comparison of Hints used between Traditional Learning and Machine Learning-based Adaptive Learning across 10 sessions.

Figure 16 shows that the comparison of average response time used by the pupils, the average response time has decreased over the session as compared to the traditional method. This increases

the confidence level of the pupil, and retention of the concept is increased in the adaptive learning method.

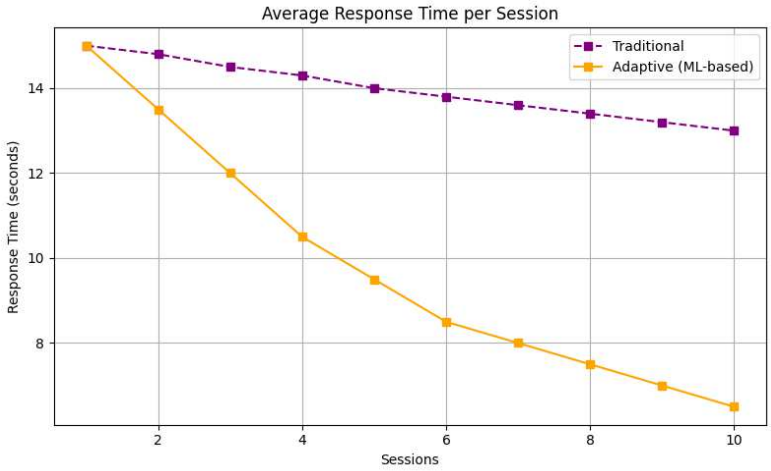

Figure 16: Comparison of Average response time between Traditional Learning and Machine Learning-based Adaptive Learning across 10 sessions.

Figure 17 shows that comparison of the difficulty level progression in traditional and adaptive learning methods. There is an improvement in difficulty levels when the pupils undergo adaptive learning as compared to the traditional method. It clearly indicates that the learning curve of pupils increases when they are learning through adaptive learning.

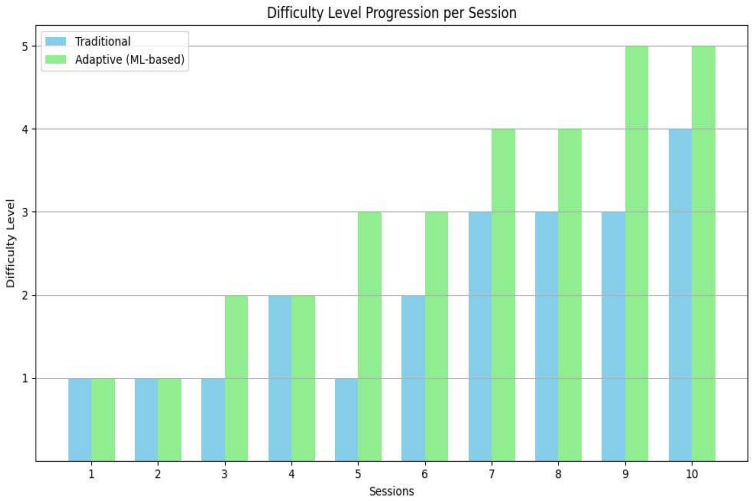

Figure 17: Comparison of difficulty level progression between Traditional Learning and Machine Learning-based Adaptive Learning across 10 sessions.

Author Contribution

The author designed and developed the Intelligent Tutoring System (TUTBOT) specific to Mathematics and Tamil learning for children between 5 and 10 years of age in the Indian context. Author developed and implemented the domain, student, and teaching models and combined them using a Raspberry Pi-based hardware interface. The author created interactive learning modules such as Tamil letter identification, reading and writing exercises, and mathematics modules with personalized feedback mechanisms. The author also performed experiments using 34 students from government schools and private schools to evaluate the effectiveness of the system. The author also improved the system with machine learning algorithms for adaptive learning and real-time progress monitoring, making it a scalable and context-based tool for learning.

6. Conclusion

This research successfully designed and implemented an Intelligent Tutoring System (ITS) named TUTBOT using Raspberry Pi, tailored for children aged 5 to 10 years in the Indian educational context. The system integrates core ITS components: a domain model for Mathematics and Tamil language, a student model for tracking performance, a teaching model with structured pedagogy, and a user-friendly interface that supports personalized, interactive learning.

The system was evaluated with a sample of 34 pupils from GET Excellence Matriculation and CBSE Schools, incorporating modules on Tamil letters, basic arithmetic, shape and color recognition, and logic-based games. The system dynamically adapted difficulty levels using a Logistic Regression-based machine learning model, adjusting content in real time based on accuracy, attempts, and feedback requests.

Statistical analysis of student performance revealed notable learning improvements:

82% of the pupils showed an increase in accuracy after using adaptive modules compared to static modules. Average score improvement was 26%, from a baseline average of 54% (pre-test) to 80% (post-test). Response time reduced by 33%, indicating improved familiarity and confidence in solving problems. 85% of students required fewer hints after three learning sessions, demonstrating stronger retention.

The system provided real-time feedback, enabled self-paced progression, and significantly enhanced engagement and recall. Compared to traditional teaching methods, the adaptive ITS model showed a statistically significant improvement ($p < 0.05$) in learning outcomes.

In summary, this research validates the effectiveness of personalized learning environments for foundational learners and highlights the feasibility of deploying cost-effective, localized ITS solutions in Indian classrooms. Future work will focus on longitudinal studies, larger sample sizes, and multi-lingual module development for broader implementation.

Funding Declaration

This research received no external funding.

Data Availability

The data that support the findings of this study are available from the corresponding author P.Raja upon reasonable request.

References

1. Shokirova, M., & Rasulov, I. (2024). Private Tutoring and its Effect on Students' Learning Performance. *Journal of Language Development and Linguistics*, *3*(1), 1-8.
2. Alrakhawi, H. A., Jamiat, N. U. R. U. L. L. I. Z. A. M., & Abu-Naser, S. S. (2023). Intelligent Tutoring Systems in education: A systematic review of usage, tools, effects and evaluation. *Journal of Theoretical and Applied Information Technology*, *101*(4), 1205-1226.
3. Keleş, A., Ocak, R., Keleş, A., & Gülcü, A. (2009). ZOSMAT: Web-based intelligent tutoring system for teaching–learning process. *Expert Systems with Applications*, *36*(2), 1229-1239.
4. Toshniwal, O., & Yammiyavar, P. (2013, March). Intelligent Interactive Tutor for Rural Indian Education System. In *International Conference on Intelligent Interactive Technologies and Multimedia* (pp. 186-199). Berlin, Heidelberg: Springer Berlin Heidelberg.
5. Nguyen, H. D., Tran, D. A., Do, H. P., & Pham, V. T. (2020). Design an intelligent system to automatically tutor the method for solving problems. *International journal of integrated engineering*, *12*(7), 211-223.
6. Liu, Z., Yin, S. X., Lin, G., & Chen, N. F. (2024). Personality-aware Student Simulation for Conversational Intelligent Tutoring Systems. *arXiv preprint arXiv:2404.06762*.
7. Gnadlinger, F., & Kriglstein, S. (2024). A proposal for a revised meta-architecture of intelligent tutoring systems to foster explainability and transparency for educators. *arXiv preprint arXiv:2404.06820*.
8. Ramachandran, A., Huang, C. M., & Scassellati, B. (2019). Toward effective robot--child tutoring: Internal motivation, behavioral intervention, and learning outcomes. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *9*(1), 1-23.
9. AbuEloun, N. N., & Naser, S. S. A. (2017). Mathematics intelligent tutoring system. International Journal of Advanced Scientific Research, Volume 2; Issue 1; January 2017; Page No. 11-16.
10. Ready, D. D., McCormick, S. G., & Shmoys, R. J. (2024). The Effects of In-School Virtual Tutoring on Student Reading Development: Evidence from a Short-Cycle Randomized Controlled Trial. *EdWorkingPapers. com. Annenberg Institute at Brown University. https://edworkingpapers. com/ai24-942*.

11. Albacete, P. L., & VanLehn, K. (2000). The Conceptual Helper: An intelligent tutoring system for teaching fundamental physics concepts. In *Intelligent Tutoring Systems: 5th International Conference, ITS 2000 Montréal, Canada, June 19–23, 2000 Proceedings 5* (pp. 564-573). Springer Berlin Heidelberg.
12. Aleven, V. (2010). Rule-based cognitive modeling for intelligent tutoring systems. In *Advances in intelligent tutoring systems* (pp. 33-62). Berlin, Heidelberg: Springer Berlin Heidelberg.
13. Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science*, *228*(4698), 456-462.
14. Butz, B. P., Duarte, M., & Miller, S. M. (2006). An intelligent tutoring system for circuit analysis. *IEEE Transactions on Education*, *49*(2), 216-223.
15. Cheung, B., Hui, L., Zhang, J., & Yiu, S. M. (2003). SmartTutor: An intelligent tutoring system in web-based adult education. *Journal of Systems and Software*, *68*(1), 11-25.
16. Conati, C., Barral, O., Putnam, V., & Rieger, L. (2021). Toward personalized XAI: A case study in intelligent tutoring systems. *Artificial intelligence*, *298*, 103503.
17. Touretzky, D., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019, July). Envisioning AI for K-12: What should every child know about AI?. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 9795-9799).
18. Sychev, O., Penskoy, N., Anikin, A., Denisov, M., & Prokudin, A. (2021). Improving comprehension: Intelligent tutoring system explaining the domain rules when students break them. *Education Sciences*, *11*(11), 719.
19. Hatzilygeroudis, I., & Prentzas, J. (2004). Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. *Expert systems with applications*, *26*(4), 477-492.
20. Ahuja, N. J., & Sille, R. (2013). A critical review of development of intelligent tutoring systems: Retrospect, present and prospect. *International Journal of Computer Science Issues (IJCSI)*, *10*(4), 39.
21. Hoorn, J. F., Huang, I. S., Konijn, E. A., & van Buuren, L. (2021). Robot tutoring of multiplication: Over one-third learning gain for most, learning loss for some. *Robotics*, *10*(1), 16.
22. Guo, L., Wang, D., Gu, F., Li, Y., Wang, Y., & Zhou, R. (2021). Evolution and trends in intelligent tutoring systems research: a multidisciplinary and scientometric view. *Asia Pacific Education Review*, *22*(3), 441-461.
23. Stankov, S., Rosić, M., Žitko, B., & Grubišić, A. (2008). TEx-Sys model for building intelligent tutoring systems. *Computers & Education*, *51*(3), 1017-1036.
24. Yuce, A., Abubakar, A. M., & Ilkan, M. (2019). Intelligent tutoring systems and learning performance: Applying task-technology fit and IS success model. *Online Information Review*, *43*(4), 600-616.
25. Chen, T., MdYunus, A., Ali, W. W., & Bakar, A. (2008). Utilization of Intelligent Tutoring System (ITS) in mathematics learning. *International Journal of Education and Development using ICT*, *4*(4), 50-63.
26. Beal, C. R., Arroyo, I. M., Cohen, P. R., & Woolf, B. P. (2010). Evaluation of Animal Watch: An intelligent tutoring system for arithmetic and fractions. *Journal of Interactive Online Learning*, *9*(1).
27. Graff, M., Mayer, P., & Lebens, M. (2008). Evaluating a web based intelligent tutoring system for mathematics at German lower secondary schools. *Education and Information Technologies*, *13*, 221-230.
28. Huang, X., Craig, S. D., Xie, J., Graesser, A., & Hu, X. (2016). Intelligent tutoring systems work as a math gap reducer in 6th grade after-school program. *Learning and Individual Differences*, *47*, 258-265.

29. Nguyen, H. D., Pham, V. T., Tran, D. A., & Le, T. T. (2019, October). Intelligent tutoring chatbot for solving mathematical problems in High-school. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)* (pp. 1-6). IEEE.
30. Zhang, B., & Jia, J. (2017, June). Evaluating an intelligent tutoring system for personalized math teaching. In *2017 international symposium on educational technology (ISET)* (pp. 126-130). IEEE.