International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

Extraction of Urban Buildings from LiDAR Point Clouds Using DBSCAN Clustering

Jurgis Zagorskas[1], Jūratė Vinogradova[2]
[1,2]Department of Engineering Graphics, Vilnius Gediminas Technical University, Vilnius, Lithuania
Email address: jurgis.zagorskas@vilniustech.lt

Abstract— *Airborne Light Detection and Ranging (LiDAR) datasets are widely used for urban analysis and three-dimensional modelling due to their high spatial accuracy and ability to represent complex forms. In many LiDAR datasets, points are already classified into thematic classes such as ground, vegetation, and buildings. However, even when building points are provided as a separate class, individual buildings remain merged within a single point cloud and require additional processing to be separated into distinct objects. This study presents a simple method for separating individual buildings from a building-class LiDAR point cloud. It applies the density-based clustering algorithm DBSCAN (Density-Based Spatial Clustering of Applications with Noise) to automatically group spatially connected points into clusters representing individual buildings. The method does not require the predefined number of buildings and is robust to noise and variations in building geometry. The resulting clusters represent separate building structures that can be exported as independent datasets for further processing. Such segmented building point clouds can serve as an initial step for applications including three-dimensional city model generation, urban analysis, and integration with geographic information systems.*

Keywords— *LiDAR; point cloud clustering; DBSCAN; building segmentation; urban modelling; 3D city models*.

I. Introduction

LiDAR sensors produce dense point clouds that accurately represent the geometry of terrain, vegetation, and built structures [1]. Due to their high spatial resolution and geometric accuracy, LiDAR datasets are widely used in applications such as urban modelling, infrastructure management, environmental analysis, and GIS [2].

One of the tasks in LiDAR data processing is the identification and extraction of buildings from point cloud datasets [3]. Accurate building detection is essential for practical applications in urban planning, disaster management, and 3D modelling [4,5]. In recent years, numerous approaches have been proposed for building extraction from LiDAR data, including rule-based algorithms, clustering methods, and machine learning techniques [6,7]. These methods typically involve several preprocessing steps such as separating ground and non-ground points [8], feature extraction, and classification of point cloud elements into thematic classes such as terrain, vegetation, and buildings .

In many modern LiDAR datasets, classification has already been performed during preprocessing, and building points are provided as a separate thematic class. While this classification significantly simplifies the identification of built structures, the building class usually still contains points belonging to many individual buildings within a single dataset. For further analysis and modelling tasks, it is often necessary to separate these points into clusters corresponding to individual building objects. Such object-level separation is an important step in workflows related to urban data analysis, building inventory creation, and the generation of three-dimensional city models [9].

Density-based clustering algorithms provide an effective approach for identifying spatially connected groups of points in large datasets. One of the most widely used algorithms is DBSCAN (Density-Based Spatial Clustering of Applications with Noise), which groups points based on spatial density and does not require the predefined number of clusters [10]. The algorithm is particularly suitable for spatial data because it can detect clusters of arbitrary shapes and is robust to noise and outliers [11]. These characteristics make DBSCAN well suited for analyzing LiDAR point clouds, when buildings can be clustered by spatial proximity between points.

The aim of this research is to demonstrate a simple and efficient method for separating individual buildings from a LiDAR point cloud dataset that already contains only building-class points. The proposed approach applies the DBSCAN clustering algorithm to group spatially connected points into clusters representing individual buildings. The resulting clusters can then be exported as separate datasets and used for further applications such as geometric analysis, building reconstruction, or the generation of three-dimensional urban models.

II. Building Extraction and Clustering Methods in LiDAR Point Clouds

Processing of LiDAR point clouds for urban environments has been widely studied, particularly in the context of building detection and urban feature extraction [12]. Numerous approaches have been proposed to identify buildings within point clouds. Early approaches focused on rule-based methods that used height thresholds, planar analysis, and color filters to distinguish buildings from terrain and vegetation [13-15]. More recent studies have incorporated machine learning and deep learning techniques to improve classification accuracy in point cloud datasets based on geometry, intensity, or context [16,17]. Although these methods have demonstrated promising results, their application often requires significant computational resources [18].

Many operational LiDAR datasets are already provided with preliminary classification performed during the data preprocessing stage. In LiDAR datasets, building-class points represent multiple buildings aggregated within a single class

Jurgis Zagorskas and Jūratė Vinogradova, "Extraction of Urban Buildings from LiDAR Point Clouds Using DBSCAN Clustering," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 10, pp. 11-15, 2026.

rather than distinct objects. The remaining task is separating individual buildings within this dataset. Object-level separation is necessary to enable further analysis, such as building inventory generation, urban modelling, or spatial statistics.

Clustering techniques are frequently applied to identify spatially coherent structures in point cloud data. Among these methods, density-based clustering algorithms have proven particularly suitable for spatial datasets due to their ability to detect clusters of arbitrary shape and to identify noise points [19,20]. The DBSCAN algorithm is one of the most widely used density-based clustering techniques for spatial analysis. Previous research has applied DBSCAN and similar clustering algorithms in various point cloud processing tasks, including object detection, terrain analysis, and infrastructure identification [21-23]. It groups points based on local density criteria defined by neighborhood radius and minimum point count parameters, allowing it to identify dense regions corresponding to individual objects [11]. The method performs well when processing large datasets with irregular geometries, which are common in urban LiDAR point clouds.

Relatively little attention has been given to the specific task of separating building points into individual building objects. Performing this step correctly can significantly improve the usability of LiDAR datasets for subsequent processing tasks, such as 3D urban modelling, spatial analysis, and automated urban inventory generation.

A. Challenges in Building Extraction

This study applies DBSCAN clustering to selected building-class LiDAR points to separate individual buildings based on spatial proximity.

Although it works with the main types of building morphology, in dense urban areas, perimetric buildings, row houses, industrial territories, where buildings often share walls and form continuous street blocks, spatial clustering alone may not fully distinguish individual structures. In such cases, clusters can be further divided using cadastral land parcel boundaries to ensure the separation of buildings belonging to different plots. This additional refinement step is also incorporated in the methodology presented in this study.

Another challenge arises from the vertical distribution of LiDAR points. Airborne LiDAR data are typically collected from a near-vertical perspective, which often results in sparse point coverage beneath roof overhangs. As a consequence, vertical gaps may occur between roof and lower building elements, and these distances can exceed the neighborhood threshold used by the DBSCAN algorithm. To address this issue, a preprocessing step is applied in which the Z coordinate is removed and clustering is performed only in the horizontal (X–Y) plane. This simplification reduces computational complexity and prevents incorrect separation caused by vertical gaps in the point cloud (see Figure 2).

During the clustering stage, particular attention was paid to the choice of dimensionality and DBSCAN parameters. Initial experiments were performed using the full three-dimensional coordinates of the point cloud (X, Y, Z). However, it was observed that clustering results were strongly influenced by vertical variations in the data. In urban LiDAR datasets, buildings often contain significant height differences caused by roof geometry, roof equipment, chimneys, dormers, or multi-level roof structures. When the Z coordinate is included in the distance computation, these vertical differences may artificially separate points belonging to the same building into multiple clusters. Consequently, clustering based purely on three-dimensional Euclidean distance can lead to fragmentation of building structures.

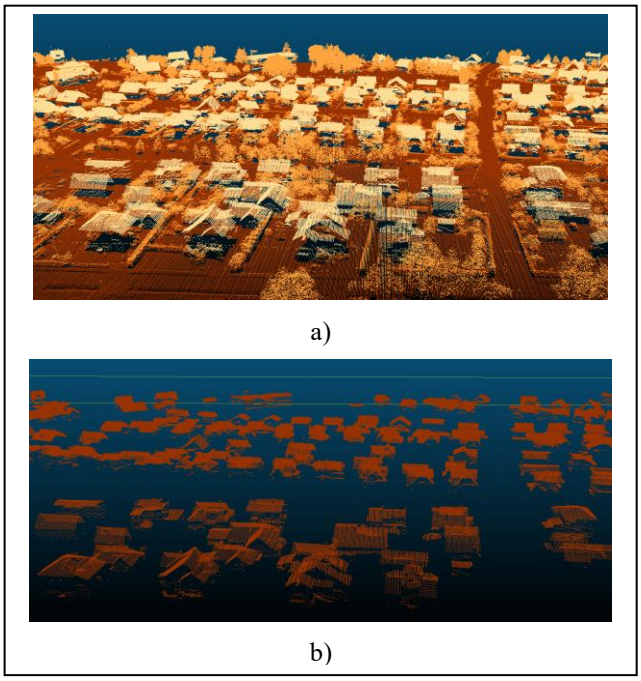

Fig. 1. a) Classified LIDAR points – buildings, vegetation and groud – marked in different colors. b) Extracted building points.

The DBSCAN parameters were determined empirically based on the typical point density and spatial characteristics of the dataset. The neighborhood radius parameter was set to ε = 1.5 m, which defines the maximum distance between neighboring points to be considered part of the same cluster. This value was chosen to be sufficiently large to maintain connectivity across roof surfaces that may contain small gaps in point distribution, while still preventing nearby buildings from merging into a single cluster. The minimum number of points parameter (min_points = 10) was selected to ensure that only spatially meaningful point groups form clusters, while small isolated groups of points are classified as noise. This helps reduce the influence of sparse measurement artifacts or minor objects located on rooftops.

The combination of horizontal clustering and empirically tuned DBSCAN parameters produced stable segmentation results for building-scale structures. While smaller ε values tend to fragment individual buildings into multiple clusters, excessively large ε values may merge neighboring structures. The selected parameters therefore represent a compromise that balances cluster continuity within individual buildings while preserving separation between distinct structures.

Jurgis Zagorskas and Jūratė Vinogradova, “Extraction of Urban Buildings from LiDAR Point Clouds Using DBSCAN Clustering,” *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 10, pp. 11-15, 2026.

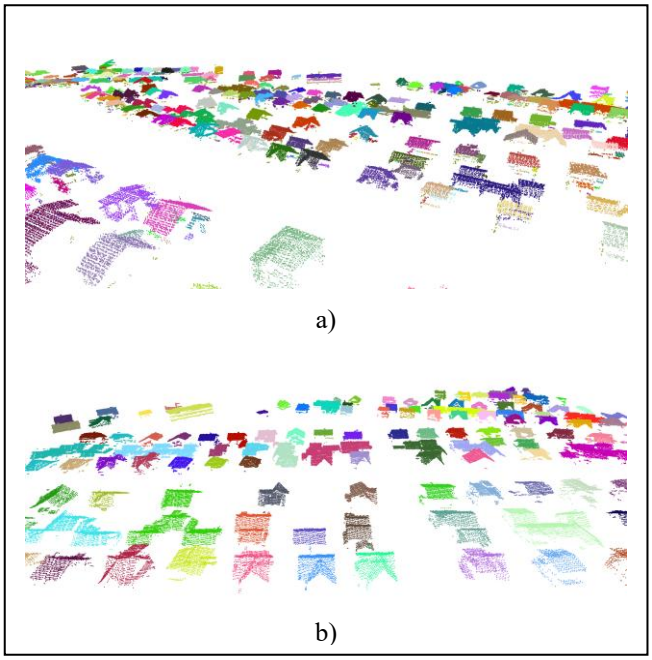

Fig. 2. *DBSCAN Separated Buildings: a) With z coordinates, b) Only x.y coordinates – reduces the errors caused by vertical distances.*

III. Methodolygy for Building Point Cloud Extraction and Clustering

In this study, Python-based open-source libraries were used to process and cluster LiDAR point cloud data. The workflow was implemented using:

- laspy – to read and process LAS/LAZ point cloud files;
- NumPy – to efficiently handle and manipulate large numerical datasets;
- Open3D – to perform point cloud processing and density-based clustering (DBSCAN). Open3D was used under Python 3.10 due to compatibility requirements.

Point cloud data stored in LAS/LAZ format typically contains classification codes that describe object types:

Class 2 – Ground
Class 5 – Vegetation / Trees / Power lines
Class 6 – Buildings

To isolate building structures, points were filtered using the classification code corresponding to buildings (usually class 6).

Example filtering procedure:

```
pmask = las.classification == 6
# Extract XYZ coordinates of building points
points_buildings = np.vstack((
    las.x[pmask], las.y[pmask], las.z[pmask])).T
```

After extracting building points, spatial clustering was performed using the built-in DBSCAN implementation in Open3D.

The building points were converted into an Open3D point cloud object:

```
pcd2 = o3d.geometry.PointCloud()
pcd2.points= o3d.utility.Vector3dVector(points_buildings)
```

Density-based clustering was then applied:

```
labels2 = np.array(
    pcd2.cluster_dbscan(
        eps=1.5,
# Neighborhood radius in meters
        min_points=10,
# Minimum number of points
        print_progress=True
    )
)
```

DBSCAN groups spatially connected points into clusters based on:

- A maximum neighbor distance (eps);
- A minimum number of points required to form a dense region (min_points).

This approach enables automatic segmentation of individual building structures from the filtered dataset.

This step produces a 3D subset containing only building-related points.

To visually distinguish detected clusters, each cluster was assigned a random color:

```
import random

colors = np.zeros((len(labels2), 3))

cluster_colors = {
    label: [random.random(), random.random(), random.random()]
    for label in np.unique(labels2)
}

# Assign colors to points
for label in np.unique(labels2):
    colors[labels2 == label] = cluster_colors[label]

pcd2.colors = o3d.utility.Vector3dVector(colors)

# Visualization
o3d.visualization.draw_geometries(
    [pcd2],
    window_name="DBSCAN Clusterization Results")
```

The visualization (see Figure 2a) illustrates spatially separated building clusters using different colors.

Refining Building Clusters Using Cadastral Parcel Boundaries

A further example, set in a denser urban environment where buildings form continuous blocks, demonstrates distinct clustering outcomes compared to sparser areas. This dataset also features a higher LiDAR point density (~5 cm), providing fine geometric detail for individual structures. Using the previously described DBSCAN-based clustering methodology,

Jurgis Zagorskas and Jūratė Vinogradova, "Extraction of Urban Buildings from LiDAR Point Clouds Using DBSCAN Clustering," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 10, pp. 11-15, 2026.

individual buildings can initially be delineated as separate urban blocks (Figure 3b).

To further refine the segmentation of buildings, cadastral land-parcel boundaries were incorporated into the processing workflow. The parcel polygons were imported from a GIS shapefile and spatially joined with the LiDAR point cloud using a spatial indexing approach. Each point classified as a building was assigned to the corresponding land parcel based on its horizontal position (X,Y). The resulting parcel identifiers were then combined with the DBSCAN cluster labels, enabling the subdivision of clusters that extended across parcel boundaries. This step allowed large building clusters to be partitioned into separate building entities corresponding to individual cadastral plots while preserving the spatial continuity of the LiDAR data (see Figure 3 c).

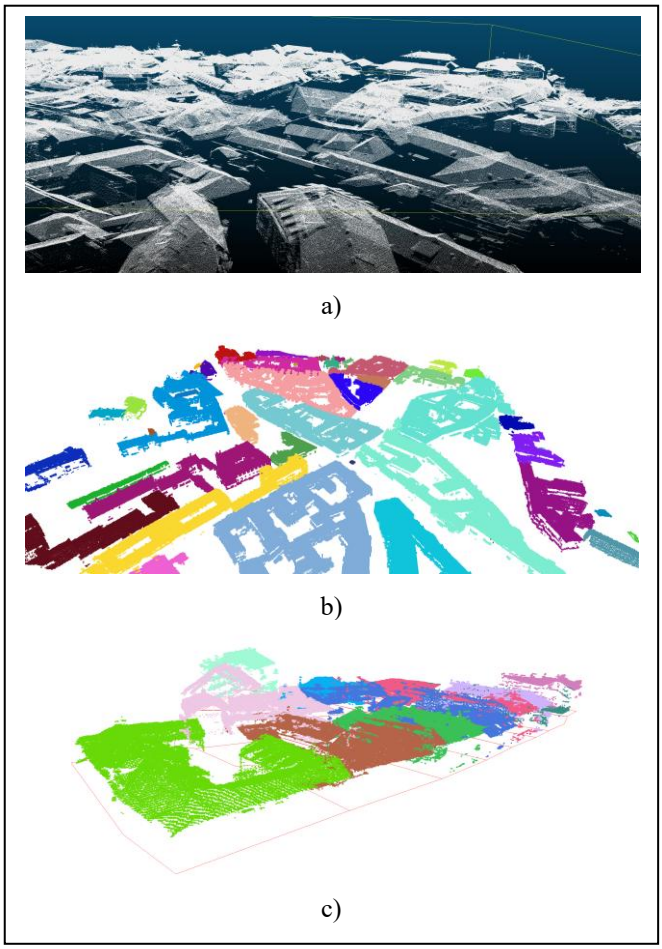

Fig. 3. *Classification results in dense urban environment where building for continuos town blocks (a-point cloud with 5cm resolution, b-separated building blocks, c-building blocks separated by land parcells)*

To implement this refinement computationally, additional Python libraries are required: GeoPandas for handling polygonal GIS data and Shapely for geometric operations. These tools enable the efficient manipulation and analysis of cadastral parcel polygons alongside the LiDAR point cloud:

```
import geopandas as gpd
import pandas as pd
from shapely.geometry import Point
import numpy as np

# Convert point cloud to a GeoDataFrame

pcd_points = np.asarray(pcd.points)

points_gdf = gpd.GeoDataFrame(
    {'cluster_label': labels},
    geometry=[Point(x, y) for x, y, z in
pcd_points],
    crs=parcels.crs
)

# Spatial join points with parcels
points_with_parcels =
gpd.sjoin(points_gdf, parcels,
 how="left", predicate='within')
```

In this refined view, large building clusters that previously spanned multiple parcels are now segmented into discrete structures aligned with cadastral boundaries (Figure 3c). This step ensures that the resulting building models respect administrative divisions while preserving the spatial integrity of the LiDAR data.

It is important to note that while this approach works effectively for moderate-sized datasets, processing very dense urban point clouds with millions of points can become computationally intensive. In such cases, optimized spatial indexing (e.g., using rtree), parallel processing, or downsampling strategies can significantly accelerate the assignment of points to parcels:

```
sindex = parcels.sindex

for point in pcd_points_gdf.geometry:

   possible_matches_index =
list(sindex.intersection(point.bounds))

   possible_matches =
 parcels.iloc[possible_matches_index]

   precise_match =
possible_matches[possible_matches.contain
s(point)]
```

By integrating cadastral parcel boundaries with DBSCAN clustering, the method provides a robust, reproducible workflow for urban building delineation from LiDAR point clouds. The combination of 3D cluster information and 2D parcel geometry allows urban planners and GIS professionals to produce high-fidelity building representations, enabling further analyses such as volumetric computations, solar potential assessments, and 3D visualization for planning applications.

IV. Conclusions

This study presents a practical and efficient methodology for extracting individual buildings from LiDAR point clouds using the DBSCAN density-based clustering algorithm. By

Jurgis Zagorskas and Jūratė Vinogradova, “Extraction of Urban Buildings from LiDAR Point Clouds Using DBSCAN Clustering,” *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 8, Issue 10, pp. 11-15, 2026.

filtering building-class points from airborne LiDAR datasets and applying DBSCAN clustering in the horizontal plane (X–Y coordinates), individual buildings can be automatically separated into distinct clusters. This approach reduces errors caused by vertical gaps, roof overhangs, and multi-level roof structures, which are common in urban LiDAR data. The method is robust to noise, variations in point density, and differences in building geometry, making it suitable for datasets that range from sparse coverage to high-resolution point clouds with densities as fine as 5 cm between points.

Clustering parameters, such as the neighborhood radius (ε) and minimum points per cluster, were empirically tuned to balance the connectivity of points within a building while preventing the merging of adjacent structures. Horizontal clustering ensures that fragmented points due to height variations do not create multiple clusters for the same building, while sparse outlier points are automatically classified as noise. In dense urban environments, where buildings often form continuous street blocks or share walls, integrating cadastral parcel boundaries allows further refinement, separating clusters that extend across multiple plots while preserving spatial integrity.

The workflow, implemented using open-source Python libraries including laspy for point cloud handling, Open3D for clustering, and GeoPandas/Shapely for spatial operations, enables automated and reproducible processing of large LiDAR datasets. Typical urban point clouds processed in this study contained millions of points per dataset, demonstrating that the method scales to realistic, city-scale scenarios. Performance can be further optimized through spatial indexing, parallel processing, or downsampling for extremely dense datasets.

The resulting building clusters can be exported as independent datasets for further analysis, including three-dimensional city model generation, volumetric calculations, solar potential assessment, and integration with geographic information systems. Overall, this methodology provides a reliable, flexible, and computationally feasible framework for urban building extraction from LiDAR point clouds, bridging the gap between raw classified point clouds and object-level building data required for urban analysis, planning, and visualization.This study presents a streamlined method for recognizing and analyzing urban trees from freely available satellite LIDAR data, offering a practical solution to the challenges of traditional tree inventory methods. By extracting essential tree attributes such as height, diameter, and crown structure, and converting them into compact GIS records, the proposed method bridges the gap between advanced remote sensing technologies and urban planning applications.

REFERENCES

[1] Irwin, L.A.; Coops, N.C.; Anders, K.; Mandlburger, G.; Winiwarter, L. Light detection and ranging of natural systems. Nature Reviews Methods Primers 2025, 5, 76.
[2] Dritsas, E.; Trigka, M. Remote sensing and geospatial analysis in the big data era: A survey. Remote Sensing 2025, 17, 550.
[3] Pan, S.; Zhang, R.; Liu, Y.; Gong, M.; Huang, H. Building LOD representation for 3D urban scenes. ISPRS Journal of Photogrammetry and Remote Sensing 2025, 226, 16-32.
[4] Patel, J.; Sharma, N.; Mohan, S. Introduction to remote sensing and GIS. In Smart buildings and cities with remote sensing and GIS; Chapman and Hall/CRC: 2025; pp. 3-34.
[5] Thakur, U.P.; Wadaskar, S.; Narnaware, E.; Dhawangle, S.; Karluke, S.; Dhenge, A. LIDAR Point Cloud usage for Mapping Vegetation and Buildings. Grenze International Journal of Engineering & Technology (GIJET) 2025, 11.
[6] Yadav, K.; Alkwai, L.M.; Almansour, S.; Siddiqui, M.A.; Sharma, D.K.; Garg, L.; Goswami, P.; Alkhayyat, A.H. An Enhanced Rule-Based Fuzzy Segmentation Approach for Automated Urban Feature Extraction Using HighResolution Satellite Imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2025.
[7] Zhang, Z.; Xu, Z.; Cao, Y.; Xu, N.; Wang, S.; Cui, S.a.; Li, Z.; Qin, R. Deep learning for 3D point cloud processing--from approaches, tasks to its implications on urban and environmental applications. arXiv preprint arXiv:2509.12452 2025.
[8] Santo, A.; Heredia, E.; Viegas, C.; Valiente, D.; Gil, A. Ground segmentation for lidar point clouds in structured and unstructured environments using a hybrid neural–geometric approach. Technologies 2025, 13, 162.
[9] Kim, M.; Boo, J.; Yoon, K.-J. Generalized Category Discovery for LiDAR Semantic Segmentation. In Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2026; pp. 8416-8426.
[10] Poleselo, H.N.; Conceicao, A.G. Comparative Analysis of Clusterization Methods Applied to a Solid-State LiDAR. In Proceedings of the 2025 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2025; pp. 3-8.
[11] Gong, Y.; Ma, Y.; Cheng, S. 3D modeling of urban buildings using improved DBSCAN algorithm and grid partitioning. International Journal of Architectural Computing 2025, 14780771251405444.
[12] Li, Y.; Xiong, B.; Wang, C.; Sun, T.; Xiong, Q.; Kong, Q. Automatic, batching and remote detection of urban building inclination information with edge computing of LIDAR point clouds on a UAV. Journal of Building Engineering 2025, 109, 112981.
[13] Jifroudi, H.M.; Mansor, S.B.; Pradhan, B.; Halin, A.A.; Ahmad, N.; Abdullah, A.F.B. A new approach to derive buildings footprint from light detection and ranging data using rule-based learning techniques and decision tree. Measurement 2022, 192, 110781.
[14] Siddiqui, F.U.; Teng, S.W.; Awrangjeb, M.; Lu, G. A robust gradient based method for building extraction from LiDAR and photogrammetric imagery. Sensors 2016, 16, 1110.
[15] Hamedianfar, A.; Shafri, H.Z.M.; Mansor, S.; Ahmad, N. Improving detailed rule-based feature extraction of urban areas from WorldView-2 image and lidar data. International Journal of Remote Sensing 2014, 35, 1876-1899.
[16] Li, Y.; Xiao, X. Deep learning-based fusion of optical, radar, and LiDAR data for advancing land monitoring. Sensors 2025, 25, 4991.
[17] Maskeliūnas, R.; Maqsood, S.; Vaškevičius, M.; Gelšvartas, J. Hybrid deep learning and geometric algorithms for individual object detection in urban LiDAR point clouds. International Journal of Remote Sensing 2025, 46, 9118-9156.
[18] Vijaywargiya, J.; Ramiya, A.M. From Data to Planning: Innovations Through LiDAR Technology and Data Science in Sustainable Urban Resource Management. In Artificial Intelligence and Machine Learning for Climate Disaster Management; Springer: 2025; pp. 199-217.
[19] Kim, Y.; Yun, S. Performance Analysis of Point Cloud Edge Detection for Architectural Component Recognition. Applied Sciences 2025, 15, 9593.
[20] He, H. A clustering algorithm based on grids for core data and adjacency relationships for edge data. Scientific Reports 2025, 15, 18390.
[21] Łabuz, R.; Blazy, R.; Hess, D.B. Identifying urban development clusters in southern Poland using the DBSCAN algorithm for spatial analysis. Town Planning Review 2025, 1-33.
[22] Liu, J. Density-based Clustering Algorithm for the Protection and Planning of Ancient Villages. In Proceedings of the 2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 2025; pp. 1-8.
[23] Ma, J.; Luo, B.; Zhao, Y.; Li, S.; Chen, M.; Wei, J.; Wang, S.; Zhang, X. Detection and automatic identification of landslide areas from the LiDAR point clouds using improved DBSCAN. Landslides 2025, 22, 3843-3854.