УДК 681.518.2 DOI: 10.18372/2073-4751.83.20511

Калашник М.О.,
e-mail: 294776@stud.kai.edu.ua

MACHINE LEARNING-BASED ANOMALY DETECTION WITH ISOLATION FOREST IN LARGE-SCALE DATA ANALYSIS

State University "Kyiv Aviation Institute"
e-mail: 294776@stud.kai.edu.ua

Introduction

The task of anomaly detection involves finding rare items or events that are inconsistent with the vast majority of the data. Practically, this is applied to identify everything from financial fraud and new forms of cyber attacks to faulty machinery. While these applications seem distinct, they all share the common challenge of operating without target variables.

Anomalies can be broadly categorized into three distinct types. The most basic form is the point anomaly (also termed global anomaly), which is an individual data point that is statistically deviant when compared to the entire dataset (e.g., a single, exceptionally large financial transaction). In contrast, contextual anomalies (or conditional anomalies) are instances that are unusual only within a specific environment or condition, a frequent occurrence in time-series data; a sudden temperature spike in winter is a classic example. Finally, collective anomalies represent a more complex case where a group of related data points behaves abnormally as a unit, even if the individual points seem normal. These collective deviations, which disturb the overall data distribution, are common in dynamic systems like network traffic and usually demand sophisticated pattern-detection algorithms.

A primary challenge in outlier detection is the lack of a clear, predefined definition for what constitutes an anomaly. This ambiguity is often compounded by the need to search for these outliers within large, high-dimensional datasets. The Isolation Forest algorithm has gained significant popularity as it is well-suited to these challenges.

Isolation Forest is an unsupervised, tree-based method for identifying anomalies. In the same way a Random Forest is an ensemble of Decision Trees, an Isolation Forest is a collection of Isolation Trees. However, these trees are constructed differently; rather than using a metric like Gini impurity, they are built using a process of random feature selection and random splitting [1].

To grasp how anomaly scores are calculated, it is crucial to first understand this fundamental building block. For simplicity, a dataset of 1000 transactions (instances) will be analyzed, considering only two variables: the transaction amount (x_1) and the time of day (x_2).

To create an Isolation Tree, the process starts with all or a sample of instances in the root node. Fig. 1 below shows a sample of 256 instances (more on that number later).

The underlying principle is that an anomalous instance, being "different," is easier to separate from the rest of the data. Therefore, during the random partitioning process, it will require fewer splits to be isolated, resulting in a shorter path length within the tree. An Isolation Forest leverages an ensemble of these trees to average out the variability of any single random tree.This ensemble of trees is used to calculate a final anomaly score, *s(x,n)*, for each instance *x*. The score is derived from *E[h(x)]*, which represents the average path length of instance *x* across all Isolation Trees (where *h(x)* is its path length in one tree). The calculation also includes *c(n)*, a normalization factor based on the sample size *n*. This normalization is critical because the average path lengths for all instances naturally grow longer as the sample size (and thus the tree depth) increases. The anomaly score can be calculated using the following formula:

$$s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}}$$

The anomaly score s is normalized to a value between 0 and 1. This score can be interpreted based on three primary outcomes:

High Anomaly Probability ($s \to 1$): This occurs when the instance's average path length *E[h(x)]* is significantly shorter than the average path length *c(n)*.

Low Anomaly Probability ($s \to 0$): This occurs when the instance's average path length *E[h(x)]* is longer than *c(n)*.

Indeterminate ($s = 0.5$): This result is produced when the instance's average path length *E[h(x)]* is approximately equal to the average path length *c(n)* [2].

Put simply, a score nearing 1 strongly suggests an anomaly because it indicates the instance was isolated much more easily (with a shorter path) than expected for the dataset.

Fig. 2 below gives some intuition for why this process isolates outliers. Instances A, B, and C seem different from the other transactions.

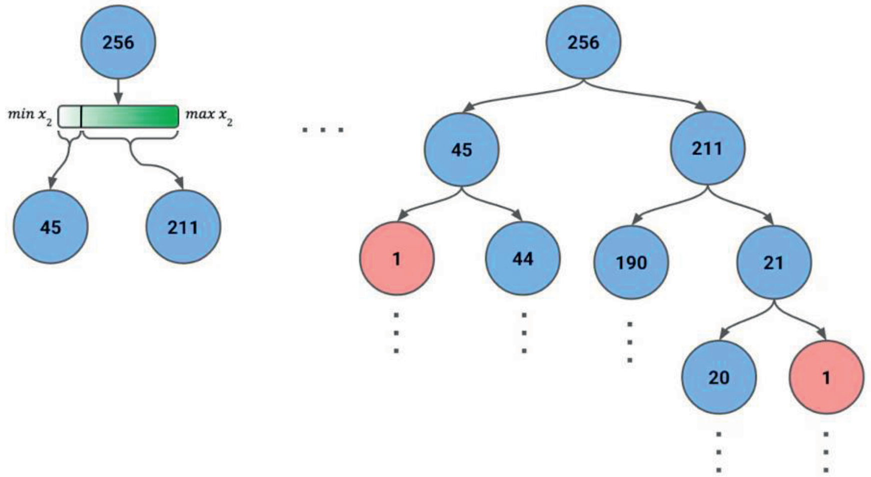

Fig. 1. The process for creating an Isolation Tree. Instances are isolated in leaf nodes by recursively splitting instances using a random feature and value in that feature's range

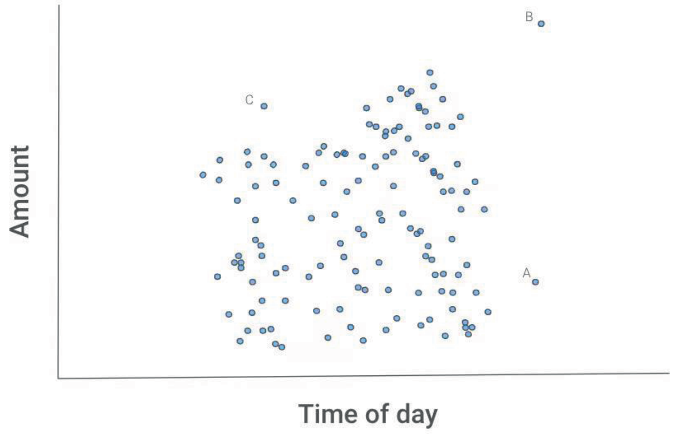

Fig. 2. Scatter plot of transactions

Based on Fig. 2, B will likely be isolated first as it can be separated from the other transactions using one split from either feature. A would take one split from x2. C may take longer as it would need to isolate it using both features. That is, C has both a normal amount and time but not a normal amount for that time of day.

The important point is that, on average, B will take fewer splits than A, A fewer splits than C, and all three points will take fewer splits than the other transactions.

Literature review and problem statement

Recent research on anomaly detection concentrates on methods that scale to large, heterogeneous data without labels, remain stable under drift, and provide controls that practitioners can tune. A central line of work keeps developing Isolation Forest by addressing structural artifacts, extending it to streaming settings, and coupling it with representation learning.

Lesouple et al. analyze structural issues that appear in extended tree variants and introduce Generalized Isolation Forest. Their design removes empty branches and mitigates partition artifacts, which leads to more stable scoring while preserving the efficiency that makes isolation-based methods attractive for high-volume workloads. Results reported in their study indicate competitive accuracy with stronger control over how trees are constructed and how scores are normalized, clarifying where implementation details matter in practice [3].

For continuous data streams, Togbe et al. study anomaly detection with a stream-ready variant implemented in scikit-multiflow. They compare Isolation Forest to Half-Space Trees on real streams and examine resource trade-offs such as memory footprint and update latency, which determine feasibility in production pipelines that operate under sliding windows and prequential evaluation. Their evidence positions isolation-based detectors as practical building blocks in streaming toolkits. An associated open-source implementation further illustrates how to deploy these ideas in Python ecosystems [4].

Problem statement. The goal here is to determine whether a straightforward, reproducible configuration of Isolation Forest can deliver stable and actionable anomaly signals in batch, large-scale measurements without labels. Concretely, this study focuses on three tasks that mirror the actual workflow in this project: align contamination-based thresholds with a manageable review volume and a zero-referenced decision score; verify that results remain consistent under routine data cleaning and reasonable parameter settings; and provide lightweight visual explanations that relate flagged observations to the underlying features. This keeps the literature insights grounded in the exact conditions of the simulations and the type of outputs intended for analysts to use.

The purpose and objectives of the study

The purpose of this study is to build and validate a simple, reproducible Isolation Forest workflow that produces reliable anomaly flags in batch, large-scale multivariate data without labels and that can be operated with clear, practical controls. The work implements a concise Python pipeline in which data are cleaned, a small set of continuous indicators is selected, the model is trained with `n_estimators` = 100, `max_samples` = 256, and `contamination` = 0.01, and the outputs are inspected through two diagnostics: decision scores over time with a zero-referenced threshold and a two-feature scatter that contrasts flagged and normal points. The objectives are to confirm that the contamination setting yields a manageable alert volume, to check that flagged observations form coherent patterns rather than isolated noise, to verify that results remain stable under modest changes to parameters and random seed, and to provide lightweight visual explanations that help an analyst understand why specific observations were flagged. Success is defined as achieving stable anomaly volumes at the chosen operating point, obtaining interpretable patterns in both diagnostics, reproducing the results from clean code, and meeting typical runtime and memory constraints for repeated batch runs.

Results of the research

The dataset of air quality measurements from a sensor in Kyiv was used in the research [6]. In the context of this dataset, an anomaly can be considered a sensor reading that indicates unusually high levels of pollution.

These parameters were used to train the Isolation Forest. Here are three values below:

n_estimators is the number of Isolation Trees used in the ensemble. A value of 100 is used in the Isolation Forest paper. Through experimentation, the researchers found this to produce good results over a variety of datasets.

contamination is the percentage of data points that expected to be anomalies.

sample_size is the number of instances used to train each Isolation Tree. A value of 256 is commonly used as it allows us to avoid using a maximum tree size stopping criteria. This is because it can be expected reasonable maximum tree sizes of log(256) = 8.

Unlike other parameters, the contamination value often lacks a rigid statistical justification and is typically set based on estimation. Its origin can be domain knowledge from prior analyses; for instance, if previous studies identified that 1% of readings signaled high pollution levels. Alternatively, the value might be dictated by resource constraints. Visualizing the anomaly scores will later demonstrate how this contamination parameter adjusts the final results.

```
# Parameters
n_estimators = 100  # Number of
trees
contamination = 0.01  # Expected
proportion of anomalies
sample_size = 256  # Number of
samples used to train each tree
```

Here is the code of how the training of the Isolation Forest was set up.

```
# Train Isolation Forest
iso_forest =
IsolationForest(n_estimators=n_estim
ators,
contamination=contamination,
max_samples=sample_size,
random_state=42)
iso_forest.fit(features)
```

The model provides two distinct outputs. The decision_function method computes the raw anomaly score for each instance, consistent with the theoretical framework previously discussed. In contrast, the predict method returns a binary classification (e.g., -1 or 1) that is determined by the contamination parameter. In our case, this means the 1% of instances with the scores indicating the highest anomaly likelihood are assigned a value of -1, while all other instances receive a value of 1.

The trained model has two useful functions:

decision_function will calculate the anomaly score in a similar way to what it was discussed previosuly.

predict will provide a binary label based on the contamination values. In our case, the 1% of instances with the worst anomaly scores will be given a value of -1. The other instances are given a value of 1.

The scatter plot highlights 80 instances in red, which correspond to the data points with the lowest anomaly scores. These are flagged as potential outliers that warrant further investigation. Here is the code for anomaly scores calculation:

```
# Calculate anomaly scores and
classify anomalies
data=data.loc[features.index].copy()
data['anomaly_score']=
iso_forest.decision_function(feature
s)
data['anomaly']=iso_forest.predict(f
eatures)
data['anomaly'].value_counts()
```

The scores shown in the plot are adjusted values, not the raw scores. The adjustment is based on the contamination parameter. First, an offset is calculated, which corresponds to the anomaly score percentile defined by the contamination value (in this case, the 0.99 percentile). The final displayed score is then computed as offset - score. This adjustment effectively shifts the decision boundary, making it simple to interpret: all scores below zero are now considered potential anomalies.

The scatter plot successfully highlights which instances are potential anomalies, but it provides no insight into the underlying reasons for their classification. To understand the features or behaviors that led to an instance being flagged, additional analysis is necessary.

To examine how flagged points relate to the pollution indicators, the plot of a two-dimensional scatter of PM2.5 against the AQI (NowCast) was implemented. Normal observations are shown in green and anomalies in red, with slight transparency to reduce overplotting. This view helps reveal whether anomalous readings cluster at high particulate levels, high AQI values, or in specific combinations of the two.

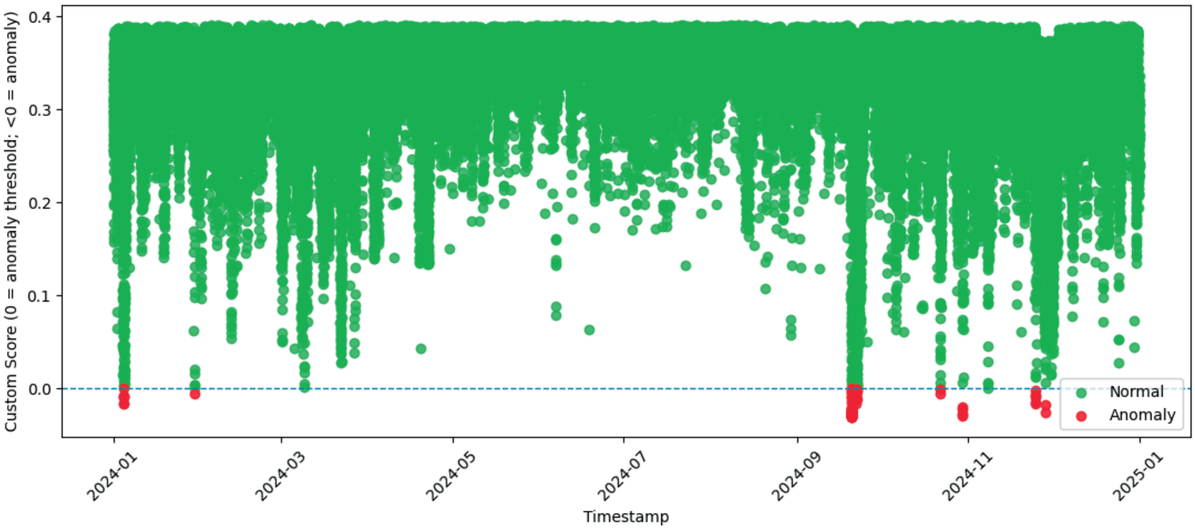

Fig. 3 - Anomaly scores

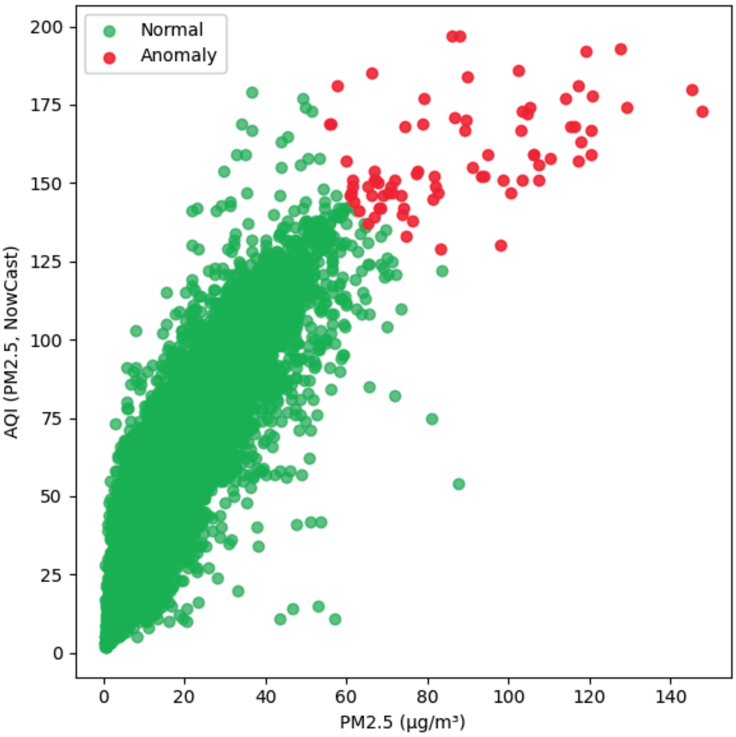

Fig. 4 - PM2.5 vs AQI with anomaly labels

From the PM2.5 vs AQI scatter (Fig.4), many flagged points cluster where PM2.5 is elevated and AQI is high, indicating that extreme particulate concentrations are a key driver of anomaly scores. That said, not all anomalies sit at the upper right. A portion appears at moderate AQI with disproportionately high PM2.5, or at moderate PM2.5 with unexpectedly high AQI, suggesting unusual combinations rather than simple extremes. This pattern implies that interactions between the two indicators, and possibly unmodeled context such as time of day, weather, or district effects, also contribute to why certain observations are classified as anomalies.

Conclusions

The simulations on Kyiv air-quality data using PM2.5 and AQI confirm that Isolation Forest can surface rare and operationally meaningful observations without labels. With contamination set to 0.01, the detector produced a small, manageable set of candidates, and the decision scores aligned with the built-in zero threshold used for labeling. In the time plot, anomalies concentrated within several contiguous intervals, which is consistent with short episodes of unusual air conditions rather than isolated single-point glitches. In the PM2.5–AQI scatter, many flagged points appeared where both indicators were elevated, but a noticeable share involved atypical combinations, such as moderate AQI paired with disproportionately high PM2.5 or vice versa. This pattern suggests that interaction effects, not just univariate extremes, drive a meaningful portion of the alerts.

The approach showed three practical strengths in this setting. It handled multiple features jointly without requiring distributional assumptions, which allowed the model to detect complex anomaly shapes rather than only large z-scores on single variables. It scaled comfortably to thousands of observations with near-linear runtime, making repeated retraining feasible. It provided a direct way to control alert volume through the contamination parameter, which is helpful for matching detection output to analyst capacity.

There are important caveats. Because trees are built with randomness, smaller samples can yield unstable rankings, so larger or aggregate views of the data are preferable for consistent results. The method identifies candidates rather than causes; understanding why a reading is flagged still requires follow-up diagnostics, such as contrasting flagged points with typical PM2.5–AQI ranges, checking meteorological context, or segmenting by district and time. The fixed contamination setting governs expected alert volume rather than true anomaly prevalence, so realized rates can drift if the underlying distribution changes.

Overall, the results indicate that Isolation Forest is a strong first-line detector for unsupervised anomaly screening on environmental data. It efficiently highlights unusual episodes and nonstandard PM2.5–AQI combinations while keeping the analyst workload predictable. For deployment, the findings support pairing the detector with lightweight interpretability steps, optional feature scaling, and simple operational safeguards such as per-district models or rolling recalibration of the threshold to maintain stable performance under changing conditions.

References

1. Yepmo V., Smits G., Lesot M.-J., Pivert O. Leveraging an Isolation Forest to Anomaly Detection and Data Clustering. *Journal of Systems and Software.* 2024. URL:https://www.sciencedirect.com/science/article/abs/pii/S0169023X24000260.
2. DataCamp. Isolation Forest Guide: Explanation and Python Implementation. 2024.URL:https://www.datacamp.com/tutorial/isolation-forest.
3. Xu H., Pang G., Wang Y., Wang Y. Deep Isolation Forest for Anomaly Detection. *arXiv preprint.* 2023. arXiv:2206.06602.URL:https://arxiv.org/pdf/2206.06602.
4. Laskar M. T. R., Huang J. X., Smetana V., Stewart C., Pouw K., An A., Chan S., Liu L. Extending Isolation Forest for Anomaly Detection in Big Data via K-Means. *ACM Digital Library.* 2021. DOI: 10.1145/3460976.

5. Ащепков В. О. Використання моделі isolation forest для виявлення аномалій у даних вимірювань. *Сучасний стан наукових досліджень та технологій в промисловості.* 2024. № 1(27). С. 236–245. DOI:https://doi.org/10.30837/ITSSI.2024.27.236.

6. Міністерство захисту довкілля та природних ресурсів України. Дані моніторингу якості атмосферного повітря в Україні. 2024. URL: https://data.gov.ua/datastore/dump/f6755e36-f910-4482-8260-6a601b8d8da4

Kalashnyk M.O.

MACHINE LEARNING-BASED ANOMALY DETECTION WITH ISOLATION FOREST IN LARGE-SCALE DATA ANALYSIS

This paper presents an applied study of unsupervised anomaly detection with Isolation Forest on large multivariate sensor data. The study implements a concise Python workflow that acquires city-scale measurements for Kyiv, merges reference metadata, removes invalid records, selects two continuous indicators, and trains Isolation Forest with parameters. Temporal analysis shows that anomalies concentrate in contiguous intervals rather than isolated single points, while a two-feature projection indicates that many flags coincide with jointly high values and others arise from atypical value combinations, highlighting multivariate effects.

The study documents practical advantages of Isolation Forest, including minimal distributional assumptions, direct control of alert volume via the contamination parameter, and near-linear scaling that supports repeated retraining. It also notes limitations, such as sensitivity on small samples due to random tree construction, dependence on threshold calibration that can drift across datasets, and limited inherent explainability of individual alerts. Configuration guidance, robustness checks, and lightweight diagnostics are provided to support deployment and to maintain stable performance under changing conditions.

Keywords: *anomaly detection; Isolation Forest; unsupervised learning; threshold calibration; multivariate analysis; scalable analytics; interpretability.*

Калашник М.О.

ВИЯВЛЕННЯ АНОМАЛІЙ НА ОСНОВІ МАШИННОГО НАВЧАННЯ ЗА ДОПОМОГОЮ АЛГОРИТМУ ISOLATION FOREST В АНАЛІЗІ ДАНИХ ВЕЛИКОГО ОБСЯГУ

Ця стаття представляє прикладне дослідження неконтрольованого виявлення аномалій за допомогою Isolation Forest на великих багатовимірних сенсорних даних. Реалізовано рішення мовою Python, яке працює з даними з датчиків моніторингу якості повітря в місті Києві, об'єднує довідкові метадані, видаляє некоректні записи, обирає два неперервні індикатори та навчає за допомогою алгортиму Isolation Forest із заданими параметрами. Часовий аналіз показує, що аномалії концентруються в суміжних інтервалах, а не є ізольованими поодинокими точками, тоді як проекція на дві ознаки вказує, що багато спрацювань збігаються зі спільно високими значеннями, а інші виникають через нетипові комбінації значень, що підкреслює багатовимірні ефекти. Дослідження документує практичні переваги Isolation Forest, включно з мінімальними припущеннями щодо розподілу даних, прямим контролем обсягу сповіщень та майже лінійним масштабуванням, що підтримує повторне перенавчання. Воно також зазначає обмеження, такі як чутливість на малих вибірках через випадкову побудову дерев, залежність від калібрування порогу, яке може *«дрейфувати» на різних наборах даних, та обмежену вбудовану пояснюваність окремих виявлених аномалій. Надано рекомендації з конфігурування, перевірки на стійкість та прості інструменти для підтримки впровадження та збереження стабільної продуктивності в умовах, що змінюються.*

Ключові слова: *виявлення аномалій; Isolation Forest; неконтрольоване навчання; калібрування порогу; багатовимірний аналіз; масштабована аналітика; інтерпретованість.*

УДК 004.8:004.93 **DOI: 10.18372/2073-4751.83.20547**

Козачук О.
orcid.org/0000-0003-3361-6197

ПОРІВНЯЛЬНИЙ АНАЛІЗ МОНОКУЛЯРНОЇ ОЦІНКИ ГЛИБИНИ СУПУТНИКОВИХ ДВОВИМІРНИХ ЗОБРАЖЕНЬ

Державний університет «Київський авіаційний інститут»

e-mail: oleksandrkozachukk@gmail.com

Вступ

Комерційні *3D*-джерела, такі як *Maxar Precision3D* [1], та візуалізації в *Google Maps/Earth* [2] надають високоякісні моделі рельєфу, але їх використання в академічних експериментах обмежене ліцензіями, доступом та охопленням або ж є складними. Відтворення *3D* форми об’єкта зі зображення меншої розмірності дає змогу отримувати додаткову просторову інформацію про сцену за мінімальних вимог до вхідних даних. Зростання обсягів супутникових зображень і доступності картографічних сервісів породжує запит на просту, відтворювану та зіставну оцінку глибини з одиночних *2D*-знімків. Традиційно завдання глибинного моделювання розв’язують за допомогою фотограмметрії та технологій дистанційного зондування, наприклад, аеро лазерного сканування, *LiDAR*, а також із залученням супутникових моделей висот рельєфу (*DEM/DSM*). Водночас комерційні *3D*-сервіси хоч і надають готові побудови рельєфу, але обмежені вартістю, ліцензійними умовами та територіальним покриттям, що ускладнює масштабоване наукове відтворення для окремих регіонів. Карти висот доповнюють дані про місцевість, однак їхня точність і просторове розрізнення варіюють залежно від сенсора та масштабу збору даних [3]. На рис. 1 представлено класичну *3D*-реконструкцію об'єкта, що попередньо потребує багаторакурсної зйомки з відомими параметрами камери. Помітно, що навіть трьох зображень не вистачає для побудови якісної *3D*-моделі.

Однак поява візуальних трансформерів [4] і великих наборів даних дала змогу оцінювати глибину з одного зображення.

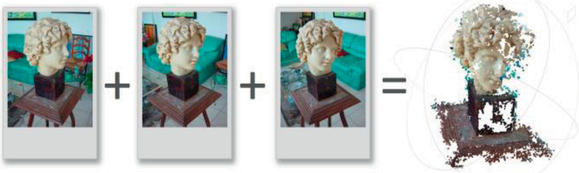

Рис. 1. Класична реконструкція об’єкта спостереження

Попри це, більшість сучасних моделей монокулярної глибини навчалися на загально змістовних, синтетичних наборах даних *MPI-Sintel* [5], *Spring* [6] та інші, що призводить до доменного зсуву під час застосування до супутникових сцен: змінюються геометрія, текстури, масштаб, освітлення; до того ж еталонна метрична глибина для зображень *Google Maps* недоступна для прямої валідації. Тому метою роботи стало перенесення навчання із загальних наборів на домен супутникових знімків.