Preprint - Unpublished - Hasso-Plattner-Institut Technical Report manuscript No.
(will be inserted by the editor)

Geospatial Artificial Intelligence: Potentials of Machine Learning for 3D Point Clouds and Geospatial Digital Twins

Jürgen Döllner

Received: 15.09.2019 / Accepted: date

Abstract Artificial Intelligence (AI) is changing fundamentally the way how IT solutions are built and operated across all application domains, including the geospatial domain. In this article, we briefly reflect on the term "AI" and outline the factors such as Machine Learning (ML) and Deep Learning (DL) that contribute to applying AI successfully for IT solutions. In the main part we discuss AI for the geospatial domain (GeoAI) focussing on 3D point clouds as a key category of geodata, describe their properties and discuss its suitability for ML and DL. In particular, we conclude that 3D point clouds constitute a corpus with similar properties than natural language corpora and formulate a naturalness hypothesis for 3D point clouds. We then outline concepts and examples of ML-based interpretation approaches that compute domain-specific and application-specific semantics for 3D point clouds without having to create explicit spatial models or explicit rule sets. Finally, we will show how ML enables us to efficiently build and maintain base data for digital twins of our environment such as virtual 3D city models, indoor models, or building information models.

Keywords Artificial Intelligence · Machine Learning · 3D Point Clouds · Digital Twins · 3D City Models

1 Introduction

Artificial Intelligence (AI) is changing the way IT solutions are designed, built and operated, whereby AI is not being limited to specific application areas—it is currently finding its way into all industries [PD19]. In particular, for geospatial domains, a fundamental question is how geospatial artificial intelligence (GeoAI) [Vop+18] can improve existing and invent new technology for geospatial information systems (GIS).

Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam
Tel.: +49 331 5509 170
Fax: +49 331 5509 172
E-mail: doellner@hpi.uni-potsdam.de

1.1 The Term "Artificial Intelligence"

The notion "AI" implies a number of well-known conceptual difficulties, such as the definition of "natural", "human" or "general-purpose" intelligence; Kelly [Kel17] discusses these ideas and misunderstandings found in AI. In the general public, AI is often associated with expectations such as simulating or overcoming human intelligence. If AI is pragmatically seen as technological progress, then "AI is going to amplify human intelligence not replace it, the same way any tool amplifies our abilities" as recently argued by LeCun [LeC17].

One of the first AI applications that exemplified these controversies was ELIZA, the famous first chatbot in computer science built by Josef Weizenbaum in 1966. It is a speech-based simulation of a psychologist's interaction with a patient [Pal14]; Weizenbaum, notably, later became one of AI's leading critics. Although ELIZA was intended to demonstrate the limitations of AI, it was considered a serious application by the general public at the time, contrary to the original intention of Weizenbaum. This topic is discussed further by Copeland [Cop93], who analyzes which challenges and obstacles AI needs to solve before "thinking" machines could be constructed. The key to AI is the ability to think rationally, to discover meaning, to generalize and learn from past experiences, and to be intelligent by using learning, thinking, problem solving, perception and languages.

If we look back to IT progress, there was generally never a sharp border between AI and Non-AI technology. In a sense, the term "AI" is frequently used to label such technology that

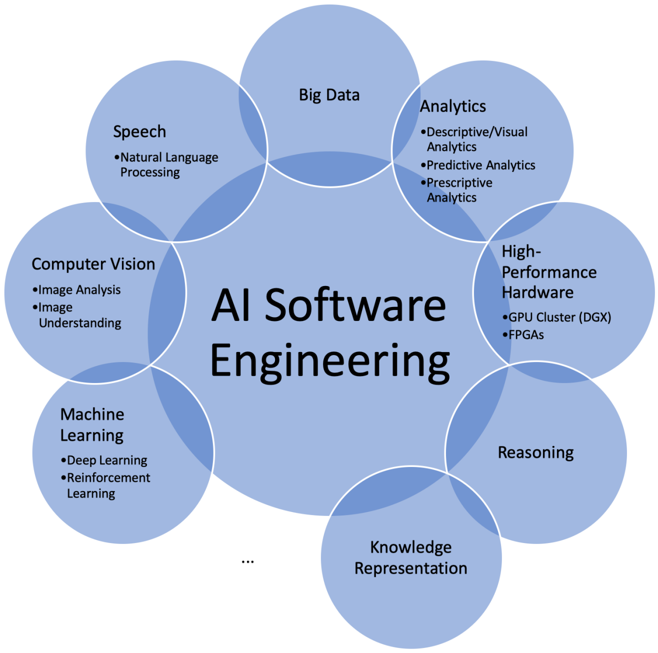

Fig. 1 Main software engineering contributions to AI.

goes beyond current technology boundaries. Consider, for example, an autopilot steering an aircraft: In its beginning, it was perceived as AI, while today it has become a common operating technical component.

1.2 AI from a Software Engineering Perspective

A software engineering perspective allows us to take a more specific look at AI. Here, the coincidence of several independent developments (Fig. 1) is accelerating the implementation and use of AI for new IT solutions:

- **Big Data**: AI and ML require big data to be applied effectively. For example, ML needs training data, which is typically distilled from big data. Big data generally is characterized by a number of criteria, including:
  - large data amounts (volume)
  - rapid data capturing or generation (velocity)
  - different data types and structures (variety)
  - manifold relations among data sets (complexity)
  - high inherent data uncertainty (veracity).

  Big data has turned out to be a key driver for digital transformation processes as summarized in the famous but also controversial [Mar18] statement: "Data is the new oil. Data is just like crude. It's valuable, but if unrefined it cannot really be used." [Hum06] In that respect, geospatial data is big data and the "oil" for the geospatial digital economy. A range of approaches exist to capture and simulate data about our geospatial reality.
- **Analytics**: Analytics is the science of analytical reasoning that aims at providing concepts, methods, techniques, and tools to efficiently collect, organize, and analyze big data. Its objectives include to examine data, to draw conclusions, to get insights, to acquire knowledge, and to support decision making. For all variants such as descriptive, predictive, and prescriptive analytics, ML plays a key role to process, understand, and classify data.
- **Hardware**: The growth of AI comes along with commodity graphics processing units (GPUs) that are evolving to become high performance accelerators for data-parallel computing. Further, there is a growing number of purpose-built systems for DL with fully integrated hardware and software (e.g., NVidia DGX/HGX).
- **Machine Learning**: ML acquires knowledge by building mathematical models based on training data, which are used to predict labels for input data. The underlying model "may be *predictive* to make predictions in the future, or *descriptive* to gain knowledge from data, or both" [Alp14].
- **Deep Learning**: DL, a specific form of representation learning, which in turn is a specific form of ML, is based on Artificial Neural Networks (ANNs) such as Convolutional Neural Networks (CNNs) [GBC16]. It builds representations expressed in terms of simpler representations, i.e., we can build complex concepts out of simpler concepts. "It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government" [LBH15].

1.3 Machine Learning

If ML is applied, we do not *explicitly* program the solution, i.e., ML does not rely on explicit or procedural problem solving strategies but is based on processing and analyzing patterns and inference. Techniques can be classified into supervised ML, unsupervised ML, and reinforcement learning. Supervised ML provides labels for a given set of data based on a previously acquired knowledge by means of labeled training data, while unsupervised ML builds models only on input data without any corresponding output data [Alp14].

Frequently, ML-based solutions describe the input data by means of feature vectors, i.e., by n-dimensional vectors whose numerical components describe selected aspects of a phenomenon to be observed. The feature space builds a corresponding high-dimensional vector space; techniques for dimensionality reduction allow us to manage, process, and visualize that space.

These concepts can be best understood by a simple, abstract example. For data analysis, consider a data labeling that generally takes place as a one-step mapping of an input datset I by an explicitly programmed function f:

- f: Input $\rightarrow$ Label
  $L \leftarrow f(I)$

Supervised ML, conceptually, uses a two-step approach that requires and depends on the training that defines what the mapping is about. For example, it can be based on training data TI with known labels TL by a (general-purpose) training function f_t:

- f_t: TestInput $\times$ TestLabels $\rightarrow$ Model
  $M \leftarrow f_t(TI, TL)$

A corresponding predictive or descriptive function f_{ML} can then process any data set:

- f_{ML}: Model $\times$ Input $\rightarrow$ Label
  $L \leftarrow f_{ML}(M, I)$

"The promise and power of machine learning rest on its ability to generalize from examples and to handle noise" [All+18]. For it, ML offers a high degree of robustness regarding the input data set. A fundamental risk, however, lies in "overtraining" the model. If overtrained, it will be able to identify all the relevant information in the training data, but will fail miserably when presented with the new data. The model becomes incapable of generalizing, i.e., it is *overfitting* the training data. For that reason, "properly controlling or regularizing the training is key to out-of-sample generalization" [Zha+18].

2 3D Point Clouds

2.1 Applications

3D point clouds are becoming ubiquitous data sources for geospatial solutions such as for environmental monitoring, disaster management, urban planning, indoor scanning, or self-driving vehicles. To acquire 3D point clouds, various technologies can be applied including airborne or terrestrial laser scanning, mobile mapping, RGB-D cameras [Zol+18], image matching, or multi-beam echo sounding. As universal 3D representations, 3D point clouds "can represent almost any type of physical object, site, landscape, geographic region, or infrastructure âĂŞ at all scales and with any precision" as Richter states [Ric18], who discusses algorithms and data structures for out-of-core processing, analyzing, and classifying of 3D point clouds as well. Generally, 3D point clouds allow for creating and maintaining geospatial "digital twins". In that respect, 3D point clouds are commonly used as base data for reconstructing 3D models (e.g., digital terrain models, virtual 3D city models, Building Information Models), but can also be understood as 3D model, e.g., if 3D point clouds are dense (Fig. 2).

Fig. 2 High-density 3D point cloud as digital twin of an indoor environment.

2.2 Characteristics

A 3D point cloud represents a set of three-dimensional points in a given coordinate system and can be characterized by:

- **Uniform Representation** – unstructured, unordered set of 3D points (e.g., in an Euclidian space);
- **Discrete Representation** – discrete samples of shapes without restrictions regarding topology or geometry;
- **Irregularity** – expose irregular spatial distribution and varying spatial density;
- **Incompleteness** – due to the discrete sampling, representations are incomplete by nature.
- **Ambiguity** – the semantics (e.g., surface type, object type) of a single point generally cannot be determined without considering its neighborhood;
- **Per-Point Attributes** – each point can be attributed by additional per-point data such as color or surface normal;
- **Massiveness** – depending on the density of the capturing technology, 3D point clouds may consist of millions or billions of points.

2.3 3D Point Cloud Time Series

For a growing number of applications, 3D point clouds are taken and processed with high frequency. For example, if a surveillance system captures its target environment every second, a stream of 3D point clouds results.

If 3D point clouds are captured or generated at different points in time having overlapping geospatial regions, these sets are inherently related. By *3D point cloud time series*, we refer to a collection of 3D point clouds taken at different points in time for a common geospatial region. The collection of 3D point clouds represents, in a sense, a *4D point cloud*.

3D point cloud time series have a high degree of redundancy, which needs to be exploited to achieve efficient management, processing, compression and storage, e.g., separating static from dynamic structures. Redundancy can also be used to improve accuracy and robustness of 3D point cloud interpretations and related predictions.

2.4 Feasibility of ML-Based Approaches

The complete absence of structure, order and semantics as well as the inherent irregularity, incompleteness, and ambiguity commonly make 3D point clouds difficult candidates for procedural and algorithmic programming. Their characteristics, however, allow us to effectively apply ML to 3D point clouds:

- **Big Data** – 3D point clouds are spatial big data that can be cost-efficiently generated for almost all types of spatial environments—big data is a prerequisite for ML-based approaches;
- **Fuzziness** – 3D point clouds show inherent fuzziness and noise as they are sampling shapes by means of discrete representations— ML is particularly handling well fuzzy and noise data;
- **Semantics** – Depending on the concrete application domain, semantic concepts can be defined and corresponding training data can be configured.

Historically, ML and DL applications have focused on object, text, and speech recognition, but today they are applied for data analytics in all domains. ML-based and DL-based approaches offer enormous potential for disruptive innovations in GIS technology. In particular, ML supports computing domain-specific and application-specific information, typically by point classification, point cloud segmentation, object identification, and shape reconstruction. Compared to traditional procedural-like, heuristic, or empiric-based algorithms, ML-based techniques generally have significantly less implementation complexity and higher stability and robustness.

2.5 Naturalness Hypothesis

To understand further, why ML and DL approaches provide effective instruments for analyzing and interpreting 3D point clouds, we take into account the ”Naturalness Hypothesis” known and investigated in many fields such as natural language recognition.

In general, one key approach to ML and DL is to find out whether a given (different) problem domain corresponds to or has similar statistical properties as **large natural language corpora** [JM00]. ML-based approaches have shown extraordinary success in natural language recognition, natural language translation, question-answering, text mining, text comprehension, etc. The most important finding in these areas is that objects (e.g., spoken or written texts) are less diverse than they initially seem: Most human expressions (”utterances”) are much simpler, much more repetitive, and much more predictable than the expressiveness of the language body suggests. This phenomenon can be understood with measures of perplexity and cross-entropy [de +05].

For example, for the domain of software engineering a recent key finding is related to this naturalness hypothesis. It states that the implementation of software “is a form of human communication; software corpora have similar statistical properties to natural language corpora” [All+18], whereby “these utterances can be very usefully modeled using modern statistical methods” [Hin+12]. In other words: Programming languages, in theory, are complex and expressive, but the programs that developers *actually* write are far less expressive, far less complex and strongly repetitive. For that reason, the programs show predictable statistical properties, which can be captured in *statistical language models*.

3D point clouds correspond to such a *form of natural communication* as all geospatial environments are ultimately repetitive regardless of the endless variations they may exhibit. In a sense, 3D point clouds are just “spatial utterances” that can be modeled using statistical methods. 3D point clouds, thereby, constitute *3D point cloud corpora* to which ML technology can be applied taking advantage of the statistical distributional properties estimated over representative point cloud corpora.

Following the schema for an argumentation stated for software engineering [Hin+12], an ML-centric natural hypothesis for 3D point clouds could be formulated as follows:

> *3D point clouds, in theory, are complex, expressive and powerful, but the 3D point clouds actually generated are far less complex, far less expressive and strongly repetitive. Their predictable statistical properties can be captured in statistical language models and leveraged for geospatial data analysis.*

General ML and DL approaches, however need to be adapted to the characteristics of 3D point clouds. “Most critically, standard deep neural network models require input data with regular structure, while point clouds are fundamentally irregular: Point positions are continuously distributed in the space, and any permutation of their ordering does not change the spatial distribution.” [Wan+18]

3 ML-Based Point Cloud Interpretation

3D point clouds provide cost-efficient raw data for building digital twins at all scales but they are purely geometric data without any structural or semantics information about the objects they represent. Motivated by the Naturalness Hypothesis, ML can be applied to understand and recognize that information: ML turns into a powerful technique if it comes to discrete irregular, incomplete, and ambiguous data of a given corpus—exactly what characterizes 3D point clouds.

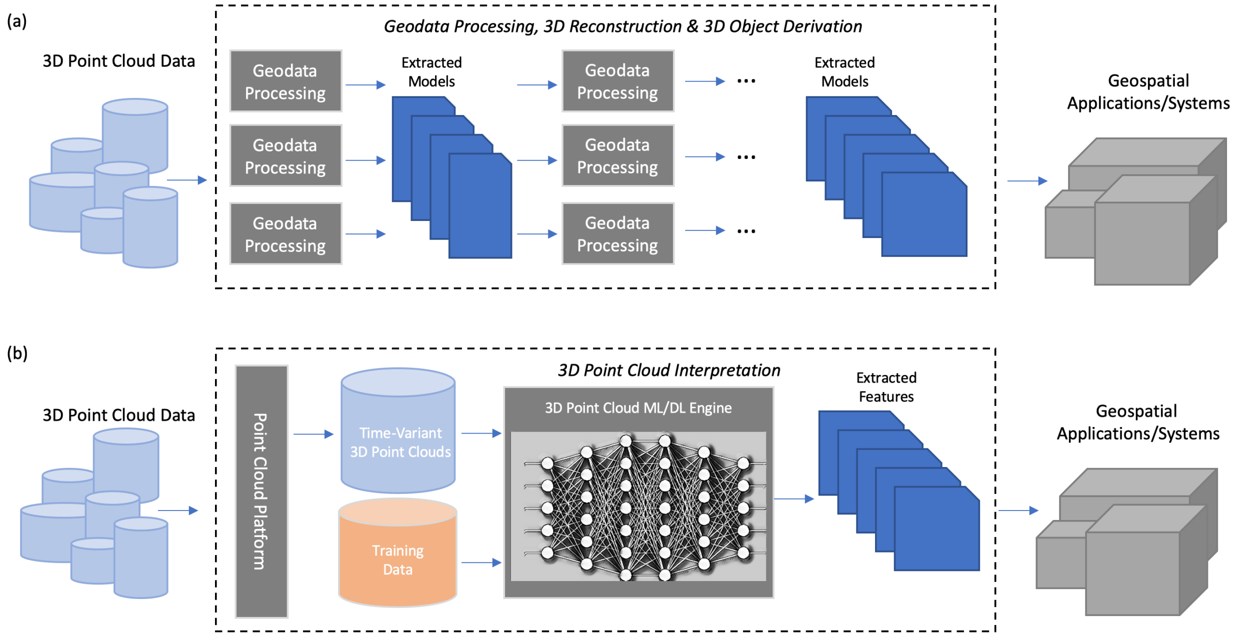

Fig. 3 3D point cloud processing: classical workflow (a) based on 3D reconstruction, 3D modeling, and object derivation; (b) ML/DL workflow based on 3D point cloud interpretation.

3.1 Interpretation Concept

The ML-based processing of 3D point clouds is based on the concept of *interpretation* known from programming languages. Analytics and semantics parsing do not require steps that "compile" raw data into higher level representations. To process data, for example, the *PointNet* neural network "directly consumes point clouds and well respects the permutation invariance of points in the input" and provides a "unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing" [Qi+16a].

Applications or services that require spatial information encoded in 3D point clouds specify the exact set of features to be extracted and the spatial extent to be searched. The available feature types depend on how the ML and DL subsystems have been trained before. The analysis processes the request by triggering the evaluation to obtain the results. At no point in time, 3D point clouds are pre-processed or pre-evaluated nor do they require any intermediate representations, i.e., the interpretation works on-demand on the raw point cloud data.

In Fig. 3 the classical geoprocessing workflow (a) is compared to the workflow enabled by 3D point cloud interpretation (b). While (a) is based on generating more and more detailed and semantically well-defined representations, the workflow (b) operates on raw data, extracting the demanded features using corresponding training data. The ML/DL engine supports core features such as:

- **Point Classification**: According to defined point categories (e.g., vegetation, built structures, water, streets) labels are computed and attached as per-point attributes together with the probability for this category assignment. For example, Roveri et al. "automatically transform the 3D unordered input data into a set of useful 2D depth images, and classify them by exploiting well-performing image classification CNNs" [Rov+18].
- **Point Cloud Segmentation**: Segmentation as a core operation for 3D point clouds helps removing clustering and subdividing large point clouds. Typically it is based on identifying 3D geometry features such as edges, planar facets, or corners. ML and DL, in contrast, allow us to take advantage of semantic cues and affordances found in 3D point clouds. For example, we can segment "local geometric structures by constructing a local neighborhood graph and applying convolution-like operations on the edges connecting neighboring pairs of points, in the spirit of graph neural networks" [Wan+18].
- **Shape Recognition**: Shapes are essential for understanding 3D environments. To recognize them, a combined 2D-3D approach [Sto+19a] consists of generating 2D renderings from 3D point clouds that are evaluated by image analysis. For this purpose, CNNs can combine "information from multiple views of a 3D shape into a single and compact shape descriptor offering even better recognition performance" [Su+15] compared to approaches that operate directly on a 3D point cloud representation. Large, general-purpose repositories of 3D objects, in addition, provide a solid training data base.
- **Object Classification**: Applications generally require object-based information to be extracted from 3D point

clouds, e.g., signs and poles of the street space. Based on classified and segmented 3D point clouds, CNNs based upon volumetric representations or CNNs based upon multi-view representations are commonly applied to this end; [Qi+16b] give an overview of the space of methods available.

The non-uniform sampling density typically found in 3D point clouds represents a key challenge for ML and DL feature learning. "Features learned in dense data may not generalize to sparsely sampled regions. Consequently, models trained for sparse point cloud may not recognize fine-grained local structures" [Qi+17]; Qi et al., therefore, propose a hierarchical CNN that operates on nested partitions of an input point set.

ML-based interpretation enables us to implement generic analysis components for 3D point clouds. As no intermediate representations are required, analysis results are only created once they are requested, and they are only computed for the specific region the application has defined. Among the advantages of this approach are:

- *Configurability*: The ML/DL training data together with feature vector definitions allow for many label types to be predicted. For it, the generic, domain-independent mechanism offers a high degree of configurability.
- *On-Demand Computation*: Downstream services allow for on-demand computation. For many classifications, the intepretation can be executed even in real-time (e.g., object detection out of point clouds for surveillance purposes).
- *Service-Based Computing*: The approach is scalable as it can be fully mapped to a service-oriented architecture and scalable hardware (e.g., GPU clusters), built by lower-level and higher-level services and mashups.
- *Raw-Data Processing*: Storage and handling of massive 3D point clouds, including time-variant ones, can be optimized independently as the interpretation only requires fast spatial access to point cloud contents.
- *Storage Efficiency*: There are no pre-selected or pre-built models or intermediate representations. The approach therefore works well for massive or time-varying 3D point clouds. In particular, the original precision of the raw data is never reduced as raw data is feed directly into the ML/DL processes.

3.2 Examples

In a joint research project, we are developing a robust, high-performance engine for experimental ML-based geospatial analytics. It provides features to store, manage, and visualize massive 4D point clouds.

In Fig. 4, 3D point cloud interpretation has been used to extract the underground infrastructure entities in the street space based on mobile mapping [WRD19]. The visualization shows the extracted tubes and also detects street elements such as manhole covers. In Fig. 5, 3D point cloud interpretation has identified and classified points according to different categories of street space furniture.

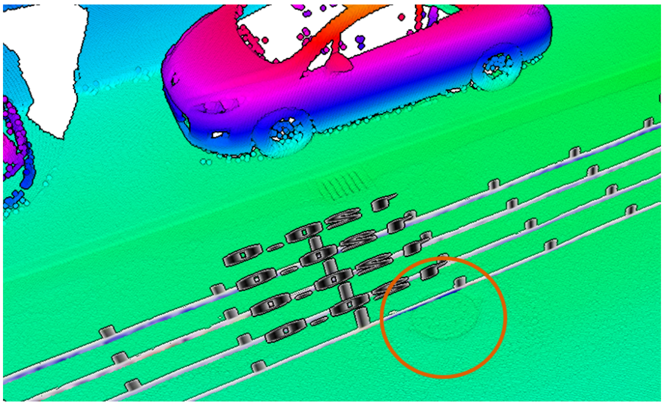

Fig. 4 Example of an analysis of underground structures based on 3D point clouds captured by radar combined with four trajectories of ground penetrating radar data. The point cloud is colored with a height gradient. Dataset from the city of Essen, Germany.

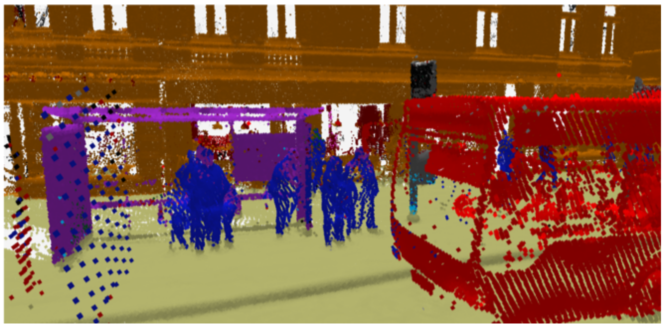

Fig. 5 Example of point classification for a typical street space scenario. Ground, buildings, vehicles, pedestrians, and different street furniture objects are classified with a PointNet-based approach and are visualized by different colors.

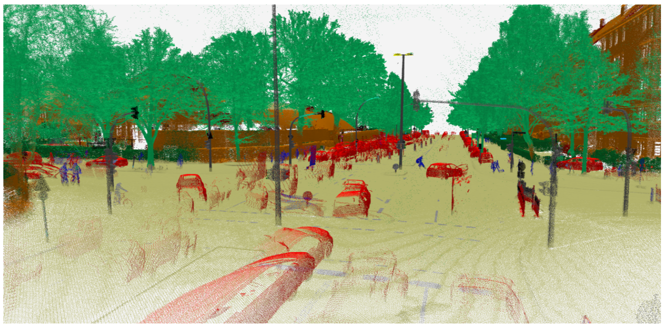

Fig. 6 Example of object classification (based on PointNet) within a dynamic scenario given by a time series of 3D point clouds.

Fig. 6 shows how time series of 3D point clouds, for example taken during a mobile scan, can be interpreted to extract relevant objects such as street signs, vehicles, vegetation etc.

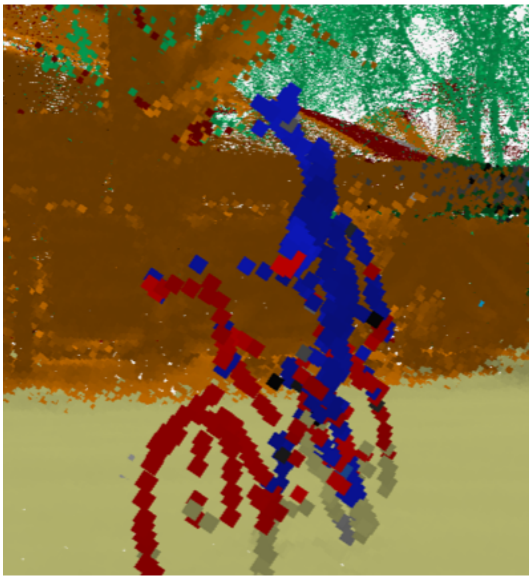

Fig. 7 Example of a composite classification (data by courtesy of Stadt Hamburg). Bicycle and cyclist are segregated by DL-based classification.

In Fig. 7, a composite classification is illustrated: The bike and the person riding the bike are identified and then can be combined as ’person-riding-a-bike’. High-level abstractions can be built in a post-processing step or as part of the ML/DL processes.

4 AI for Digital Spatial Twins

A key demand in digital transformation processes represent *digital twins*, i.e., digital representations that reflect form, structure, relations, and state of real-world objects [El 18]. 3D point clouds represent raw data of geospatial entities in a *well-defined, consistent, and simple* way, in particular, for spatial environments such as indoor spaces [Sto+19b], building information models, and cities. The reconstruction of digital twins based on explicitly defined 3D model schemata is a notoriously cumbersome and error-prone process (e.g., virtual 3D city models with high level of detail such as CityGML LOD3 or LOD4 [Löw+16]) as the 3D models have a strict scope of expressiveness. Whether we apply strong mathematics or fine-tuned heuristics, a reconstructed 3D model almost always lacks details and it can hardly mirror weakly sampled, unusual, or fuzzy entities.

ML-based interpretation can both efficiently and effectively, analyze and organize 3D point clouds without being restricted by explicitly defined modeling schemata. Above all, it flexibly generates semantics on-demand and on-the-fly, that is, it helps “healing” one of the biggest weaknesses of 3D point clouds—the lack of structure and semantics. There is virtually no limitation for the specific types of 3D objects,

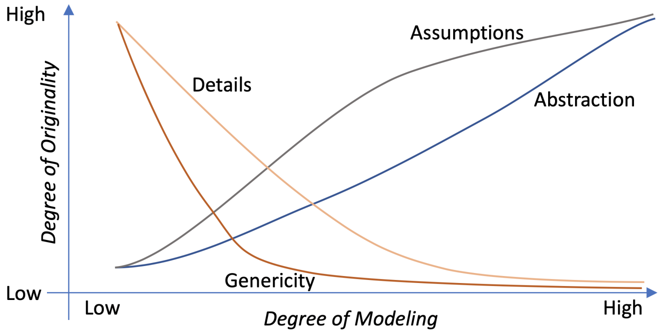

Fig. 8 The higher the degree of modeling, the less of the original raw data is preserved as more and more assumptions and abstraction are introduced. Hence, data details disappear and less general-purpose models result.

structures, or phenomena that can be identified and extracted by ML-based 3D point cloud interpretation.

In addition, ML-based interpretation operates on the raw geospatial data, i.e., it retains a high degree of originality, while using only a moderate degree of explicit modeling for extracted features (Fig. 8). This, on the one hand, simplifies management and storage, in particular, if it comes to time series. On the other hand, it helps identifying and classifying ambiguous or fuzzy entities as need, for example, to robustly and automatically build spatial digital twins.

5 Conclusions

AI is radically changing programming paradigms and software solutions in all application domains. In the geospatial domain, the data characteristics are particularly suitable for ML and DL approaches as geodata fits into the concept of a “linguistic corpus” as sketched in the context of the naturalness hypothesis. ML-based analysis and extraction of features out of 3D point clouds, for example, can be used to supply application-specific, domain-specific and task-specific semantics.

Above all, ML-based interpretation of 3D point clouds enables us to transcend explicit geospatial modeling and, therefore, to overcome complex, heuristics-based reconstructions and model-based abstractions. Insofar, AI technology can be used to simplify and accelerate workflows for geodata processing and geoinformation systems. Of course, crucial ML-related challenges result from the demand for effective training data and efficient feature representations.

Last but not least, AI-based solutions offer drastic simplifications in the dimension of software engineering. Large parts of today’s implementations (often historically grown with large amounts of so called technical debts) will be replaced by ML and DL “black box” subsystems, which have far less management and software development complexity. In particular, most heuristics-based, explicitly programmed analysis routines, which tend to be difficult to parameterize, can be migrated this way. ML-based approaches, in the long run, will “eat up” most of today’s explicitly programmed GIS implementations.

Acknowledgements We thank Benjamin Hagedorn, Johannes Wolf, Rico Richter, and Vladeta Stojanovic for their contributions to HPI’s ML research. We also thank the GraphicsVision.AI association for their support by the Malaga research retreat and pointcloudtechnology.com for providing us the PunctumTube 3D point cloud platform. This research work was partially supported by the German Federal Ministry of Education and Research (BMBF) as part of the research grands for PunctumTube and GeoPortfolio.

References

[Cop93] Jack Copeland. *Artificial Intelligence: A Philosophical Introduction*. Wiley-Blackwell, 1993.

[LeC17] Yann LeCun. “AI is going to amplify human intelligence not replace it”. In: *FAZ Netzwirtschaft* (20.06.2017). URL: `https://www.faz.net/-gqm-8yrxk`.

[JM00] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 1st. Prentice Hall PTR, 2000. ISBN: 0130950696.

[de +05] Pieter-Tjerk de Boer et al. “A Tutorial on the Cross-Entropy Method”. In: *Annals of operations research* 134.1 (2005), pp. 19–67. ISSN: 0254-5330.

[Hum06] Clive Humby. 2006. URL: `http://www.humbyanddunn.com`.

[Hin+12] Abram Hindle et al. “On the Naturalness of Software”. In: *Proceedings of the 34th International Conference on Software Engineering*. ICSE ’12. IEEE Press, 2012, pp. 837–847.

[Alp14] Ethem Alpaydin. *Introduction to Machine Learning*. 3rd ed. Adaptive Computation and Machine Learning. MIT Press, 2014. ISBN: 978-0-262-02818-9.

[Pal14] Kerstin Palatini. “Joseph Weizenbaum, Responsibility and Humanoid Robots”. In: *Robotics in Germany and Japan. Philosophical and Technical Perspectives*. Ed. by M. Funk and B. Irrgang. Peter Lang, 2014, 163ff.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521 (2015).

[Su+15] Hang Su et al. “Multi-view Convolutional Neural Networks for 3D Shape Recognition”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV ’15. IEEE Computer Society, 2015, pp. 945–953.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: `http://www.deeplearningbook.org`.

[Löw+16] Marc-Oliver Löwner et al. “Proposal for a New LOD and Multi-Representation Concept for CityGML”. In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4 (2016).

[Qi+16a] Charles Ruizhongtai Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *CoRR* abs/1612.00593 (2016).

[Qi+16b] Charles Ruizhongtai Qi et al. “Volumetric and Multi-View CNNs for Object Classification on 3D Data”. In: *CoRR* (2016).

[Kel17] Kevin Kelly. *The AI Cargo Cult: The Myth of a Superhuman AI*. Tech. rep. Backchannel, 2017.

[Qi+17] Charles Ruizhongtai Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *CoRR* (2017).

[All+18] Miltiadis Allamanis et al. “A Survey of Machine Learning for Big Code and Naturalness”. In: *ACM Comput. Surv.* 51.4 (2018), 81:1–81:37.

[El 18] Abdulmotaleb El Saddik. “Digital Twins: The Convergence of Multimedia Technologies”. In: *IEEE MultiMedia* 25.2 (2018), pp. 87–92.

[Ric18] Rico Richter. *Concepts and Techniques for Processing and Rendering of Massive 3D Point Clouds*. Dissertation, 2018.

[Rov+18] Riccardo Roveri et al. “A Network Architecture for Point Cloud Classification via Automatic Depth Images Generation”. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 2018, pp. 4176–4184.

[Vop+18] Trang Vopham et al. “Emerging trends in geospatial artificial intelligence (geoAI): Potential applications for environmental epidemiology”. In: *Environmental Health* 17 (2018).

[Wan+18] Yue Wang et al. “Dynamic Graph CNN for Learning on Point Clouds”. In: *CoRR* abs/1801.07829 (2018).

[Zha+18] Chiyuan Zhang et al. “A Study on Overfitting in Deep Reinforcement Learning”. In: *CoRR* abs/1804.06893 (2018).

[Zol+18] Michael Zollhöfer et al. “State of the Art on 3D Reconstruction with RGB-D Cameras”. In: *Computer Graphics Forum* 37 (2018), pp. 625–652.

[PD19] Marc Purdy and Paul Daugherty. *How AI Industry Profits and Innovation Boosts*. Accenture, 2019.

[Sto+19a] Vladeta Stojanovic et al. “Classification of Indoor Point Clouds Using Multiviews”. In: *The*

24th International Conference on 3D Web Technology, Web3D, Los Angeles, July 26-28, 2019. 2019, pp. 1–9.

[Sto+19b] Vladeta Stojanovic et al. "Generation of Approximate 2D and 3D Floor Plans from 3D Point Clouds". In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2019, Volume 1: GRAPP, 2019*. 2019, pp. 177–184.

[WRD19] Johannes Wolf, Rico Richter, and Jürgen Döllner. "Techniques for Automated Classification and Segregation of Mobile Mapping 3D Point Clouds". In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2019, Volume 1: GRAPP, Prague, Czech Republic, February 25-27, 2019*. 2019, pp. 201–208.

[Mar18] Bernard Marr. *Here's Why Data Is Not The New Oil*. Forbes, March 15, 2018. URL: www.forbes.com.