AIMS Mathematics

AIMS Mathematics, 10(11): 28004–28019.
DOI: 10.3934/math.20251230
Received: 21 July 2025
Revised: 23 September 2025
Accepted: 23 October 2025
Published: 28 November 2025

https://www.aimspress.com/journal/Math

Research article

Machine learning-based automated theorem verification of nano-semi-weakly generalized closed sets in nano-topological spaces

S. Sathya Priya[1,*], N. Nagaveni[1], R. Lavanya[1], R. Saveeth[2]

[1] Department of Mathematics, Coimbatore Institute of Technology, Coimbatore 641014, Tamil Nadu, India
[2] Department of Computer Science & Engineering, Coimbatore Institute of Technology, Coimbatore 641014, Tamil Nadu, India

* **Correspondence:** Email: sathyapriya.s@cit.edu.in.

Abstract: This study introduces and investigates a new class of sets, termed nano-semi-weakly generalized closed sets (NSWG-CS), within the framework of nano-topological spaces (NTS). Their key properties, formal definitions, and relationships with other generalized closed sets are examined. To bridge theoretical insights with computational applications, we present a machine learning (ML)-based automated theorem verification system. A graph neural network model (GNN) is implemented to classify and validate the NSWG-CS by leveraging structured representations of subset relations. The model is trained on a synthetic dataset and achieves an accuracy of 86.7%, an F1-score of 85.4%, a recall of 83.2% and precision of 87.8%, demonstrating reliable and realistic performance. These findings highlight the feasibility of applying ML techniques to verify mathematical properties within nano-topological structures. The integration of nano-topology and artificial intelligence contributes to the broader field of computational mathematics and automated theorem verification.

Keywords: nano-topological spaces; nano-semi-weakly generalized closed sets; machine learning; graph neural networks; automated theorem verification
Mathematics Subject Classification: 54A40, 68T07

1. Introduction

Research on generalized closed sets (GCs) in topological spaces has advanced significantly over the years. Levine initiated this line of study by introducing generalized closed sets [10]. In 1965, Njastad introduced another specific class of open sets, contributing to the foundational understanding of topological spaces (TS) [16]. Similarly, Biswas introduced semi-open sets [3] while Long et al. examined the properties of regular closed sets [12]. Mashhour et al. further enriched the field by studying pre-open sets, advancing the development of topological concepts [14].

Lellis Thivagar introduced the framework of nano-topological spaces (NTS), where sets are described using lower and upper approximations. In this settings, nano open sets (NOS) play a fundamental role, and further variations such as nano-semi-open sets (NSOS) and nano-pre-open sets (NPOS) have been explored [9]. Nagaveni later introduced weakly generalized closed (WGC) sets in general topological spaces [18] and subsequently extended the concept to NTS [15].

This paper introduces semi-weakly generalized closed sets (SWGCS) in NTS and examines their key properties. Manually verifying theorems in this area is both labor-intensive and prone to human error. With the rapid progress in artificial intelligence, particularly machine learning (ML) [4], new techniques have emerged to automate and enhance the accuracy of theorem verification. This study leverages these techniques to develop an intelligent system capable of determining whether a given subset satisfies the properties of nano-semi-weakly generalized closed sets (NSWG-CS).

2. Preliminaries

2.1. Basic definitions

This section summarizes the essential concepts and attributes that are pertinent to this investigation.

Definition 2.1. [10] A subset $\mathcal{A}$ of a TS $(\mathcal{X},\tau)$ is called WGC if $c\ell(int(\mathcal{A})) \subseteq \mathcal{U}$, whenever $\mathcal{A} \subseteq \mathcal{U}$ and $\mathcal{U}$ is open.

Definition 2.2. [18] A subset $\mathcal{A}$ of a TS $(\mathcal{X},\tau)$ is called SWGC if $c\ell(int(\mathcal{A})) \subseteq \mathcal{U}$, where $\mathcal{A} \subseteq \mathcal{U}$ and $\mathcal{U}$ is semi-open.

Definition 2.3. [2] Let $(\mathcal{U},\ \tau_{\mathcal{R}}(\mathcal{X}))$ be a NTS and $\mathcal{A}$ be any subset of $(\mathcal{U},\ \tau_{\mathcal{R}}(\mathcal{X}))$. Then, $\mathcal{A}$ is said to be nano-generalized closed set (NG-CS) if $\mathcal{N}c\ell(\mathcal{A}) \subseteq \mathcal{A}$, where $\mathcal{A} \subseteq V$ and V is nano-open in $(\tau_{\mathcal{R}}(\mathcal{X}), \mathcal{U})$.

Definition 2.4. [15] Let $(\mathcal{U},\ \tau_{\mathcal{R}}(\mathcal{X}))$ be a nano-topological space. A subset $\mathcal{A}$ of $(\mathcal{U},\ \tau_{\mathcal{R}}(\mathcal{X}))$ is called nano-weakly generalized closed (NWG-CS) set if $\mathcal{N}c\ell\big(\mathcal{N}int(\mathcal{A})\big) \subseteq V$, where $\mathcal{A} \subseteq \mathcal{V}$ and $\mathcal{V}$ is nano-open. The complement of a nano-weakly generalized closed set is a nano-weakly generalized open set.

2.2. Machine learning approach for theorem verification

To develop an efficient theorem verification system, we employ graph neural networks (GNNs), which are well-suited for capturing relationships between elements in a topological structure. The main components of our approach include the following:

- Graph representation: Nano-semi-weakly generalized closed subsets and their relationships are modeled as a directed graph, where the nodes represent subsets, and the edges depict inclusion relationships.
- Feature engineering: Each node is assigned a unique feature vector to distinguish different subset properties.
- Training and optimization: The GNN model is trained on synthetic datasets, with cross-entropy loss and the Adam optimizer to enhance learning efficiency.
- Evaluation metrics: We assess the model using the accuracy, F1-score, and recall to determine its effectiveness in theorem verification.

2.2.1. Input details

The input for this theorem verification task is a graph representation of the subsets in a nano-topological space.

2.2.2. Graph representation of NSWG-CS

- **Nodes:** These represent different subsets in the nano-topological space, e.g., $\{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{F}\}$.
- **Edges:** These represent relationships between subsets, such as inclusion relations, e.g., $\{(\mathcal{A},\mathcal{B}), (\mathcal{A},\mathcal{C}), (\mathcal{B},\mathcal{D}), (\mathcal{C},\mathcal{D}), (\mathcal{D},\mathcal{E}), (\mathcal{E},\mathcal{F})$ and $(\mathcal{C},\mathcal{F})\}$.
- **Node features:** $\mathcal{A}$, a one-hot encoded identity matrix representing each node.

The graph structure used in our model is shown in Figure 1.

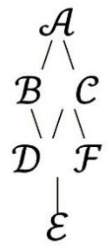

Figure 1. Graph used in the model.

This graph is processed using GNNs to learn the structural patterns.

3. Nano-semi weakly generalized closed sets

This section introduces NSWG-CS and examines some of their key characteristics.

Definition 3.1. Let $\left(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X})\right)$ be a NTS. A subset $\mathcal{A}$ of $\left(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X})\right)$ is called a nano-semi-weakly generalized closed set, if $\mathcal{N}c\ell(\mathcal{N}\, int(\mathcal{A}) \subseteq \mathcal{V}$ whenever $\mathcal{A} \subseteq \mathcal{V}$ and $\mathcal{V}$ is nano-semi-open.

Definition 3.2. A subset $\mathcal{A}$ of a space $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$ is called nano-regular generalized weakly closed (NRGW-CS) if $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{U}$ whenever $\mathcal{A} \subseteq \mathcal{U}$ and $\mathcal{U}$ is nano-regular semi-open in $(\mathcal{U}, \tau_{\mathcal{R}}(\mathcal{X}))$.

Remark 3.1. The union of two NSWG-CS need not be an NSWG-CS.

The statement above is true, as seen from the example given below.

Example 3.1. Let $\mathcal{U} = \{a, b, c, d\}$ with $\mathcal{U}/_{\mathcal{R}} = \{\{b\}, \{c\}, \{a, d\}\}$ and $\mathcal{X}$=$\{a,c\}$, then the nano topology is $\tau_{\mathcal{R}}(\mathcal{X}) = \{\mathcal{U}, \emptyset, \{c\}, \{a, d\}, \{a, c, d\}\}$. Let $\mathcal{A} = \{a\}$ and $\mathcal{B} = \{d\}$, where $\mathcal{A}$ and $\mathcal{B}$ are NSWG-CS but their union $\mathcal{A} \cup \mathcal{B} = \{a, d\}$ is not an NSWG-CS.

Theorem 3.1. If $\mathcal{A}$ is a nano-closed subset of a NTS $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$, then $\mathcal{A}$ is an NSWG-CS.

Proof. Let $\mathcal{A} \subseteq \mathcal{U}$ be nano-closed. By definition, $\mathcal{N}c\ell(\mathcal{A}) = \mathcal{A}$. Let $\mathcal{V}$ be any nano-semi-open set such that $\mathcal{A} \subseteq \mathcal{V}$. Since $\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{A}$, it follows that $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{N}c\ell(\mathcal{A}) = \mathcal{A} \subseteq \mathcal{V}$. Thus, the condition $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{V}$ is satisfied for every nano-semi-open superset $\mathcal{V}$ of $\mathcal{A}$, proving that $\mathcal{A}$ is an NSWG-CS.

Remark 3.2. The following example illustrates that the converse of the theorem above does not necessarily hold.

Example 3.2. Let $\mathcal{U} = \{a, b, c, d\}$ with$\mathcal{X} = a, c\}$, and $\tau_R(\mathcal{X}) = \{\mathcal{U}, \emptyset, \{a\}, \{c, d\}, \{a, c, d\}\}$ in this NTS $\left(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X})\right)$, in which case the set $\{c\}$ is NSWG-CS but it is not nano-closed.

Theorem 3.2. Every nano-weakly generalized closed subset of a NTS $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$ is an NSWG-CS.

Proof. Let $\mathcal{A} \subseteq \mathcal{U}$ be NWG-CS. Then, by definition, $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{A}$. Now take any nano-semi-open set $\mathcal{V}$ such that $\mathcal{A} \subseteq \mathcal{V}$. From the inclusion above and the fact that $\mathcal{A} \subseteq \mathcal{V}$, we obtain $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{A} \subseteq \mathcal{V}$. Thus, for every nano-semi-open superset $\mathcal{V}$ of $\mathcal{A}$, the relation $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})) \subseteq \mathcal{V}$ holds. Therefore, $\mathcal{A}$ satisfies the condition for being an NSWG-CS.

Remark 3.3. Every NSWG-CS need not be an nano-weakly generalized closed set (NWG-CS).

Example 3.3. Let $\mathcal{U} = \{a, b, c, d\}$ with $\mathcal{X} = \{a, c\}$, and $\tau_R(\mathcal{X}) = \{\mathcal{U}, \emptyset, \{a\}, \{c, d\}, \{a, c, d\}\}$ in this NTS $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$. The set $\{a, d\}$ is an NSWG-CS not an NWG-CS.

Theorem 3.3. Let $\mathcal{A}$ be a nano-generalized closed set of an NTS $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$, then $\mathcal{A}$ is an NSWG-CS.

Proof. If $\mathcal{A}$ is nano-generalized closed, then we have $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{F}$ for every nano-open set $\mathcal{F}$ with $\mathcal{A} \subseteq \mathcal{F}$. Since every nano-open set is also nano-semi-open, it follows that the same condition holds for all nano-semi-open supersets of $\mathcal{A}$. Therefore, $\mathcal{A}$ satisfies the defining property of NWG-CS.

Example 3.4. Let $\mathcal{U} = \{a, b, c, d\}$ with $\mathcal{X} = \{a, c\}$ and $\tau_R(\mathcal{X}) = \{\mathcal{U}, \emptyset, \{a\}, \{c, d\}, \{a, c, d\}\}$ in this NTS $(\mathcal{U}, \tau_{\mathcal{R}}(\mathcal{X}))$, the set $\{c\}$ is an NSWG-CS but it is not an NG-CS.

Theorem 3.4. Every nano-pre-closed set in an NTS $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$ is an NSWG-CS.

Proof. Assume that $\mathcal{A}$ be a nano-pre-closed set in an NTS $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$. Let $\mathcal{V}$ be a nano-open set in $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$ such that $\mathcal{A} \subseteq \mathcal{V}$. By assumption, $\mathcal{N}c\ell\left(\mathcal{N}int(\mathcal{A})\right) \subseteq \mathcal{A}$. Then $\mathcal{N}c\ell\left(\mathcal{N}int(\mathcal{A})\right) \subseteq \mathcal{A} \subseteq \mathcal{V}$. Since every nano-open set is nano-semi open set, $\mathcal{A}$ is an NSWG-CS.

Example 3.5. Let $\mathcal{U} = \{a, b, c, d\}$ with $\mathcal{X} = \{a, c\}$, and $\tau_R(\mathcal{X}) = \{\mathcal{U}, \emptyset, \{a\}, \{c, d\}, \{a, c, d\}\}$ in this NTS$(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X}))$. The set $\{a, d\}$ is NSWG-CS but it is not a nano-pre-closed set.

Theorem 3.5. If $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) - \mathcal{A}$ contains no nonempty semi-closed set, then $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) = \mathcal{A}$ and hence $\mathcal{A}$ is nano-regular-closed.

Proof. By assumption, the set $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})) - \mathcal{A}$ does not contain any nonempty semi-closed subset. This means that $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{A}$. But since $\mathcal{A} \subseteq \mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}))$ always holds, we obtain the equality $\mathcal{N}c\ell\left(\mathcal{N}int(\mathcal{A})\right) = \mathcal{A}$. Therefore, $\mathcal{A}$ is nano-regular-closed.

Theorem 3.6. If $\mathcal{A}$ is an NSWG-CS and $\mathcal{F}$ is closed, then $\mathcal{A} \cap \mathcal{F}$ is an NSWG-CS.
Proof. Let $\mathcal{V}$ be a nano-semi-open set with $\mathcal{A} \cap \mathcal{F} \subseteq \mathcal{V}$. Since $\mathcal{F}$ is closed, $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A} \cap \mathcal{F}))\subseteq\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})) \cap \mathcal{F}$. Because $\mathcal{A}$ is an NSWG-CS, we know that $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A} \cap \mathcal{F})) \subseteq \mathcal{V} \cup (\mathcal{U}\backslash\mathcal{F})$. Intersecting with $\mathcal{F}$, we obtain $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A} \cap \mathcal{F})) \subseteq \mathcal{V}$. Thus, $\mathcal{A} \cap \mathcal{F}$ satisfies the condition for being an NSWG-CS.

Theorem 3.7. If $\mathcal{A}$ is NSWG-CS and $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{N}c\ell\big(\mathcal{N}int(\mathcal{A})\big)$, then $\mathcal{B}$ is an NSWG-CS.

Proof. Since $\mathcal{B} \subseteq \mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})$, we have $\mathcal{N}int(\mathcal{B}) \subseteq Nint(A)$. Taking closures gives $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{B}) \subseteq \mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})$. Because $\mathcal{A}$ is an NSWG-CS, we know that for any nano-semi-open set $\mathcal{V}$ with $\mathcal{A} \subseteq \mathcal{V}$, $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})) \subseteq \mathcal{V}$. Since $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$, it follows that $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}) \subseteq \mathcal{V}$. Hence $\mathcal{B}$ is an NSWG-CS.
Theorem 3.8. Every NSWG-CS subset of an NTS $(\mathcal{U}, (\tau_{\mathcal{R}}(\mathcal{X})\,)$ is nano-regular generalized weakly closed (NRGW-CS) set but the converse is not true.
Proof. Let $\mathcal{A} \subseteq \mathcal{U}$ be an NSWG-CS. Take any nano-regular semi open set $\mathcal{V}$ with $A \subseteq \mathcal{V}$. Since every nano-regular semi-open set is also nano-semi-open, the NSWG-CS property implies that $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})) \subseteq \mathcal{V}$. This is precisely the condition for $\mathcal{A}$ to be nano-regular weakly closed. However, there are nano-regular weakly closed sets which are not NSWG-CS. Hence, the converse fails.
Theorem 3.9. If a subset $\mathcal{A}$ of a topological space $\mathcal{X}$ is pre-closed, then it is nano-semi-weakly generalized closed but not vice versa.
Proof. Suppose $\mathcal{A}$ is pre-closed. Then by definition, $\mathcal{A} \supseteq \mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}))$. Let $\mathcal{V}$ be any nano-semi-open set with $\mathcal{A} \subseteq \mathcal{V}$. From the inclusion above, we have $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})) \subseteq \mathcal{A} \subseteq \mathcal{V}$. Thus, $\mathcal{A}$ satisfies the definition of an NSWG-CS. This converse fails in general.
Theorem 3.10. If a subset $\mathcal{A}$ of a topological space $\mathcal{X}$ is α-closed, then it is an NSWG-CS but the converse need not be true.
Proof. Assume that $\mathcal{A}$ is α-closed. Then, by definition, $\mathcal{A} \supseteq \mathcal{N}c\ell\Big(\mathcal{N}int\big(\mathcal{N}c\ell(\mathcal{A})\big)\Big)$. Let $\mathcal{V}$ be a nano-semi-open set with $\mathcal{A} \subseteq \mathcal{V}$. Since $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A})) \subseteq \mathcal{A} \subseteq \mathcal{V}$, it follows that $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}))\subseteq \mathcal{V}$. Therefore, $\mathcal{A}$ is an NSWG-CS. The converse is not valid in general.
Theorem 3.11. If a subset $\mathcal{A}$ of a topological space $\mathcal{X}$ is regular closed, then it is nano-semi-weakly generalized closed, but not conversely.
Proof. Suppose $\mathcal{A}$ is regular-closed. Then $\mathcal{A} = \mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}))$. Let $\mathcal{V}$ be a nano-semi-open set such that $\mathcal{A} \subseteq \mathcal{V}$. From the equality above, we get $\mathcal{N}c\ell(\mathcal{N}int(\mathcal{A}))= \mathcal{A} \subseteq \mathcal{V}$. Hence, $\mathcal{A}$ is NSWG-CS. Vice versa, not every NSWG-CS is regular-closed.
Example 3.6. Let $\mathcal{U} = \{a, b, c, d\}$ with $\mathcal{X} = \{a, c\}$ and $\tau_R(\mathcal{X}) = \big\{\mathcal{U}, \emptyset, \{a\}, \{c, d\}, \{a, c, d\}\big\}$ in this NTS $(\mathcal{U}, \tau_R(\mathcal{X}))$, then the set $\{c\}$ is an NSWG-CS but it is not an nano-generlised regular closed set (NG-RCS).
Remark 3.4. From the discussion and results above, the relationship between the nano-weakly generalized closed sets and existing sets are as shown in Figure 2.

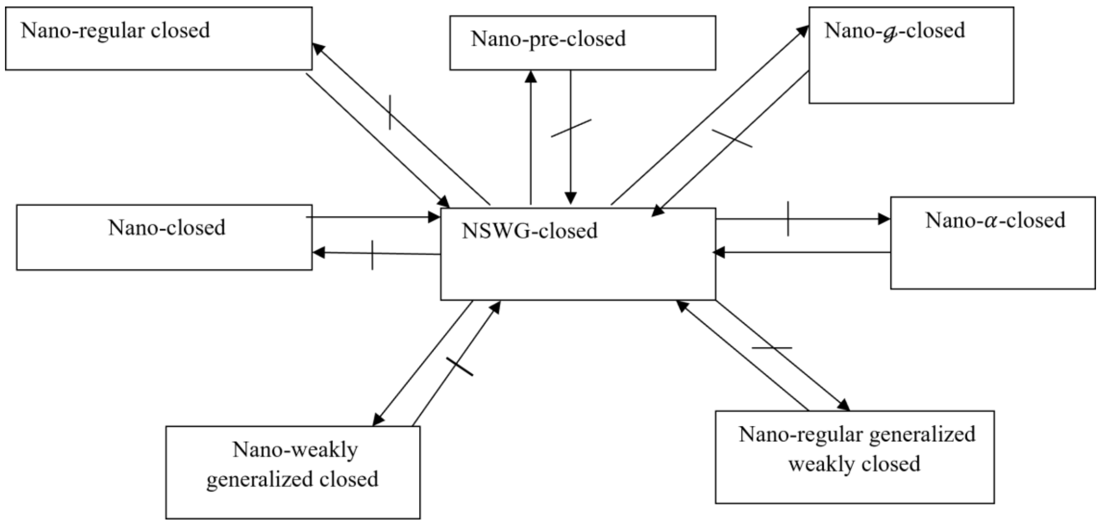

Figure 2. The relationship between the nano-weakly generalized closed sets and the existing sets.

4. Machine learning methodology

4.1. Synthetic data generation for theorem verification

Due to the lack of publicly available datasets for nano-topological theorem verification, a synthetic dataset was generated to train and evaluate the graph convolutional network (GCN) model. The dataset was constructed by simulating NTS and encoding logical relationships among subsets to reflect characteristics of NSWG-CS.

4.1.1. Structure of the synthetic data

Each sample in the dataset represents a finite nano-topological space (U,τ), modeled as a directed graph $G = (V, E)$, where the nodes (V) correspond to subsets of the universe U. The edges (E) represent subset relations (e.g., $A \subseteq B$). the node features include one-hot encoded identifiers and additional logical attributes (e.g., whether a set is nano-open, nano-semi-open, or satisfies certain closure properties).

4.1.2. Labeling strategy

Each graph was labeled as follows:

1 (True): If the target subset $\mathcal{A}$ in the space satisfies the NSWGC property (i.e., $A \subseteq S$, where S is nano-semi-open and $cl(A) \subseteq S$).

0 (False): If the NSWGC condition does not hold for the given configuration.

The labeling process was automated using Python scripts that encoded logical rules derived from the formal mathematical definitions of NSWG-CS. Approximately 10,000 samples were generated to ensure a balanced distribution of positive and negative cases.

4.1.3. Diversity of graphs

To ensure generalizability of the model, the universe size |U| varied between 5 and 15 elements. Topological structures included combinations of open, semi-open, and regular semi-open sets. Subset relationships were randomly generated while maintaining logical consistency.

4.1.4. Advantages of synthetic generation

- **Scalability**: It allows generation of large, diverse training sets for deep learning models.
- **Control**: It enables precise construction of edge cases and theoretical corner conditions.
- **Reproducibility**: It provides full control over the logical rules applied during graph construction.

This synthetic data pipeline enables effective training of the GCN model for theorem classification while laying the groundwork for future extensions using real-world mathematical datasets or symbolic reasoning systems.

4.1.5. Limitations and distribution gap

Although synthetic data provide a flexible and scalable approach for training models on NSWG-CS verification, it may not entirely mirror the intricacies of real mathematical structures derived from formal proofs. Actual topological constructs often involve deeper logical hierarchies, richer interdependencies, and subtler boundary behaviors that artificial datasets might overlook. As a result, despite adhering to mathematically sound rules, the synthetic graphs used in this study may fall short in capturing the full diversity and complexity found in human-developed proofs. Addressing this limitation is an important step forward. Future work may involve augmenting the dataset with examples sourced from formal theorem repositories such as Lean or Coq, applying transfer learning from symbolic reasoning frameworks, or injecting tailored structural noise and edge-case scenarios to bridge the realism gap.

4.2. GCN architecture and training process

To perform automated classification of NSWG-CS, a GCN model was designed and implemented, as shown in Figure 3. GCNs are particularly well-suited for tasks that require learning over graph-structured data, such as set inclusion hierarchies within NTS.

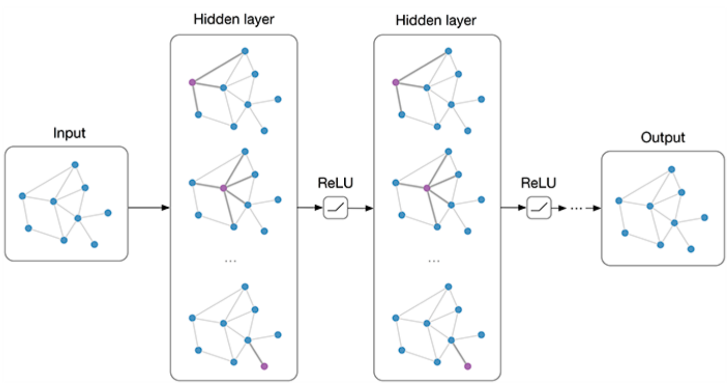

Figure 3. Architecture of the GCN.

4.2.1. Model overview

The model processes each input graph ($G = (V, E)$) with node features (X) and learns to predict whether a given subset of nodes satisfies the NSWGC property. The GCN passes the message across the graph, aggregating the features from neighboring nodes to capture the topological and logical dependencies.

4.2.2. Architecture details

The GCN consists of the following layers:

Input Layer: This accepts one-hot encoded feature vectors representing subsets.

GCN Layer 1: This projects input to a 16-dimensional hidden space using ReLU activation.

GCN Layer 2: This further refines the representation into another 16-dimensional space.

Output Layer: This applies a final GCN layer with a soft max activation to predict binary class labels (0 or 1).

Algorithm 1. GCN training on the subset relation graph

Input:

- Set of nodes VVV defined by subsets
- Corresponding node labels YYY
- Learning rate η, weight decay λ, number of training epochs TTT

Output:

- Trained GCN model *M*

1. **Graph construction:**
   a. Initialize graph $G = (V, E)$
   b. For each pair of nodes (vi,vj)∈V×V, where i≠j:
   i. If $v_i \subset v_j$, then add a directed edge (i,j) to E
2. **Node feature encoding:**
   a. Generate a one-hot encoded feature matrix:
   X←OneHotEncode(V)
3. **Model initialization:**
   a. Set the input dimension: din←dim(X)
   b. Set the hidden dimension: dhidden←16
   c. Set the output dimension: dout←2
   d. Initialize the GCN model:
   M←GCN(din,dhidden,dout,)
4. **Loss function and optimizer setup:**
   a. Set the loss function: *L←CrossEntropyLoss()*
   b. Set the optimizer:
   O←Adam(M.parameters(),η,λ)
5. **Training loop:**
   For epoch=1 to T **do**
   a. Set *M* to training mode

Continued on next page

b. Zero gradients: O.zero_grad()
c. Forward pass: Y^←M(G,X)
d. Compute loss: J←L(Y^,Y)
e. Backpropagation: J.backward()
f. Update model weights: O.step()
End For
6. **Return:** Trained GCN model M

4.2.3. Hyperparameter configuration

The hyper parameter settings used in training the GCN model are summarized in Table 1.

Table 1. Hyperparameter configuration.

Hyperparameter	Value	Description
Learning rate	0.005	Step size during optimization
Weight decay	5e-4	L2 regularization to prevent overfitting
Hidden layer size	16	Number of units in each GCN layer
Dropout rate	0.5	Dropout applied to prevent overfitting
Epochs	150	Total training iterations
Optimizer	Adam	Adaptive optimization algorithm
Loss function	Cross-entropy loss	Classification loss metric

4.3. Performance metrics and evaluation

To assess the effectiveness of the proposed GCN model in verifying NSWG-CS properties, several standard classification metrics were used. These metrics provide a comprehensive view of the model's accuracy, reliability, and predictive power.

4.3.1. Evaluation metrics

- **Accuracy**: The proportion of correctly predicted instances among the total number of predictions.
- **F1-score**: The harmonic mean of precision and recall, used to balance false positives and false negatives.
- **Recall (sensitivity)**: The ability of the model to correctly identify true positive instances (correctly classified NSWG-CS).
- **Precision**: The fraction of relevant instances among the retrieved ones (how many predicted NSWG-CS are truly valid).

These metrics were computed after splitting the dataset into 80% training and 20% test sets. The trained model yielded the following results.

The classification performance of the proposed model is detailed in Table 2, showing an accuracy of 86.7% and an F1-score of 85.4%.

Table 2. Metrics of the proposed GCN model.

Metric	Value
Accuracy	86.7%
F1-score	85.4%
Recall	83.2%
Precision	87.8%

4.3.2. Interpretation of the results

The results demonstrate that the GCN model effectively captures the structural and logical relationships required for theorem verification. The high precision indicates that the model rarely misclassifies a non-NSWG-CS as NSWG-CS, while the recall value confirms its ability to detect most valid NSWG-CS instances. The balanced F1-score reflects robustness in both detecting and avoiding false predictions.

4.3.3. Model generalization

The near-equal values of accuracy and recall suggest that the dataset has a balanced class distribution and that the model generalizes well across unseen graph structures. The use of dropout and weight decay contributed to regularization, minimizing overfitting despite the complex logical structure embedded in the data.

These evaluations confirm the viability of GCNs for automated theorem verification within nano topological spaces and encourage further integration of artificial intelligence in formal mathematical reasoning tasks.

As illustrated in Figure 4, the confusion matrix presents the classification outcomes of the GCN model, providing insight into the model's performance. The matrix shows 79 true positives, 61 true negatives, 10 false positives, and 16 false negatives for NSWG-CS identification.

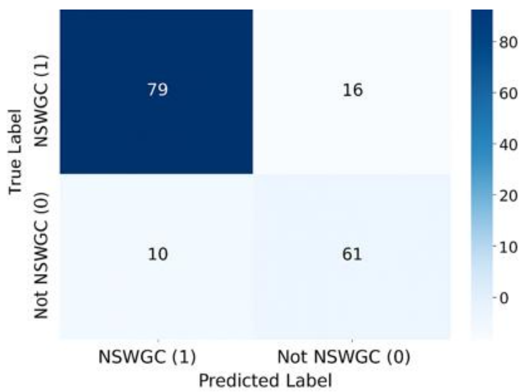

Figure 4. Classification results of the GCN model.

The structure of subset relationships in NTS can be effectively modeled using directed graphs, as demonstrated in Figure 5.

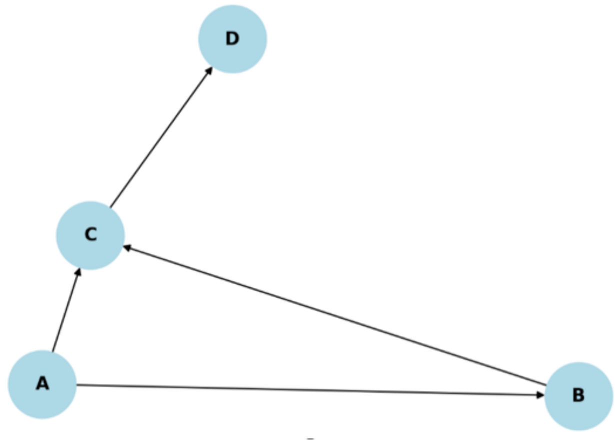

Figure 5. Example of a subset relationship graph in nano-topological space.

Here is the example subset relationship graph used as input for the GCN model. It shows the following inclusion hierarchy among sets:

- $A \subseteq B \subseteq C \subseteq D$.
- $A \subseteq C$ directly as well.

5. Results and discussion

The GCN-based model for NSWG-CS classification demonstrated consistent performance across multiple evaluation metrics. The outcomes were visualized and analyzed to provide a deeper understanding of the model's behavior, learning dynamics, and potential avenues for improvement.

5.1. Model output and visualization

Graph-based visualizations were used to illustrate the classification results. Each subset node was color-coded to reflect the model's predictions as follows:

Green: Correctly classified NSWG-CS (true positives);

Red: Misclassified instances (false positives or false negatives).

This interpretability enabled insights into which types of subset configurations were more prone to misclassification, often related to ambiguous closure and semi-open boundary definitions.

5.2. Learning curve and training stability

The training loss was monitored across 150 epochs. A steady decline in the cross-entropy loss confirmed successful learning convergence. Table 3 summarizes the loss progression, and Figure 6 plots the training curve.

Table 3. Loss curve over training epochs.

Epoch	Loss	Epoch	Loss
0	0.6503	80	0.4871
10	0.6391	90	0.4373
20	0.4993	100	0.4220
30	0.4807	110	0.2330
40	0.5109	120	0.3932
50	0.4364	130	0.5264
60	0.4580	140	0.3626
70	0.3349		

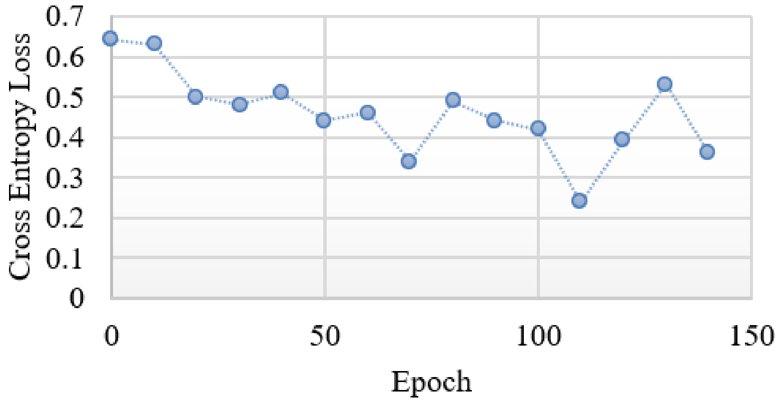

Figure 6. Training loss over epochs for the GCN model.

The model's learning progression is shown in Figure 7, where the cross-entropy loss decreases gradually across epochs, indicating the convergence of the GCN model.

The increase in model accuracy across training epochs, as shown in Figure 7, demonstrates the effective convergence of the GCN model.

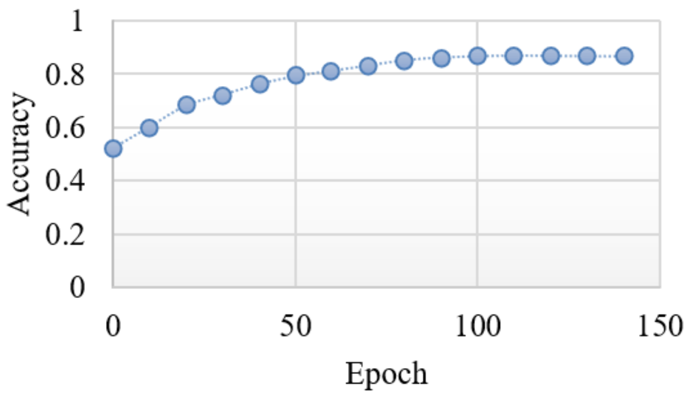

Figure 7. Accuracy over epochs for GCN model training.

5.3. Analysis of misclassifications

An error analysis revealed that most false predictions occurred in cases where subsets were close to the decision boundary, i.e., when the closure of the subset partially overlapped with non-semi-open supersets. These boundary cases highlight the complexity of representing approximate logical rules using neural embeddings.

5.4. Implications and applications

The results validate the application of GCNs to topological theorem verification, especially in domains where logical structures can be encoded as graph relations. Potential applications include automated verification systems for topological proof assistants, integration with symbolic reasoning engines, expansion to other types of generalized closed sets in fuzzy or intuitionistic topologies.

These findings suggest that deep learning can play a complementary role in mathematical logic, particularly for automated hypothesis evaluation and theorem exploration.

A comparative analysis of various machine learning models [5,6] is illustrated in Figure 8, where the GCN model clearly outperforms logistic regression, random forest, and support vector machines (SVM) [1] across all standard performance metrics. To conduct a meaningful comparison against the GCN model, conventional machine learning algorithms such as logistic regression, and random forest [7,8] were adapted to work with graph-based input data. As these models are typically designed to process flat, tabular formats, each graph was first converted into a fixed-length feature vector using a node embedding strategy.

We utilized the node2vec algorithm to extract both the local and global structural characteristics of the graph into dense vector representations. For every input instance, node embeddings were aggregated via averaging to generate a single feature vector representing the entire graph structure. These vectors served as the input for the baseline classifiers. The classification labels remained consistent across all models, based on whether the target subset fulfilled the NSWG-CS property. To ensure fairness, hyper parameters were optimized through a grid search and the model's performance was validated using five-fold cross-validation [11,13]. This approach allowed for a uniform evaluation framework, lending credibility and reproducibility to the performance comparison shown in Figure 8.

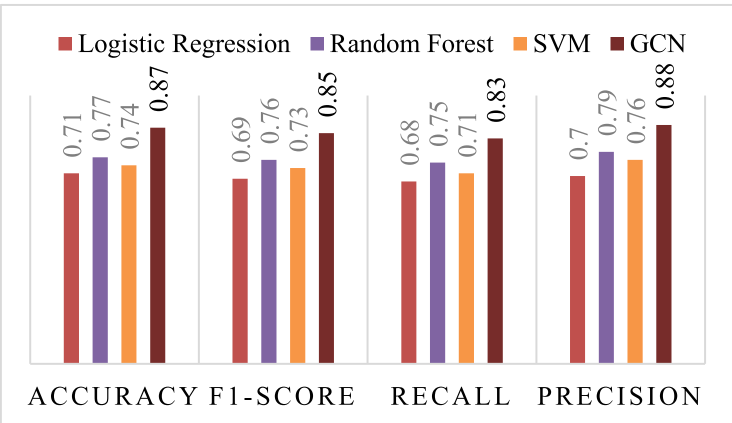

Figure 8. Performance comparison of ML models for NSWG-CS verification.

5.5. Benefits of machine learning in theorem verification

Machine learning introduces several advantages to theorem verification in topological studies

- Automated reasoning: Reduces the manual effort required for proving complex theorems.
- Error reduction: Ensures higher accuracy by minimizing human-induced errors.
- Scalability: Can handle larger datasets and diverse topological structures efficiently.
- Graph-based interpretability: Provides visual insights into the topological relationships, aiding in better understanding.
- Generalization: The trained model can be extended to verify other mathematical properties beyond NSWG-CS.

6. Conclusions and future work

6.1. Conclusions

This study introduced a new class of sets, namely NSWG-CS, within the framework of NTS and examined their relationships with other generalized closed sets. The theoretical analysis established the inclusion properties and boundary conditions under various forms of nano-open and nano-semi-open sets.

To automate theorem verification and reduce the reliance on manual logical reasoning, a machine learning-based framework [17] was proposed. Specifically, a GCN was designed to learn topological relationships from structured graph data. The synthetic dataset generation approach ensured diverse, balanced training samples, allowing the model to generalize effectively across unseen configurations. Achieving an accuracy of 86.7%, the model confirmed the feasibility of applying deep learning to formal mathematical logic.

The visualizations and training diagnostics demonstrated the model's robustness and interpretability. Although minor fluctuations were observed beyond 100 epochs, dropout and regularization mitigated overfitting. The approach also revealed specific challenges in accurately modeling boundary cases where the closure properties are ambiguous.

6.2. Future work

Future research could extend the current model to verify other forms of generalized closed sets such as α-closed, regular weakly closed, or fuzzy closed sets. Another research direction would be to incorporate symbolic reasoning tools alongside neural networks for hybrid logic ML verification. We could also explore the integration of attention-based graph models (e.g., GAT) for enhanced focus on critical node relationships, or validate the approach using real mathematical proof datasets or theorem libraries in proof verification.

This research demonstrates a meaningful step toward intelligent theorem verification and underscores the potential of artificial intelligence-augmented methods in computational topology and beyond.

Author contributions

S. Sathya Priya: Conceptualization, Methodology, Writing the original draft; N. Nagaveni: Supervision; R. Lavanya: Validation; R. Saveeth: Data curation, Software. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have used ChatGPT (OpenAI GPT-5) for language improvement and formatting assistance only. All intellectual content and analysis were developed by the authors.

Acknowledgments

The authors sincerely thank Coimbatore Institute of Technology for providing the necessary support and facilities to conduct this research.

Conflict of interest

The authors declare no conflicts of interest.

References

1. S. N. Bhagat, P. S. Rath, A. Mitra, Coupling of rough set theory and predictive power of SVM towards mining of missing data, *Int. Res. J. Multidiscip. Scope (IRJMS)*, **5** (2024), 732–744. https://doi.org/10.47857/irjms.2024.v05i02.0631
2. K. Bhuvaneswari, K. M. Gnanapriya, Nano generalized closed sets in nano topological space, *Int. J. Sci. Res. Publ.*, **4** (2014), 1–3.
3. N. Biswas, On characterization of semi-continuous functions, *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Nat.*, **48** (1970), 399–402.
4. Y. El Touati, W. Abdelfattah, Feature imputation using neutrosophic set theory in machine learning regression context, *Eng. Technol. Appl. Sci. Res.*, **14** (2024), 13688–13694. https://doi.org/10.48084/etasr.7052
5. S. P. Kristanto, R. S. Bahtiar, M. Sembiring, H. Himawan, L. Samboteng, Hariyadi, et al., Implementation of ML rough set in determining cases of timely graduation of students, *J. Phys. Conf. Ser.*, **1933** (2021), 012031. https://doi.org/10.1088/1742-6596/1933/1/012031
6. L. P. Khoo, L. Y. Zhai, A prototype genetic algorithm-enhanced rough set-based rule induction system, *Comput. Ind.*, **46** (2001), 95–106. https://doi.org/10.1016/S0166-3615(01)00117-8
7. J. Krupa, M. Minutti-Meza, Regression and machine learning methods to predict discrete outcomes in accounting research, *J. Financ. Report.*, **7** (2022), 131–178. https://doi.org/10.2308/JFR-2021-010
8. S. Kumar, V. Bhatnagar, A review of regression models in machine learning, *J. Intell. Syst. Comput.*, **3** (2022), 40–47.
9. M. Lellis Thivagar, C. Richard, On nano forms of weakly open sets, *Int. J. Math. Stat. Invent. (IJMSI)*, **1** (2013), 31–37.

10. N. Levine, Generalized closed sets in topology, *Rend. Circ. Mat. Palermo*, **19** (1970), 89–96. https://doi.org/10.1007/BF02843888
11. R. Z. Li, J. M. Herreros, A. Tsolakis, W. Z. Yang, Machine learning regression based group contribution method for cetane and octane numbers prediction of pure fuel compounds and mixtures, *Fuel*, **280** (2020), 118589. https://doi.org/10.1016/j.fuel.2020.118589
12. P. E. Long, L. L. Herington, Basic properties of regular-closed functions, *Rend. Circ. Mat. Palermo*, **27** (1978), 20–28. https://doi.org/10.1007/BF02843863
13. P. Mahajan, R. Kandwal, R. Vijay, Rough set approach in machine learning: a review, *Int. J. Comput. Appl.*, **56** (2012), 1–13. https://doi.org/10.5120/8924-2996
14. A. S. Mashhour, M. E. Abd El-Monsef, S. N. El-Deeb, On pre-continuous and weak pre-continuous mapping, *Proc. Math. Phys. Soc. Egypt*, **53** (1982), 47–53.
15. N. Nagaveni, M. Bhuvaneswari, On nano weakly generalized closed sets, *Int. J. Pure Appl. Math.*, **106** (2016), 129–137.
16. O. Njastad, On some classes of nearly open sets, *Pac. J. Math.*, **15** (1965), 961–970. http://dx.doi.org/10.2140/pjm.1965.15.961
17. I. H. Sarker, Machine learning: algorithms, real-world applications, and research directions, *SN Comput. Sci.*, **2** (2021), 160. https://doi.org/10.1007/s42979-021-00592-x
18. P. Sundaram, N. Nagaveni, Weakly generalized closed sets in topological space, *Proc. 84th Indian Sci. Congress*, Delhi, India, 1997.