Neutrosophic Sets and Systems, Vol. 99, 2026
University of New Mexico

# Neutrosophic Logic as a Framework for Managing Uncertainty in Artificial Intelligence

Conf. univ. dr. **N. Grigorie Lăcrița*, Craiova, Romania**

* Correspondence: n.grigorie@yahoo.com

**Abstract:** Artificial Intelligence systems are increasingly required to operate in environments characterized by incomplete, ambiguous, and contradictory information. Traditional probabilistic and fuzzy-logic approaches often compress uncertainty into a single scalar value, limiting their ability to distinguish between ignorance, conflict, and evidential support. This paper explores the conceptual and practical relationship between Neutrosophic Logic and modern Artificial Intelligence architectures. The neutrosophic framework represents knowledge through three independent dimensions—Truth (T), Indeterminacy (I), and Falsity (F)—thereby providing a richer representation of uncertainty than conventional binary or probabilistic models. The study examines how neutrosophic triplets can be implemented as software primitives and applied in Retrieval-Augmented Generation (RAG) systems, multi-agent architectures, cybersecurity workflows, and AI-assisted decision-making. Particular attention is given to the role of indeterminacy as a measurable and actionable variable that enables systems to recognize incomplete knowledge, avoid hallucinations, and improve reliability. The paper argues that the integration of neutrosophic principles into AI engineering offers a promising pathway toward more robust, transparent, and trustworthy intelligent systems capable of operating under real-world uncertainty.

**Keywords:** Neutrosophic Logic, Artificial Intelligence, Uncertainty Modeling, Indeterminacy, Retrieval-Augmented Generation (RAG), Multi-Agent Systems, Decision Support Systems, Data Fusion, Explainable AI, Intelligent Software Engineering.

---

## 1. Introduction

> "Between truth and falsehood stretches an ocean of indeterminacy. True intelligence does not consist only in choosing between black and white, but in navigating precisely through shades of gray." — Florentin Smarandache

In modern software development, we have become accustomed to decisions being clear: a binary indicator, True or False; a status code, 200 OK or 500 Error; or a confidence score from 0.0 to 1.0.

When it comes to Artificial Intelligence (AI), we often encode uncertainty as a single scalar value.

For example, a classification model tells us that an image contains a cat with a probability of 0.85.

Although convenient for APIs and dashboards, this compression into a single number hides critical operational details.

What does a score of 0.5 returned by an LLM or by a cybersecurity system actually mean?

1. Does it mean that the evidence is weak?
2. Does it mean that the data is entirely missing?
3. Or does it mean that we have two excellent sources that violently contradict each other?

In software engineering, each of these three states requires a different reaction in code.

The answer to this problem comes from a universal mathematical discipline conceptualized by Florentin Smarandache: Neutrosophic Logic.

**2. The Problem of Information Reduction in Current AI**

> "The world is governed not only by certainties or contradictions, but above all by neutrality, ignorance, and the unknown. Neutrosophy is the mathematics of these unresolved states." — Florentin Smarandache

Many contemporary AI systems (such as RAG—Retrieval-Augmented Generation pipelines or multi-agent systems) attempt to express the nuances of reality through classical probabilities.

In fuzzy logic or in fully probabilistic models, the sum of certainty values must be strictly equal to 1 (or 100%).

If the probability that a statement is true is 0.6, then automatically the probability that it is false is 0.4.

This constraint becomes a limitation in complex AI systems.

Consider an example from AI-assisted medicine:

- An algorithm analyzes a patient's medical record.
- The medical document has been poorly scanned, and several crucial lines are illegible.
- The model returns a diagnostic probability of 0.5.

If the system relies on standard probabilistic logic, it will conclude: "I am 50% certain that the disease is present and 50% certain that it is not."

In reality, however, the diagnosis is not necessarily false; rather, the available information is incomplete.

This is precisely where the neutrosophic triad comes into play.

**3. The (T, I, F) Triplet as a Software Primitive**

> "Absence of evidence is not evidence of absence. Truth and falsehood become complete only when we measure the neutral." — Florentin Smarandache

Neutrosophic logic breaks through this barrier by separating analysis into three completely independent dimensions, which are not required to sum to 1:

1. **T (Truth):** The degree of positive evidence supporting a statement.
2. **I (Indeterminacy):** The degree of ambiguity, ignorance, missing text, or unresolved context.
3. **F (Falsity):** The degree of negative evidence or direct contradiction.

For a programmer, this triplet is not merely an abstract theory; it can become an extremely simple and elegant software contract in Python:

```
from dataclasses import dataclass
@dataclass(frozen=True)
class NeutrosophicTriplet:
    T: float   # Positive support (0.0 -> 1.0)
    I: float   # Indeterminacy / Ambiguity (0.0 -> 1.0)
    F: float   # Contradiction / Negative support (0.0 -> 1.0)
    def _post_init_(self) -> None:
        # Validate that each dimension remains within the unit interval
        for name, value in {"T": self.T, "I": self.I, "F": self.F}.items():
            if not 0.0 <= value <= 1.0:
                raise ValueError(
                    f"Dimension {name} must be between 0.0 and 1.0"
                )
```

*Use code with caution.*

Why is this **NeutrosophicTriplet** object revolutionary for the logic underlying AI systems?

Because it allows us to represent states that cannot be expressed by a simple percentage:

- NeutrosophicTriplet(T=0.0, I=1.0, F=0.0) → **Completely unknown** (we have a blank screen; the data is missing).
- NeutrosophicTriplet(T=0.9, I=0.0, F=0.9) → **Paradox / Severe conflict** (Source A says "YES" with certainty, while Source B says "NO" with certainty).
- NeutrosophicTriplet(T=0.8, I=0.1, F=0.1) → **A stable and well-documented conclusion.**

## 4. Practical Applications in Modern AI Architectures

> "A truly intelligent system does not assume the absence of falsehood where it cannot see; rather, it rigorously measures its own blind spot." — Florentin Smarandache

To understand the value of neutrosophic logic, we must examine it through the lens of production-grade software.

We will analyze two of the most popular architectural patterns in use today:

a) Retrieval systems (RAG – Retrieval-Augmented Generation)
b) Collaborative autonomous agent systems (Swarm)

### A. Context Evaluation in RAG (Retrieval-Augmented Generation) Systems

Systems such as ChatGPT or Gemini often search vector databases to formulate responses.

Suppose a user asks:

"Does Company X provide dental health insurance?"

The RAG algorithm retrieves two text chunks:

1. **Source A:** "Company X offers premium healthcare packages to its employees."
2. **Source B:** "Dental services are not included in Company X's basic package."

Instead of computing a simple average score, we can implement a neutrosophic evaluation function that semantically analyzes both sources:

```
def evaluate_rag_support(
    question: str,
    sources: list[str]
) -> NeutrosophicTriplet:
    # The LLM performs an internal assessment
    # across the three independent dimensions.
    # Source A provides a vague positive clue
    # -> T = 0.3, but leaves room for interpretation
    # -> I = 0.7
    # Source B introduces a clear negation
    # regarding the basic package
    # -> F = 0.8, but still leaves ambiguity
    # concerning the premium package
    # -> I = 0.4
    t_final = 0.3   # Weak positive support
    i_final = 0.7   # High ambiguity/indeterminacy
                    # (we do not know whether the premium
                    # package includes dental coverage)
    f_final = 0.8   # Strong negative evidence
    return NeutrosophicTriplet(
        T=t_final,
        I=i_final,
        F=f_final
    )
```

*Use code with caution.*

**Decision in Code (The Engineering Action)**

Because we have isolated **I = 0.7**, our system will not generate an uncertain answer and thereby avoids hallucination.

Instead, a simple conditional statement routes execution toward a safer action:

```
evaluation = evaluate_rag_support(question, sources)
if evaluation.I > 0.6:
    # Automatically escalate or request clarification,
    # since the context is indeterminate.
    return (
        "The information found is ambiguous. "
        "Please specify whether you are referring "
        "to the basic package or the premium package."
    )
```

*Use code with caution.*

**B. Governance of Multi-Agent and Swarm Systems (AI Swarms)**

In Swarm-style architectures (such as the Aden/Hive framework), a central coordinating agent ("Queen") delegates tasks to multiple specialized sub-agents ("Workers").

Imagine an AI system that analyzes application code to identify security vulnerabilities.

**Agent 1 (Static Analysis)** reports:

"I found no direct vulnerabilities."

(T = 0.0, I = 0.1, F = 0.9)

**Agent 2 (Dynamic Pentesting)** reports:

"Execution was blocked by an internal firewall; I could not complete the test."

(T = 0.0, I = 1.0, F = 0.0)

If the coordinating agent simply averaged the results, it would incorrectly conclude that the application is completely secure.

However, by applying neutrosophic logic, it aggregates the triplets and notices that the **I (Indeterminacy)** component produced by Agent 2 is maximal.

The system immediately understands that **absence of evidence is not evidence of absence.**

Consequently, the coordinating agent makes the correct decision: it does not approve the code. Instead, it launches a third agent within the execution workflow to properly configure the firewall and re-enable dynamic testing.

**5. Why the "I" Dimension (Indeterminacy) Is the Key to AI Engineering**

> "Paradox is the gateway through which science advances. When an algorithm learns to say, 'I don't know,' only then does it take its first step toward true knowledge." — Florentin Smarandache

For a software engineer, the **I** dimension of the neutrosophic triplet represents the true innovation.

In traditional logic, everything that is neither true nor false is often thrown into a gray error zone (a *catch Exception* block).

In AI, however, **I** actively captures:

- Noise in the data (blurred images, corrupted or truncated text).
- Poorly formulated user questions (incomplete or ambiguous prompts).
- Areas where the language model recognizes that it lacks knowledge within its available information base (thereby directly avoiding fabricated answers or hallucinations).

By treating **I** as a native, measurable variable, software architecture becomes genuinely **anti-fragile**.

We can develop regression tests (*anti-drift testing*) that verify whether, after a model update, the level of indeterminacy (**I**) on a fixed dataset has increased or decreased. This provides a much more precise indicator of architectural stability than traditional accuracy metrics.

## 6. Decision Matrix: A Comparison of Logical Paradigms

| Logical Paradigm | Representation in Code | Mathematical Constraints | What Happens in Case of Conflict or Missing Data? |
|---|---|---|---|
| **Binary Logic** | bool (True / False) | Excludes a third possibility ($A \vee \neg A$) | Fails or throws an exception. It cannot handle contextual nuances. |
| **Fuzzy Logic** | float (from 0.0 to 1.0) | The sum of probabilities must be strictly equal to 1 (100%) | Flattens information. Severe conflict and complete absence of data receive the same ambiguous average score (e.g., 0.5). |
| **Neutrosophic Logic** | Triplet (T, I, F) (three values) | Each dimension independently belongs to the interval [0.0, 1.0] | Isolates the problem. Clearly distinguishes between what has been refuted (**F**) and what is simply unknown, ambiguous, or contradictory (**I**). |

## 7. Conclusion: The Future of AI Is Not Binary, It Is Ternary

The connection between neutrosophic logic and artificial intelligence is no longer merely a topic of abstract academic research; it is becoming an urgent necessity in production engineering as of 2026.

As AI systems evolve from simple text generation to the autonomous execution of complex decisions, compressing uncertainty into a single scalar value becomes an increasingly risky practice.

By translating Smarandache's mathematical triad—Truth, Indeterminacy, and Falsity—into concrete data structures, software engineers gain a powerful new tool.

This tool enables them to model reality with far greater precision, transforming ambiguity from a fatal system error into a controllable variable.

The discipline of neutrosophic software engineering can begin today, within our existing binary systems, through the adoption of a simple data contract and a profoundly improved routing logic.

## Appendix A: Key Concepts

### Scalar Values

In mathematics and physics, **scalar values**, or simply **scalars**, are quantities that are completely defined by a numerical value and a unit of measurement. Unlike vectors, scalars have neither direction nor orientation.

**Characteristics of Scalar Values**

- **Simple Definition:** They are expressed by a single real number.
- **Spatial Independence:** They remain unchanged regardless of the coordinate system chosen.
- **Mathematical Operations:** They obey simple algebraic rules (addition, subtraction, multiplication, and division).

**Common Examples of Scalar Quantities**

- **Time:** Expressed in seconds, hours, or days (e.g., $t = 10\ s$).
- **Mass:** The amount of matter contained in a body (e.g., $m = 5\ kg$).
- **Temperature:** The degree of thermal agitation (e.g., $T = 25°C$).
- **Length / Distance:** The distance traveled between two points (e.g., $d = 100\ m$).
- **Energy / Work:** The capacity of a system to perform mechanical work (e.g., $E = 200\ J$).

### Fuzzy Logic

**Fuzzy logic**, also known as **graded logic**, is a form of mathematical logic that allows the modeling of nuances and imprecise concepts. Unlike traditional Boolean logic, where everything is strictly **True (1)** or **False (0)**, fuzzy logic accepts intermediate values between 0 and 1. This approach mimics human reasoning, in which decisions are not made solely in terms of "black and white" but involve various "shades of gray."

**The Fundamental Difference: Classical vs. Fuzzy Logic**

Imagine that you want to define whether a person is “tall.”

**Classical (Boolean) Logic:** A fixed threshold is established, for example, 1.80 m. A person who is 1.79 m tall is considered “short” (value = 0), while a person who is 1.80 m tall is considered “tall” (value = 1).

**Fuzzy Logic:** A gradual transition is allowed. A person who is 1.75 m tall may be considered tall to a degree of 0.4, a person who is 1.80 m tall to a degree of 0.8, and a person who is 1.90 m tall to a degree of 1.0.

**Key Concepts in Fuzzy Logic**

**Degree of Membership**

A numerical value between 0 and 1 indicating the extent to which an element belongs to a category.

**Membership Functions**

Graphical representations (usually triangular or trapezoidal shapes) that map real-world data into fuzzy values.

**If–Then Rules**

Simple instructions that define the system’s logic.

Example: If the temperature is high, then the fan speed is high.

**Real-World Applications**

Fuzzy logic is widely used in automation systems and smart appliances:

1. **Smart Washing Machines:** Detect how dirty clothes are and adjust water consumption and washing time instead of relying on rigid programs.
2. **Air Conditioning Systems:** Finely adjust compressor power according to the exact room temperature to save energy.
3. **Automotive ABS Systems:** Regulate brake pressure according to wheel slip, ensuring smoother stopping.
4. **Camera Autofocus Systems:** Quickly identify image sharpness using nuanced contrast criteria.

**Advantages and Disadvantages**

**Advantages**

- Allows computers to work with imprecise verbal descriptions from the real world (e.g., “warm,” “fast,” “cheap”).
- The resulting systems are robust and often simpler than those based on complex mathematical models.

**Disadvantage**

- The initial configuration of fuzzy rules requires human validation and domain expertise.

**Neutrosophic Logic**

**Neutrosophic logic** is a revolutionary extension of mathematical logic that analyzes a proposition through three independent components: **Truth (T), Falsity (F), Indeterminacy (I).**

This branch of logic was founded in 1995 by the Romanian-American mathematician and writer **Florentin Smarandache** as part of a new philosophical movement called **Neutrosophy**, meaning “the study of neutralities.” Neutrosophic logic represents a direct generalization of fuzzy logic. While fuzzy logic measures only the transition between truth and falsehood, neutrosophic logic formally introduces the middle component—**uncertainty, paradox, or neutrality**.

**The Three Fundamental Components (T, I, F)**

Every idea or proposition in neutrosophic logic is evaluated as a triplet:

1. **T (Truth):** The degree of truth of the proposition.
2. **I (Indeterminacy):** The degree of indeterminacy, neutrality, uncertainty, or ignorance.
3. **F (Falsity):** The degree of falsity of the proposition.

In practical applications (such as engineering or artificial intelligence), these three components are real numbers within the classical interval **[0, 1]**.

**Complete Freedom of the Sum (Unlike Other Logics)**

Unlike classical intuitionistic logic, where the sum of the components must always equal 1

$$(T + I + F = 1),$$

in neutrosophic logic the three components are completely independent.

As a result, their sum may be:

- **Less than 1** (incomplete information).
- **Equal to 1** (complete and exact information).
- **Greater than 1** (contradictory or overlapping information).

**Intuitive Example: Political Elections**

If we analyze a voting process in a country, the logical approaches differ significantly.

**Boolean Logic (0 or 1)**

You can vote only **YES** or **NO**. No nuances exist.

**Fuzzy Logic (0 to 1)**

A voter supports **60% YES** and **40% NO**.

**Neutrosophic Logic (T, I, F)**

When all votes are counted, we may observe:

- **T = 0.5** (50% of people voted YES)
- **F = 0.3** (30% of people voted NO)
- **I = 0.2** (20% are undecided, spoiled their ballots, or did not vote)

The sum is:

$$T + I + F = 0.5 + 0.2 + 0.3 = 1.0$$

If contradictory information or incomplete data emerges from polling stations, the sum may vary beyond unity.

**Fields of Application**

Neutrosophic logic is a powerful mathematical tool with extensive applications in modern technologies:

1. **Artificial Intelligence and Big Data:** It helps algorithms make sound decisions when sensor data is partially corrupted, incomplete, or contradictory.
2. **Decision Support Systems:** Used in economics and medicine to evaluate financial risks or complex medical diagnoses involving numerous uncertain parameters.
3. **Data Fusion:** Combines information from multiple sources that may partially contradict one another.
4. **Robotics:** Enables commercial and industrial robots to navigate dynamic environments where obstacles must be evaluated approximately rather than with absolute certainty.

## References

1. Smarandache, Florentin. *A Unifying Field in Logics: Neutrosophic Logic, Neutrosophy, Neutrosophic Set, Neutrosophic Probability and Statistics*. 6th ed. Phoenix, AZ: InfoLearnQuest, 2007. Accessed June 3, 2026. https://fs.unm.edu/eBook-Neutrosophics6.pdf
2. Smarandache, Florentin. 2013. *Introduction to Neutrosophic Measure, Neutrosophic Integral, and Neutrosophic Probability*. Craiova, Romania: Sitech & Education Publishing. Accessed June 3, 2026. https://fs.unm.edu/NeutrosophicMeasureIntegralProbability.pdf
3. Smarandache, Florentin. *Introduction to Neutrosophic Statistics*. Craiova, Romania: Sitech & Education Publishing, 2014. Accessed June 4, 2026. https://fs.unm.edu/NeutrosophicStatistics.pdf.
4. Smarandache, Florentin. 2016. *Neutrosophic Overset, Neutrosophic Underset, and Neutrosophic Offset: Similarly for Neutrosophic Over-/Under-/Off-Logic, Probability, and Statistics*. Brussels: Pons Editions. Accessed June 4, 2026. https://fs.unm.edu/NeutrosophicOversetUndersetOffset.pdf
5. Wang, Haibin, Florentin Smarandache, Yan-Qing Zhang, and Rajshekhar Sunderraman. 2005. *Interval Neutrosophic Sets and Logic: Theory and Applications in Computing*. Neutrosophic Book Series, no. 5. Phoenix, AZ: Hexis. Accessed June 4, 2026. https://fs.unm.edu/INSL.pdf
6. Smarandache, Florentin. *Neutrosophic Set—A Generalization of the Intuitionistic Fuzzy Set*. Gallup, NM: University of New Mexico, Gallup Campus, n.d. PDF manuscript. Accessed June 4, 2026. https://fs.unm.edu/IFS-generalized.pdf
7. Smarandache, Florentin. "Definitions Derived from Neutrosophics." *Multiple-Valued Logic: An International Journal* 8, no. 5–6 (2002): 591–604. Accessed June 5, 2026. https://fs.unm.edu/DefinitionsDerivedFromNeutrosophics.pdf

4. Smarandache, Florentin. "Degree of Dependence and Independence of the (Sub)Components of Fuzzy Set and Neutrosophic Set." *Neutrosophic Sets and Systems* 11 (2016): 95–97. Accessed June 5, 2026. https://fs.unm.edu/NSS/DegreeOfDependenceAndIndependence.pdf
5. Smarandache, Florentin. 2016. "Operators on Single-Valued Neutrosophic Oversets, Neutrosophic Undersets, and Neutrosophic Offsets." *Journal of Mathematics and Informatics* 5: 63–67. Published June 29, 2016. Accessed June 5, 2026. https://fs.unm.edu/SVNeutrosophicOverset-JMI.pdf
6. Smarandache, Florentin. 2016. "Interval-Valued Neutrosophic Oversets, Neutrosophic Undersets, and Neutrosophic Offsets." *International Journal of Science and Engineering Investigations* 5 (54): 1–4. Accessed June 6, 2026. https://fs.unm.edu/IV-Neutrosophic-Overset-Underset-Offset.pdf
7. Smarandache, F., & Dezert, J. (2004–2015). *Advances and Applications of DSmT for Information Fusion* (Vols. 1–5). American Research Press. Accessed June 5-6, 2026. https://fs.unm.edu/DSmT.htm
8. Smarandache, F., & Vlăduțescu, Ș. (2013). *Neutrosophic Emergences and Incidences in Communication and Information*. LAP LAMBERT Academic Publishing. Accessed June 6, 2026. https://fs.unm.edu/LAP-NeutrosophicEmergencies.pdf
9. Smarandache, Florentin. *Plithogeny, Plithogenic Set, Logic, Probability, and Statistics*. Pons, 2017. Accessed June 6, 2026. https://fs.unm.edu/Plithogeny.pdf
10. Smarandache, Florentin. "Refined Neutrosophic Crisp Set (RNCS)." In *Nidus Idearum, Vol. VII*, 3rd ed., 114–116. Pons, 2019. Accessed June 6, 2026. https://fs.unm.edu/RefinedNeutrosophicCrispSet.pdf
11. Smarandache, Florentin, & Andrușa R. Vătuiu. *Evoluție Neutrosofică Umană în Spirală sau Divinul este în Om / Human Neutrosophic Evolution in Spiral or The Divine is in the Man*. Kalendarium, 2019. Accessed June 6, 2026. https://fs.unm.edu/SpiralNeutrosophicEvolution.pdf
12. Peng, Xindong, and Jingguo Dai. "A Bibliometric Analysis of Neutrosophic Set: Two Decades Review from 1998 to 2017." *Artificial Intelligence Review* 53, no. 1 (2020): 199–255. Accessed June 6, 2026. https://doi.org/10.1007/s10462-018-9652-0.


Received: Dec 9, 2025. Accepted: May 30, 2026