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.

Knowledge-based Expert System for Identifying and Classifying Electronic Sensors Using Vp-expert

Fidelis Jumare Asengi
Nigerian Defence Academy
Valarie Oru Agbor
valarieoru.agbor@neu.edu.tr
Near East University
Loplem Joseph Maisamary
Mewar International University
Abdulrazak Shehu
Mewar International University

Research Article

Keywords: VP-Expert, Expert System, Sensor Identification, Arduino, Rule-Based System, Electronic Sensors

Posted Date: July 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10162527/v1

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

Additional Declarations: No competing interests reported.

KNOWLEDGE-BASED EXPERT SYSTEM FOR IDENTIFYING AND CLASSIFYING ELECTRONIC SENSORS USING VP-EXPERT.

Fidelis Jumare Asengi [1], Valarie Oru Agbor [2,3*], Loplem Joseph Maisamary [4], Abdulrazak Shehu[4]

[1] Department of Physics, Nigerian Defence Academy, Kaduna, Nigeria.
[2] Department of Biomedical Engineering, Near East University, Nicosia, TRNC
[3] Research Centre for Science, Technology and Engineering (BILTEM), Near East University, Nicosia 99138, Cyprus
[4] Department of Physics, Mewar International University, Nasarawa, Nigeria.

Corresponding author: Valarie Oru Agbor (valarieoru.agbor@neu.edu.tr) Telephone: +905338527416

Contributing authors: Fidelis Jumare Asengi **(fjumare4@gmail.com) institution email (fajumare@nda.edu.ng),** Loplem Joseph Maisamary (loplem101@gmail.com)institutional email ok its fidelisjumare@miu.edu.ng, Abdulrazak Shehu (abdulzigau@gmail.com) **institutional email (shehu.abdurrazaq@miu.edu.ng)**

The authors contributed equally to this work

Abstract

Accurate identification of electronic sensors is a common problem for undergraduate engineering students, hobbyists and makers, especially when they are unlabelled, have no datasheets or have been removed from their original packaging. This is a common issue in teaching labs, recycled electronics kits and community projects on microcontroller platforms such as Arduino, often limiting the hands-on learning and prototyping experience.

In order to solve this problem, this study developed a knowledge-based expert system using the VP-Expert shell. The system classifies different electronic sensors based on their physical properties and the output response on the serial monitor of Arduino IDE for some inputs. Expert knowledge has been modelled by IF-THEN production rules, so that the system can simulate the diagnostic reasoning of experienced practitioners of electronics.

This intelligent tool is a practical learning aid and diagnostic resource, showing how artificial intelligence can be effectively used to help electronics education and sensor experimentation.

Keywords: VP-Expert, Expert System, Sensor Identification, Arduino, Rule-Based System, Electronic Sensors.

1. Introduction

The electronic sensor is a basic part of the embedded and cyber-physical systems. These are used to transform a physical phenomenon like temperature, humidity, light intensity, or sound into an electrical signal which can be read and acted upon by computational devices like microcontrollers or embedded processors [1]. They are used in various applications such as home automation, environmental monitoring, health informatics, and industrial control systems [2].

Hands-on experience with sensors is central to the education process in educational settings and especially in undergraduate engineering and computer science courses. The ability to use microcontrollers such as the Arduino and Raspberry Pi has made sensor-based experimentation more accessible and allowed students and hobbyists to easily prototype and deploy in the real world [3]. An often-present problem, however, is the accurate identification of electronic sensors, particularly when they are not in packaging, not labelled, or not provided with datasheets. It is a common problem in a "recycled" teaching laboratory or in electronics kits for the classroom or community projects that involve making.

As the number of sensors in the market grows, so does the complexity of this problem. Some sensors may have similar pinout and/or physical layout but may differ significantly in their electrical characteristics and output format [4]: Some of them transmit analogue (continuous) voltages, others transmit binary (discrete) signals, and more sophisticated sensors transmit digital data protocols. The wrong sensor can cause the circuit connections to be wrong, because a software malfunction could damage the components, and can be detrimental to education and experimentation [5].

To solve this practical problem, this project aims to develop an expert system that is knowledge-based with the VP-Expert shell. The system is designed to help a user, especially a student or an electronics beginner, to identify a type of sensor by analysing the response of the sensor in a known condition of the environment or input. The expert system uses If-Then rule logic to determine the identity of the sensor based on the observed parameters of the sensor, which include the type of signal it generates (analogue or digital), the way it is reacting (continuous or threshold-triggered or encoded), its physical features and the modality of its response (a voltage level, resistance, signal pulse, etc.). For example, the sensor that produces a variable voltage if it receives light intensity can be called an LDR [6].

VP-Expert is a rule-based expert system shell in which the domain knowledge can be encoded in a declarative form of logic. Expert systems are based on structured knowledge acquisition and decision rules based on human knowledge or experimental observation, unlike general-purpose AI [7]. For this project, a rule base was created for the classification of 10 common electronic sensors that are used with the Arduino microcontroller, based on specifications provided by the manufacturers, testing, and literature from the education sector. The system queries the user about the unknown sensor's characteristics and then, based on the rule base, it provides an approximation of the sensor's analog/digital/hybrid classification, likely identity, and likely uses.

The system has significant real-world implications. It eliminates the need for trial and error learning, avoids sensor misclassification and improves learning through an interactive and

intelligent support tool. The method is especially well suited to low-resource situations where internet-based identification methods may not be available. Moreover, the knowledge base can be expanded to accommodate additional sensors over time, making it a scalable solution.

To conclude this project, the need for sensor identification in real-world applications (academic and hobbyist) has been highlighted. It helps create an intelligent support system by combining theoretical studies with practical experimentation with hardware.

1.1 Aim

The primary aim of this project is to develop a rule-based expert system using VP-Expert for the classification and identification of electronic sensors based on their physical features and output characteristics**,** to enable users regardless of their technical background, to correctly identify the category of a sensor and understand its signal behaviour and uses without requiring deep knowledge of electronics or sensor datasheets.

1.2 Objectives

To achieve the central aim of designing a knowledge-based expert system capable of identifying and classifying electronic sensors based on their physical features and output characteristics, the following specific objectives have been identified:

1. To Conduct a Thorough Review of Sensor Output Characteristics
2. To Acquire Expert Knowledge and Formalise it into Structured Rules
3. To Design a Rule-Based Inference System Using VP-Expert:
4. To Implement a User-Interactive Decision Support Interface
5. To Ensure Consistency and Accuracy of Sensor Identification
6. To Provide an Educational Tool for Learning Sensor Behaviour
7. To Evaluate and Test the System for Reliability and Usability

2. Literature Review

2.1 Expert Systems

Expert systems are a prominent subset of artificial intelligence (AI) that emulates the decision-making capabilities of human experts in specific domains. These systems rely on a structured knowledge base and an inference engine to solve problems, provide recommendations, or make decisions with a degree of accuracy comparable to that of domain specialists [8]. Over the past few years, expert systems have been widely used in several technical areas, such as diagnosis, system monitoring and classification.

Characterising the sensor according to the output and physical properties is an important and difficult problem in electronics. In practice, sensors often have overlapping features or nonlinear behaviour, which would be inefficient and prone to error to identify them manually [9]. To address this challenge, the development of a knowledge-based system using VP(Visual-Prolog)-Expert, a rule-based expert system shell, provides a practical solution. VP-Expert allows for the encoding

of heuristic knowledge into a system that can interactively guide users through identification processes based on predefined rules and sensor attributes.

2.2 VP-Expert Shell

Among the various tools developed for building such systems, VP-Expert stands out as a widely used expert system shell, particularly valued for its simplicity and flexibility in rule-based reasoning. Developed for DOS-based environments, VP-Expert enables users to create expert systems without extensive programming experience, using IF-THEN rules and a forward-chaining inference mechanism to reach conclusions [10].

The two prime elements of an expert system kernel are the Knowledge Base (KB) and Inference Engine, which are also key components in most expert system dedicated compute [11]. A knowledge base is a collection of all the knowledge and instructions about a certain subject in a certain area. A set of algorithms for judgment, reasoning and decision making is known as Inference Engine. An expert system's typical conceptual architecture is shown in Figure 1

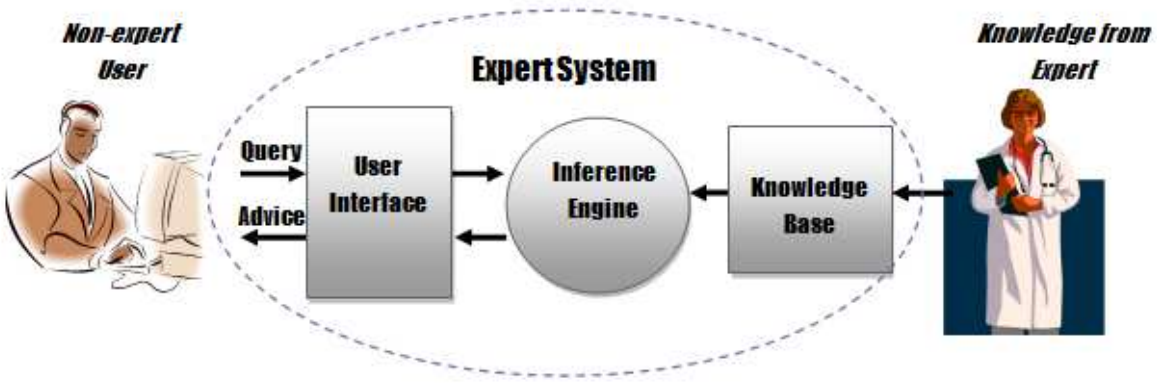


Figure 1: Expert System Kernel [12].

In technical applications, like distinguishing electronic sensors by their output properties and physical features, the development of expert systems is advantageous in efficiency and accuracy. Sensors often share overlapping features, such as similar voltage outputs or physical connectors, making manual identification challenging [13]. In this context, the use of VP-Expert can lead to the codification of the experts knowledge in the form of decision rules, and automated reasoning applied on the sensor data can lead to the determination of the sensor types effectively.

This research is concerned with the design and implementation of a Knowledge-Based system in the form of VP-Expert shell to recognize different kinds of electronic sensors. The system tries to help users, especially technicians and students, to classify the right sensors and understand their functional characteristics through analyzing the key sensor parameters, like output signal types, physical features.

2.3 Sensors

The sensors are the devices that are designed to sense different events and changes in the environment and the changes are converted into electrical signals which can be measured, interpreted and processed by microcontrollers and electronic system [14]. They are used to connect the real world to the digital world, allowing for automation, monitoring, and intelligent decision-making in various applications.

Sensors are essential in the modern technology. Sensors can be used in a variety of applications, such as smartphones and thermostats, all the way to more complex systems found in aerospace, healthcare, and robotics, to provide data for real-time processing and control [15].

2.3.1 Sensor Classification

In today's technology, sensors are a critical part of any electronic system, serving as the front line of communication between the physical world and digital systems. They transform physical phenomena such as temperature, pressure, motion, or light into electrical signals that can be processed and analysed. As there are a wide variety of sensors with different functions, design, and output characteristics, proper classification is significant for proper usage, integration and troubleshooting of electronic and automation systems [16].

The classification of sensors is usually based on certain parameters such as sensed variable (thermal, mechanical, optical, electronic), output signal (analog/digital), physical characteristics (size, housing, connector, etc.).

2.3.2 Electronic Sensor

Electronic sensors can be categorised in several ways depending on how they behave electrically and functionally. One of the most relevant classifications to this study is based on output characteristics [17].

2.3.3 Electronics Sensor Categories and Output Types

Sensors commonly found in Arduino kits can be classified based on their output characteristics:

2.3.3.1. Analogue Sensors

Analogue sensors produce a continuous range of voltage signals that vary proportionally with the physical quantity being measured. These sensors require the use of the Arduino's analogue input pins and typically utilise the ADC (Analog-to-Digital Converter) for signal interpretation [18].

1. LM35 (Temperature Sensor): Provides a linear analogue voltage output directly proportional to the temperature (10 mV/°C).
2. KY-037 (Sound Sensor): Outputs an analogue signal representing sound intensity, allowing sensitivity to be adjusted using a potentiometer.
3. Water Sensor: Produces a variable analogue voltage depending on the water level or moisture content detected via exposed parallel traces.
4. Potentiometer (10K): A variable resistor that outputs a voltage based on the wiper position, useful for manual input and calibration tasks.

5. LDR (Light Dependent Resistor): When combined with a voltage divider, the LDR provides an analogue voltage that changes with ambient light intensity [19].

2.3.3.2. Digital Sensors

Digital sensors have a binary output: HIGH (5V) and LOW (0V) depending on whether a threshold condition is achieved or not. These sensors are interfaced with the digital pins and are usually read via simple logic states [20].

1. IR Receiver (e.g., TSOP1738): Outputs digital signals when receiving modulated infrared pulses from a remote control.
2. Tilt Switch: Acts as a simple digital switch that toggles between HIGH and LOW based on the physical tilt or orientation.
3. Pushbutton: A momentary switch that provides a HIGH or LOW digital signal when pressed or released [20].

2.3.3.3 Hybrid Sensors

Hybrid sensors have both analogue and digital output. They can communicate digitally via communication protocols or have both types of output pins for flexible interfacing [21].

1. DHT11 (Temperature and Humidity Sensor): Although internally analogue, it transmits temperature and humidity data via a digital serial protocol**.** Thus, it's not a pure digital sensor in the binary sense but rather a hybrid that communicates structured data over a digital line.
2. Joystick Module: Consists of two potentiometers (for X and Y axes) providing analogue outputs and a pushbutton (Z-axis click) with a digital signal—thus fitting into the hybrid category [21].

Table 1: Decision Work Table of Class of Sensors

Sensors	Analog	Digital	Hybrid
DHT11	-	Yes	-
LM35	Yes	-	-
KY-037	-	-	Yes
Water Sensor	Yes	-	-
IR-Receiver	-	Yes	-
Tilt Switch	-	Yes	-
Potentiometer 10K	Yes	-	-
Joystick	-	-	yes
LDR	Yes	-	-
Pushbutton	-	Yes	-

2.3.4 Physical Features of Sensors

The table below shows the physical attributes of sensors from which they can be identified.

Table 2: Physical Attributes of Sensors

Sensor/Component	No. of Legs/Pins	Physical Description
DHT11	4 pins (sometimes 3 usable)	Small blue plastic case with vent holes, rectangular shape.
LM35	3 pins	Looks like a small transistor (TO-92 package), cylindrical/flat face.
KY-037 (Sound Sensor)	4 pins	Small rectangular PCB with a microphone capsule on top.
Water Sensor	3 pins	Flat, fork-shaped PCB with exposed copper traces to detect water.
IR Receiver	3 pins	Black epoxy dome or rectangular case, small and similar to a photodiode but bigger.
Tilt Switch	2 pins	Small cylindrical metal can (glass/metal body) with two leads.
Potentiometer (10K)	3 pins	Rotary knob with 3 terminals, often mounted on a round or rectangular housing.
Joystick Module	5 pins	Rectangular module with a thumb-stick (similar to a game controller joystick).
LDR (Light Dependent Resistor)	2 pins	Small round head with a zig-zag photoresistive track on top.
Pushbutton	4 pins	Small square button (tactile switch), plastic top with metal legs.

2.4 Related Work Done

2.4.1 Educational Expert Systems

Expert systems have long been applied in education and diagnostics. Projects such as the "Microprocessor Troubleshooting Expert System" [22] and "EDXPERT," a rule-based tutor for basic electronics, demonstrate how symbolic logic and decision trees can help students learn by simulating the guidance of a human instructor [23].

As regards the electronics-specific expert systems, [24] built an expert system using a shell that supports the selection of electrical instruments in laboratory setups. They had to enter the device range, resolution, and application type into their system, and it would generate the best instrument. While their work was not devoted to sensors, it did illustrate the capability of expert systems to aid in the streamlined selection of components by correlating user-specified attributes with expert logic. The high accuracy and consistency of the system were achieved by using the structured expert knowledge; fast response time was achieved by using the rule-based inference mechanism. Further, it enhanced the efficiency of the user and minimised errors in decision-making in the laboratory. Like other rule-based systems, however, it was only as effective as the completeness of its knowledge base and did not have adaptive learning capabilities, so it was not as robust in the face of new instruments or criteria.

One early implementation of this type of system is SENSORIX [25], designed to detect defective sensors in industrial environments using a rule-based approach. While not intended to be used in general sensor classification, SENSORIX illustrated how the status and behaviour of a sensor could be determined using an inference engine and by comparing the value and thresholds given in the knowledge base. It was found to be very reliable in fault detection, especially in controlled industrial environments, and responded quickly (due to its rule-based structure). But it suffered from the fact that it was rule-based and difficult to adapt to new or unknown types of sensors.

The identification of sensors is achievable in a data-driven and heuristic manner in recent studies. Shahid suggested a classification approach based on the temporal and statistical characteristics of the output signals of sensors. They used clustering algorithms based on the similarity of the outputs of the sensors, which allowed them to classify sensors like temperature, humidity and motion detectors in an ambient intelligent environment. The method demonstrated moderate-high classification accuracy (usually >80%) and was found to be scalable to multiple types of sensors, with the quality of the signal features. It was, however, very dependent on the quality of the feature extraction, and it was only effective for continuous data streams and not for sparse or discrete-output sensors [26].

In another interesting work, Jan used machine learning techniques such as Support Vector Machines (SVM) and k-Nearest Neighbours (kNN) to classify the types of sensors based on the output features generated in real-time. The results were very good in terms of classification accuracy (90% or more) and strong generalisation of results for various types of sensors. It did, however, need large labelled training sets, a lot of computing power, and a lot of signal data preprocessing. Such requirements limit the applicability of such systems for resource-limited systems, such as 8-bit or 32-bit microcontroller environments. [27]

3. Materials and Method

3.1 Hardware and Tools

i. Arduino Uno
ii. Electronic sensors: LM35, DHT11, LDR, Sound Sensor etc.
iii. Breadboard and jumper wires
iv. VP-Expert framework. Computer system for development and simulation

3.1.1. Arduino Uno Overview

The Arduino Uno is based on the ATmega328P microcontroller and features:

i. 14 digital I/O pins (6 capable of PWM)
ii. 6 analogue input pins (10-bit resolution ADC)
iii. Operating voltage: 5V
iv. USB interface for communication and power

These features make it an ideal platform for both analogue and digital sensor interfacing.

3.2. Data Collection

The data collection process utilised two primary sources:

1. Primary Data (Experimental Observation):

   Direct measurements and behavioural analysis were conducted by interfacing each sensor with the Arduino Uno. Real-time monitoring of voltage outputs, signal states (HIGH/LOW), and analog/digital behaviors was carried out using the Arduino IDE's Serial Monitor. It permits the verification of output types under varying physical conditions.

2. Secondary Data (Documentation and Manufacturer Datasheets): Data sheets, online resources (Arduino Sensor Libraries) and previous academic research were used to get technical specifications and expected behaviours of sensors. These references were used for baseline data like operating voltage, signal type, response time and internal circuitry.

During primary data collection, analogue sensors such as the LM35 Temperature Sensor, LDR with voltage divider, and Vcc was connected to 5V pin of the Arduino board, GND to the GND pin on the Arduino board and Output to one of the analogue input pins (A0 in this case).



Figure 2: Circuit Connection Sample (LM35 to Arduino) [28]

The digital Sensors (e.g., Tilt Sensor, IR Obstacle Sensor), the Vcc was connected to 5V, GND to GND and the digital output to the digital input pin (e.g., D2)

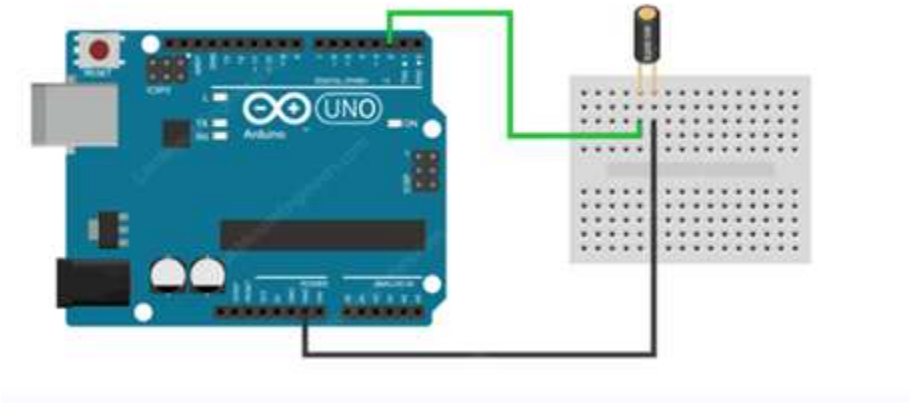


Figure 3: A Tiltsensor linked to Arduino [29]

Table 3: Showing the sample data collected from the serial monitor

Sensor	Signal/ Output Type	Output Behavior	Arduino Pin Used	Observation Method
LM35	Analog	Linear voltage proportional to temperature	A0	Measured temperature
DHT11	Digital	Encoded digital signal via protocol	D2	Measured humidity/temperature
KY-037	Analog/ Digital	Analogue amplitude + digital threshold	A1, D3	Detected claps/sounds
Water Sensor	Analog	Varies with water conductivity	A2	Dipped in water
LDR	Analog	Varies with light intensity	A3	Exposed to torchlight
Tilt Switch	Digital	ON/OFF depending on orientation	D4	Physically tilted
Potentiometer	Analog	Variable resistance (voltage)	A4	Adjusted dial
Joystick	Analog/ Digital	X-Y axis analogue + push button (digital)	A5, A6, D5	Pressed and tilted
IR Receiver	Digital	Encoded IR pulses	D6	Transmitted IR signal
Push Button	Digital	HIGH/LOW toggle on press	D7	Manually pressed

3.2.1 Data Validation and Preprocessing

The raw sensor readings were validated against known behaviours from datasheets and prior research. Erratic values or outliers were filtered out, and final validated behaviours were encoded into symbolic variables (e.g., analog_continuous, digital_pulse, hybrid_signal) used in the expert system rule base.

3.3 Developing the System (ESIS)

In developing ESIS, Prologue (Programming logic) was utilised as the programming language, which is frequently applied to tasks that involve symbolic and logical reasoning, including creating programs for expert systems. A software tool called VP (Visual Prologue) ES (expert system) shell was utilised for the project. Although there are other expert system shells out there like Clips, VPES was chosen because it offers a user-friendly interface for creating and developing expert systems without requiring the writing of complex code, and it is based on Visual Prologue.
The VPES shell was built by a company owned by Brian Sawyer called Visual Prolog International and was founded by Idea Ware Inc [12]. In this project, the ESIS design was integrated into the VPES shell's production environment.

VP-Expert is based on inference approach for backward reasoning. The user interface, the inference engine, and all other parts required for complete expert system architecture are all included in it.

3.3.1 Knowledge Representation

In expert systems, knowledge representation is the structured methodology of storing and applying expert-level knowledge to solve specific problems [28]. For this project, which involves classifying and identifying electronic sensors using VP-Expert, the chosen approach is rule-based representation. This approach encodes knowledge as a set of production (IF-THEN) rules derived from sensor attributes, behavioral analysis, and datasheet documentation obtained from the primary and secondary data collected.

1. Attribute-Based Sensor Representation. Each sensor used in this study was analyzed and encoded based on the following standardized attributes:

i. Physical Description: Describes the physical shape and size, helping distinguish sensors that appear similar. Example: KY-037 is a red board with
ii. Pin Count: The number of usable pins serves as an identifying feature. Example: DHT11 has 3 pins (VCC, Data, and GND), Joystick module has 5 pins.
iii. Stimulus Type: What the sensor reacts to e.g., temperature, sound, light, touch, motion, humidity.
iv. Output Type: Whether the output is analog, digital, or both. Example: LDR produces analog output and Push Button gives digital HIGH/LOW.
v. Output Behavior: How the sensor behaves under stimulus—continuous variation, threshold, binary toggling, etc. Example: KY-037 outputs HIGH when sound is detected above a threshold.
vi. Needs Library: Indicates whether the sensor requires an external Arduino library to interpret signals. Example: DHT11 requires a library; LM35 does not.

These features were exploited to develop rules that could correctly recognize a sensor without any labels even if it's behavior was only observed in controlled experiments.

2. Rule Design Using Sensor Attributes: The rules in VP-Expert were built by the logical combination of sensor attributes into rules. For each rule, the pattern is as follows:

```
IF <condition 1> AND <condition 2> AND ... THEN SENSOR = <sensor name> CNF
<confidence score>
```

Each condition corresponds to one attribute such as:

- `PHYSICAL_DESCRIPTION = round_with_zigzag_lines`
- `PIN_COUNT = 2`
- `OUTPUT_BEHAVIOR = voltage_decreases_as_light_increases`

Example 1: RULE to identify LM35

IF PHYSICAL DESCRIPTION = Tiny metal can AND PIN COUNT= 3 AND STIMULUS = Temperature AND OUTPUT TYPE = Analog Voltage AND OUTPUT BEHAVIOR = Voltage Increases AND NEEDS LIBRARY = No THEN SENSOR TYPE = LM35

Example 2: RULE to identify DHT11

IF PHYSICAL DESCRIPTION = Small blue box with vents AND PIN COUNT = 3 AND STIMULUS = Temperature & Humidity AND OUTPUT TYPE = Digital Pulses AND OUTPUT BEHAVIOR = Bitstream Encoding AND NEED LIBRARY = yes THEN SENSOR TYPE = DHT11

3.3.2 Implementation in VP-Expert

In VP-Expert, these rules are written in a structured text and the variables associated with these rules (e.g., STIMULUS, OUTPUT_TYPE, etc.) are declared in the Knowledge Base. The system uses forward chaining to evaluate known facts (e.g., observed output behavior, sensor response to light or heat) and apply rules that match these characteristics.

Below is a rule format in VP-Expert syntax:

RULE R1

PHYSICAL_DESCRIPTION = Round_with_zigzaglines AND PIN_COUNT = 2 AND STIMULUS_TYPE = light AND OUTPUT_TYPE = Analog_voltage AND OUTPUT_BEHAVIOR = Voltage_decreases AND NEEDS_LIBRARY = NO

THEN SENSOR = PHOTORESISTOR-LDR

DISPLAY "The Sensor is an ANALOG SENSOR.

It is used to detect light intensity by changing its resistance based on the amount of light it receives.";

These rules are evaluated sequentially or in parallel depending on the structure of the inference engine. Once a matching rule fires, the system classifies the sensor and displays its name or outputs a corresponding message.

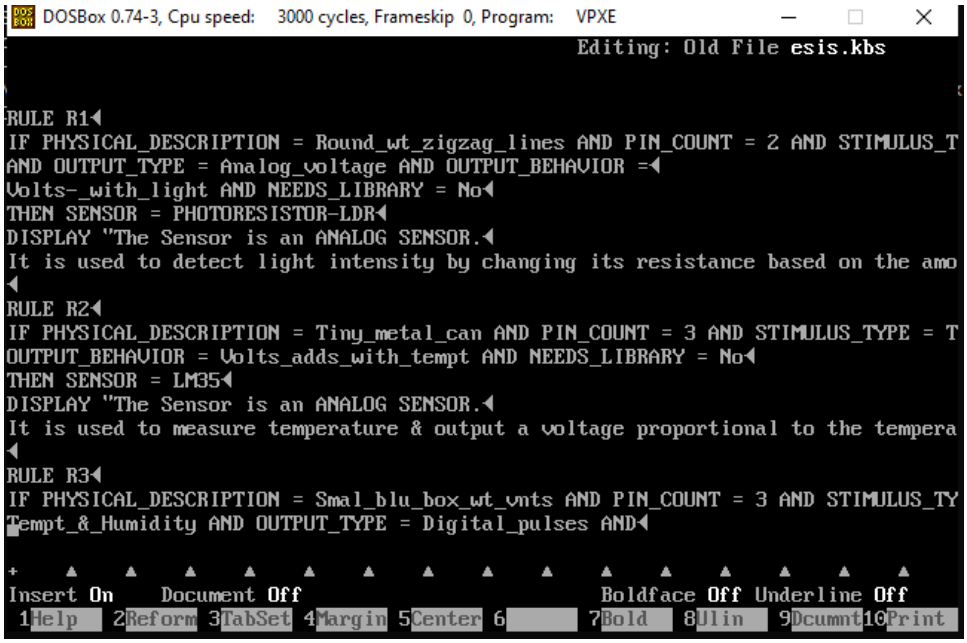


Figure 4: Example of Rules formed and stored in the KB

3.3.3 ACTIONS block

This is the code that controls how the inference engine is implemented. Declarations that regulate system activities are included in the ACTIONS block. These declarations are made inside the command where they appear. The DISPLAY explanation instructs the client on what to perform. The FIND statement expresses the framework's goals, which is to diagnose in this case. The outcomes are presented in the final declaration [29].

3.3.4 Query Statements:

When a query is conducted, Visual Prolog looks for solutions that match the query in the knowledge base. It will then display the results, which may include variable bindings if they were specified in the query. If a query matches multiple facts, it will return multiple answers. If no matches are discovered, it returns false. The ASK and CHOICES statements conclude this process [30].

3.3.4.1 ASK statement:

The ASK statement describes how to prompt for a variable. These prompts must be useful, just like any other application. It includes the following steps:

ASK variable: "prompt";

For instance:
ASK PIN_COUNT:" How many pins does the sensor module have?";

3.2.4.2 The CHOICES statement:

If there are just a limited number of possible answers to a query, they may be found in a menu created by a CHOICES declaration. It includes the following procedure:
CHOICES variable: list of values separated by a comma;
For instance:
CHOICES PIN_COUNT: 2, 3, 4, 5, more_than_5:

Once the inquiry has been submitted, this menu is published. Please be aware that if there was no CHOICES statement for a variable, the user must enter a value on the cursor later in the prompt.

The identifying system is comprised of six condition questions such as, the physical description of the sensor, Number of pins or legs on the sensor, the stimulus applied to the sensor e.g. Light or sound, the output type of the sensor when connected to Arduino, the output behavior and lastly the sensors' need for an external library.

3.4 System Architecture

The Electronic Sensor Identification System (ESIS) is structured into five key modules, each representing a functional stage in the rule-based classification workflow. These include:

1. Arduino + Sensor (Offline Data Collection): At this stage, various sensors are interfaced with an Arduino system to collect electrical output data under controlled conditions. This process is carried out offline and serves as the foundation for building the expert system.
2. Knowledge Base (Rules): The analyzed data from the Arduino (primary data) and the secondary data is used to formulate rules that represent the behavior of each sensor.
3. User Input: The user input parameters to the system, such as observed sensor pin number. These acts as facts that the system will evaluate.
4. Inference Engine: The Inference engine processes the user input by matching it against the rules in the knowledge base. Using logical reasoning (Forward Chaining), it determines which rule(s) are satisfied and arrives at a conclusion.
5. Output Display: The final result of the reasoning process is presented to the user through the output module. This typically includes the identified sensor type, class and uses.

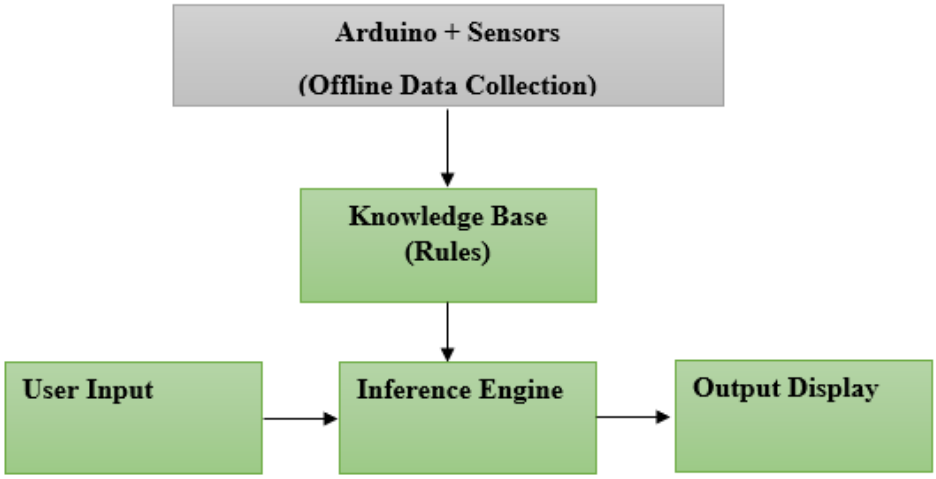


Figure 5: ESIS Architecture

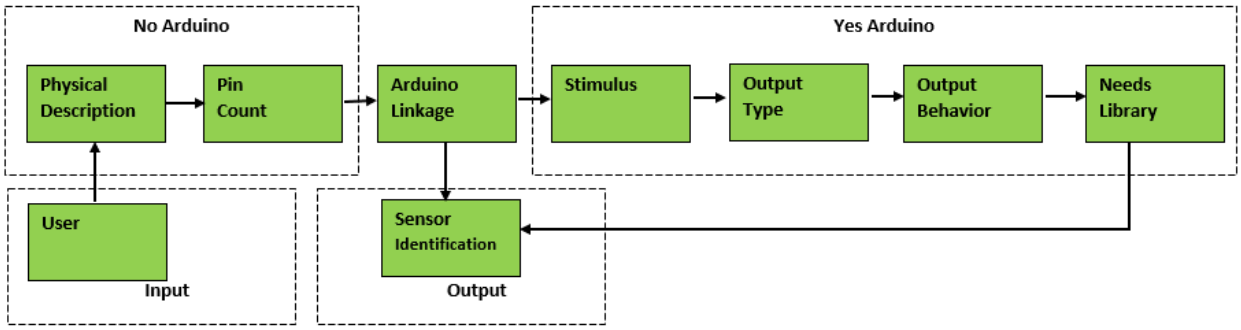


Figure 6: Decision Flow Chart for Sensor Identification

1. Input Block

   i. User: The process begins with a human user providing basic, observable sensor information.

2. No Arduino Block (User Enters Sensor Characteristics)

   i. Physical Description: The user describes the physical features of the sensor—its shape, size, color, casing, etc.

   ii. Pin Count: The number of pins on the sensor is noted, which helps narrow down possible types (e.g., 2-pin = LDR, 3-pin = LM35 or IR Receiver, etc.).

These two parameters are independent of the Arduino board, hence grouped under "No Arduino". They allow for preliminary classification.

3. Arduino Linkage Block (Hardware Interface)

   i. Arduino Linkage: The sensor is connected to an Arduino Uno board for further electronic testing. This step transitions the process into the "Yes Arduino" domain.

4. Output Characterization Block (Yes Arduino)

In this section, user observes various characteristics from the serial monitor by connecting sensors with Arduino and responding to the ESIS user interface:

i. Stimulus: Identifies what physical input the sensor responds to (e.g., temperature, light, tilt, sound).

ii. Output Type: Determines whether the sensor produces analog, digital, or serial (hybrid) signals.

iii. Output Behavior: Analyzes the pattern of the signal (e.g., linear voltage, threshold-based switching, serial communication).

iv. Needs Library: Checks if the sensor requires a specialized software library for interfacing (e.g., DHT11 needs the DHT library).

These attributes are critical to uniquely identifying the sensor**.**

5. Final Output Block (Sensor Identification)

Once all attributes (both physical and electronic) are entered into the system, the ESIS powered by the VP-Expert knowledge-based system, matches this data to a set of predefined rules to determine the exact sensor type.

3.5 Implementation in VP-Expert

The VP-Expert system asks a set of questions (e.g., "Which of the following best describes the physical appearance of the sensor?") and then applies rules and infers from the user's answers the type of sensor as depicted in Figure 7 below.

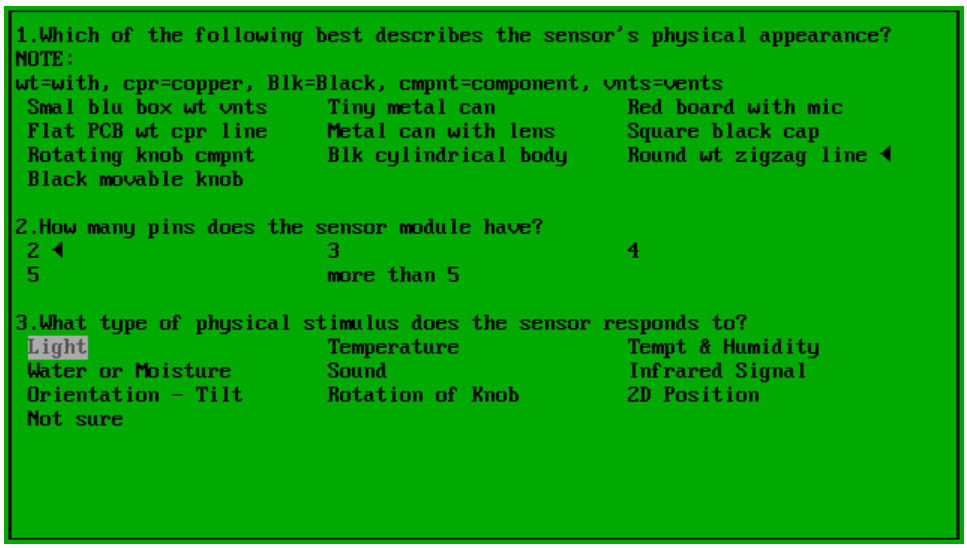


Figure 8: User responding to questions

Table 4: Sensor Identification Decision Table

Facts	Physical Description	Pin Count	Stimulus Type	Output Type	Output Behavior	Needs Library	Sensor Identification
1	Small blue box with vents	3	Temperature & Humidity	Digital pulses	Bitstream encoding	Yes	DHT11
2	Tiny metal can	3	Temperature	Analog voltage	Voltage increases	No	LM35
3	Red board with mic	4	Sound	Analog & Digital voltage	Voltage varies + ON & OFF	No	KY-037
4	Flat PCB with copper lines	3	Water/moisture	Analog	Voltage varies with water.	No	Water Sensor
5	Metal can with lens	3	Infrared signal	Digital pulses	Remote data encoding	Yes	IR Receiver
6	Square black Cap	2	Orientation/tilt	Digital signal	HIGH/LOW on tilt change	No	Tilt Switch
7	Rotating knob component	3	Manual rotation	Analog	Voltage varies with rotation	No	Potentiometer
8	Black movable knob	5	Manual movement (X/Y/)	2 Analogue voltages	Voltage varies with joystick movement	No	Joystick
9	Round with zigzag lines	2 (in divider)	Light intensity	Analog voltage	Resistance decreases w/ light	No	Photoresistor-LDR
10	Small switch, 4 pins	4	Manual press	Digital	HIGH/LOW on press	No	Pushbutton

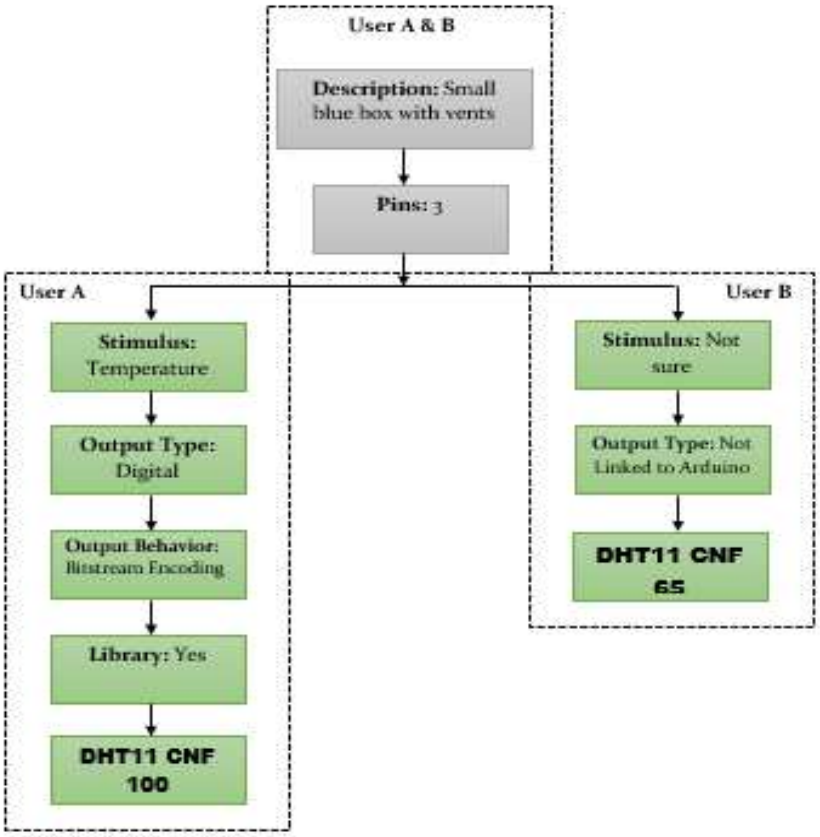


Figure 7: Flowchart for Identification and Classification of DHT11 Sensor Variants

This diagram shows how ESIS reaches a decision with different levels of user input.

Initially, both User A and User B are provided with the same basic information: Physical Description: small blue box with vents, Pins Count: 3. User A provided complete information: Stimulus: Temperature, Output Type: Digital, Output Behavior: Bitstream encoding, Library: Yes. Then the system recognizes sensor as DHT11 with 100% accuracy and the information is complete and specific. User B selected “Not sure” for Stimulus and “Not liked to Arduino” then the system recognizes the sensor as DHT11 with 65% accuracy based on the Physical features (Physical Description and Pin Count) of the sensor alone.

4. Results and Analysis

The system was proven with true sensor data from the Arduino. It has correctly recognized all the sensors according to the output of the sensors. For instance, in case the voltage is analog and it decreases in the presence of light, ESIS recognized the sensor as an LDR.

The system offers a simple interface and decision tree that is similar to the way an electronics expert would determine to manually identify sensors. The number of sensors is restricted to some common ones, but can be expanded to others.

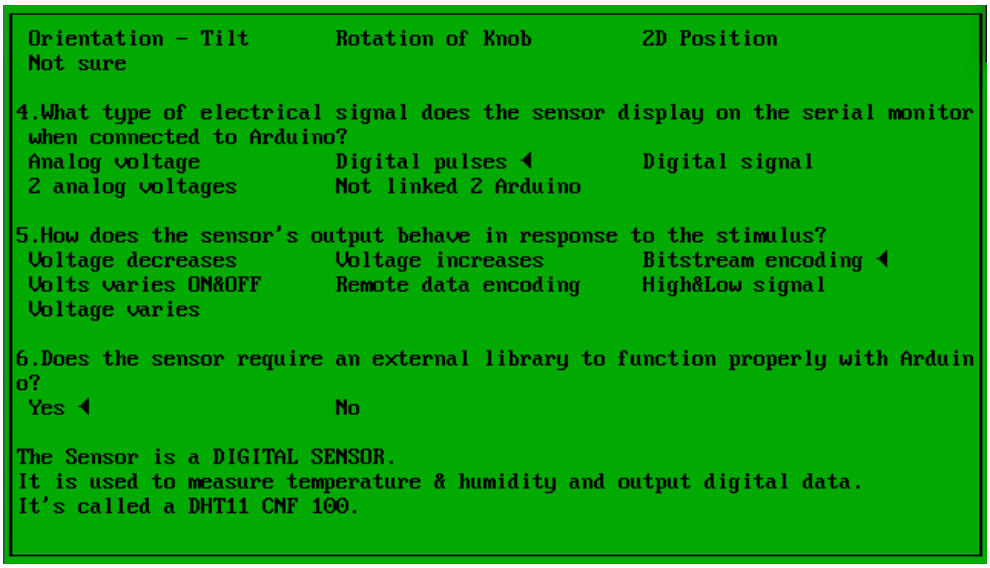


Figure 9: Result showing DHT11 Digital Sensor Identified (User A)

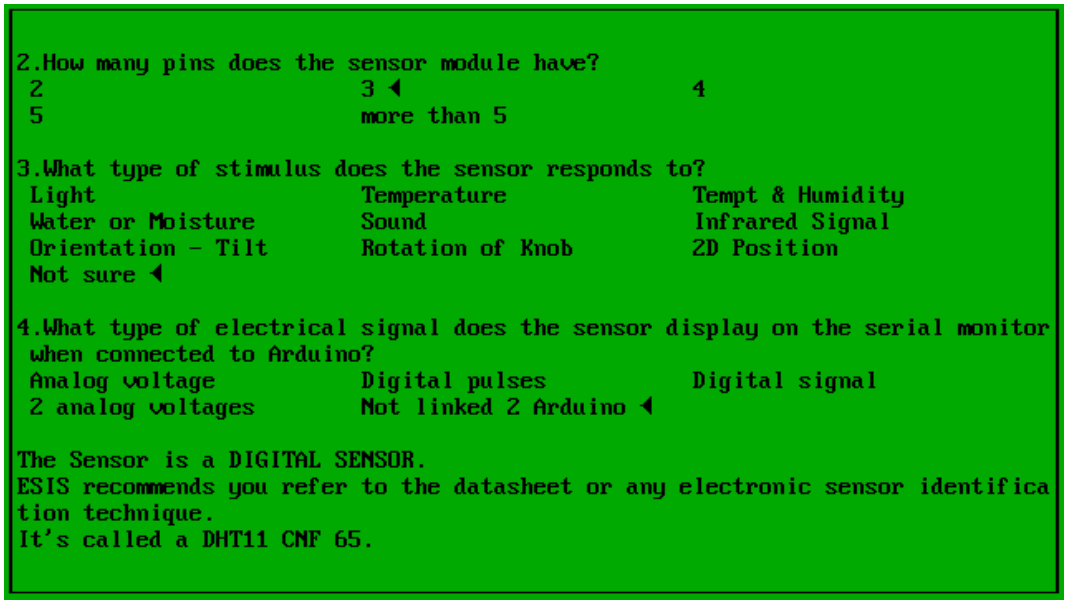


Figure 10: Result showing DHT11 partly identified (User B)

4.1 Evaluation Methodology and Metrics

The ESIS was evaluated using a structured method to determine the accuracy, reliability and utility of the system. The evaluation was conducted on three major aspects of the system performance which are the System Usability, Identification Accuracy and Response Time.

4.1.1 Test Design

The ten sensors tested in this study were individually tested in the complete identification process. The following steps were followed for a test process:

i. Connecting each sensor to the Arduino Uno under controlled conditions.
ii. Applying relevant stimuli (e.g., light for LDR, temperature for LM35, sound for KY-037).
iii. Recording sensor outputs on the Arduino IDE serial monitor.
iv. Manually inputting observed characteristics into the VP-Expert shell.
v. Logging the system's identification output.

Each sensor underwent five repeated tests to verify consistency and rule robustness. This resulted in a total of 30 test cases.

Table 5: Sample Test cases

Test Case	Sensor Used	Input Characteristics (as entered)	Expected Output	Actual Output	Result
TC1	DHT11	3 pins, digital, needs library, temp & humidity stimuli	DHT11	DHT11	Pass
TC2	LM35	3 pins, analog, no library, temp stimulus	LM35	LM35	Pass
TC3	LDR	2 pins, analog, no library, light stimulus	LDR	LDR	Pass
TC4	Tilt Switch	2 pins, digital, tilt stimulus	Tilt Switch	Tilt Switch	Pass
TC5	IR Receiver	3 pins, digital pulse, requires library, IR stimulus	IR Receiver	IR Receiver	Pass
TC6	Push Button	4 legs, digital, manual press stimulus	Push Button	Push Button	Pass
TC7	Potentiometer	3 pins, analog, rotation stimulus	Potentiometer	Potentiometer	Pass
TC8	Water Sensor	3 pins, analog & digital, conductivity-based moisture stimulus	Water Sensor	Water Sensor	Pass
TC9	KY-037	4 pins, dual mode (analog + digital), sound stimulus	KY-037	KY-037	Pass

TC10	Joystick Module	5 pins, dual (analog + digital), 2D motion + press	Joystick Module	Joystick Module	Pass
TC11	Tilt Switch	2 pins, black cylindrical body, not linked to Arduino	Tilt Switch	Tilt Switch	Pass
TC12	KY-037	4 pins, red board with mic, not linked to Arduino	KY-037	KY-037	Pass
TC13	Potentiometer	3 pins, rotating knob component, not linked to Arduino	Potentiometer	Potentiometer	Pass
TC14	LM35	3 pins, tiny metal can	LM35	IR Receiver	Fail

4.1.2 Evaluation Metrics

Total Classification Attempts: 30
Correctly Identified Sensors: 27
Misclassified: 3
Classification Accuracy: $\frac{27}{30} \times 100 = 90\%$

4.1.3 Observations

The system consistently matched each sensor's profile to a unique rule without ambiguity ad the output confidence scores (e.g., CNF 100) helped indicate rule certainty.Users found the ASK/CHOICES interface intuitive, though suggested GUI integration as a future enhancement.

4.1.4 Comparison with Manual Identification:

Manual identification required datasheet analysis and functional testing using Arduino IDE but ESIS reduced dependency on external datasheets and avoided code compilation with time saved per identification (5 minutes approximately)

5.0 Conclusion

This study has demonstrated the feasibility and effectiveness of using a rule-based expert system developed on the VP-Expert shell for the classification and identification of commonly used electronic sensors based on their output characteristics and physical attributes. The system converts the empirical observations and the knowledge of experts into structured IF-THEN rules that emulate the way humans think when faced with a question about electronics.

Under controlled conditions, the system was evaluated using ten different types of sensors with a 90% classification rate. It offers an offline, interpretable and low cost approach for sensor

identification in educational lab, maker spaces; resource limited environments where datasheets or online connection might be unavailable.

The system is not just a working laboratory tool, but a pedagogical tool to enhance the students' understanding of the behavior of sensors and logical reasoning in electronics. It brings the best of both worlds: hardware experimentation and AI by using expert systems to real-world embedded scenarios.

The ESIS system has its drawbacks however. It is currently implemented for a fixed set of sensors, it have to be entered manually, and there is no real-time integration. These restrictions have the potential to be fertile ground for further development.

5.1 Recommendations for Future Work

The following improvements are suggested to improve the capability and applicability of the ESIS framework:

1. Expansion of the Knowledge Base to include more sensor types and modules (including I2C/SPI devices).
2. Integration with a GUI to make the system more user-friendly and visually intuitive.
3. Incorporation of Real-Time Serial Data Capture from Arduino to reduce manual input.
4. Combining Rule-Based Logic with Machine Learning, to enable adaptive classification of previously unknown sensors.
5. Deployment as a Standalone Educational Tool or plugin for Arduino IDE for widespread classroom use.

This foundation can be extended to develop future iterations of the system that can be expanded to a more dynamic and intelligent diagnostic and learning assistant in the field of electronics and embedded systems.

.

Author Contributions All the authors have contributed equally.

Funding The authors did not receive support from any organization for the submitted work.

Code Availability The implementation of the system can be found here: Code Link.

Declarations
Conflict of Interests The authors have no conflicts of interest to declare that are relevant to the content of this article

References

[1] Gazis, A., Papadongonas, I., Andriopoulos, A., Zioudas, C., & Vavouras, T. (2025). A comprehensive review of sensor technologies, instrumentation, and signal processing solutions for low-power Internet of Things systems with mini-computing devices. *arXiv preprint arXiv:2503.13466*.

[2] Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0. *Sensors*, *23*(16), 7194.

[3] Prabhu, G. R. D., & Urban, P. L. (2020). Elevating chemistry research with a modern electronics toolkit. *Chemical Reviews*, *120*(17), 9482-9553.

[4] LALALI, M. A. (2024). Study and Implementation of an Infrared Remote Control.

[5] Belgacem, H., & Chihi, I. (2025). Toward Reliable and Intelligent Sensor Systems: A Comprehensive Study of Fault Diagnosis and Mitigation. *IEEE Sensors Reviews*.

[6] Setya, W., Ramadhana, A., Ramadhana, A., Restu Putri, H., Santoso, A., Malik, A., & Chusni, M. M. (2019, December). Design and development of measurement of measuring light resistance using Light Dependent Resistance (LDR) sensors. In *Journal of Physics: Conference Series* (Vol. 1402, No. 4, p. 044102). IOP Publishing.

[7] Sadi, A. R. (2013). Design and Implementation of an Expert System for Monitoring and Management of Web-Based Industrial.

[8] Sabzaliyev, A. (2024). Knowledge representation in expert systems: structure, classification, and applications. *Luminis Applied Science and Engineering*, *1*(2), 1-15.

[9] Sharma, S., Gupta, K., Gupta, D., Rani, S., & Dhiman, G. (2024). An insight survey on sensor errors and fault detection techniques in smart spaces. *Computer Modeling in Engineering & Sciences*, *138*(3), 2029.

[10] Chau, K. W., & Albermani, F. (2005). A knowledge-based system for liquid retaining structure design with blackboard architecture. *Building and environment*, *40*(1), 73-81.

[12] Asengi, F. J., Agbo, V. O., Gana, J. J., Aboh, M. O., & Ibrahim, A. (2025). Artificial intelligence-based expert system for diphtheria diagnosis. *AJSE*, *20*(1).

[13] Abdulhussain, S. H., Mahmmod, B. M., Alwhelat, A., Shehada, D., Shihab, Z. I., Mohammed, H. J., ... & Hussain, A. (2025). A comprehensive review of sensor technologies in IOT: Technical aspects, challenges, and future directions. *Computers*, *14*(8), 342.

[14] Kumar, K., Sharma, A., & Tripathi, S. L. (2021). Sensors and their application. In *Electronic devices, circuits, and systems for biomedical applications* (pp. 177-195). Academic Press.

[15] Abduljawwad, M., Khaleel, M., Ogedengbe, T. S., & Abraheem, S. (2023). Sensors for daily utilization. *Int. J. Electr. Eng. and Sustain.*, 106-119.

[16] Chowdhury, A., & Nuruzzaman, M. (2023). Design, Testing, And Troubleshooting of Industrial Equipment: A Systematic Review Of Integration Techniques For US Manufacturing Plants. Review of Applied Science and Technology, 2(01), 53-84.

[17] Vetelino, J., & Reghu, A. (2017). *Introduction to sensors*. CRC press.

[18] Barrett, S. F. (2022). Analog to digital conversion (ADC). In *Arduino Microcontroller Processing for Everyone!* (pp. 163-215). Cham: Springer International Publishing.

[19] Susilawati, A., Kustiawan, I., Rochintaniawati, D., Hasanah, L., & Lim, Y. (2024). Light intensity distribution in the room using light dependent resistor: Through the engineering design process. *Indonesian Journal of Science and Technology*, *9*(3), 679-708.

[20] Kumar, A. A. (2016). *Fundamentals of digital circuits*. PHI Learning Pvt. Ltd..

[21] Ma, Y., Zhang, Y., Cai, S., Han, Z., Liu, X., Wang, F., ... & Feng, X. (2020). Flexible hybrid electronics for digital healthcare. *Advanced Materials*, *32*(15), 1902062.

[22] El Kahlout, F., & Abu-Naser, S. S. (2023). Developing an expert system to computer troubleshooting.

[23] Nkambou, R., Brisson, J., Tato, A., & Robert, S. (2023). Learning logical reasoning using an intelligent tutoring system: a hybrid approach to student modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 37, No. 13, pp. 15930-15937).

[24] Ghatol, A. A., & Wanjari, R. S. (2002). *Artificial intelligence and expert systems*. Tech-Max Publications.

[25] Rothmaier, M. (2003). SENSORIX AG: Development, Optimization, and Control of Biotechnological Processes Through Continuous Online Analytics with Chemical Sensors and Biosensors. *Chimia*, *57*(6), 352-352.

[26] Shahid, N., Naqvi, I. H., & Qaisar, S. B. (2015). Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey. *Artificial Intelligence Review*, *43*(2), 193-228.

[27] Jan, S. U., Lee, Y. D., Shin, J., & Koo, I. (2017). Sensor fault classification based on support vector machine and statistical time-domain features. *IEEE Access*, *5*, 8682-8690.

[28] Das, D. (2022, April 13). *How Does a LM35 Temperature Sensor Work and how to Interface it with Arduino?* CircuitDigest. https://circuitdigest.com/microcontroller-projects/interfacing-lm35-sensor-with-arduino

[29] Expert, A. (2022, May 30). How to use tilt switch sensor with arduino. *Arduino Expert*. https://arduinoexpert.com/blog/how-to-use-tilt-switch-sensor-with-arduino/

[30] Sabzaliyev, A. (2024). Knowledge representation in expert systems: structure, classification, and applications. *Luminis Applied Science and Engineering*, *1*(2), 1-15.

[31] Janjanam, D., Ganesh, B., & Manjunatha, L. (2021, February). Design of an expert system architecture: An overview. In *Journal of Physics: Conference Series* (Vol. 1767, No. 1, p. 012036). IOP Publishing.

[32] West, R., Gabrilovich, E., Murphy, K., Sun, S., Gupta, R., & Lin, D. (2014, April). Knowledge base completion via search-based question answering. In *Proceedings of the 23rd international conference on World wide web* (pp. 515-526).