Environmental Modelling and Software 151 (2022) 105362

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft

PyLUSAT: An open-source Python toolkit for GIS-based land use suitability analysis

Changjie Chen [a,b,*], Jasmeet Judge [b], David Hulse [a]

[a] Florida Institute for Built Environment Resilience, University of Florida, 720 SW 2nd Ave, Gainesville, 32611, FL, United States

[b] Center for Remote Sensing, Department of Agricultural and Biological Engineering, University of Florida, 1741 Museum Road, Gainesville, 32611, FL, United States

ARTICLE INFO

Keywords:
Land use planning
Suitability modeling
GIS
Geospatial analysis
Python
Open-source software

ABSTRACT

Desktop software applications, such as *ArcGIS* and *QGIS*, provide GIS tools for conducting suitability analysis, a fundamental step in formulating a land-use plan. When building complex suitability models with these applications, there are several limitations, including operating system (OS) dependency, lack of dedicated modules, less efficient model building process, and difficult, if not impossible, deployment on a computing cluster. In attempting to address the challenges, this paper introduces PyLUSAT: Python for Land Use Suitability Analysis Tools, an open-source package dedicated to fulfilling tasks in a suitability modeling workflow. PyLUSAT tools were evaluated against comparable tools in *ArcMap 10.4* with respect to accuracy and computational efficiency. Results showed that PyLUSAT tools were two to ten times faster depending on the job's complexity while generating outputs with equivalent spatial accuracy. Besides, PyLUSAT features cross-platform compatibility and high extensibility, allowing leveraging parallel computing in land-use suitability modeling, and permitting automation and customization.

1. Introduction

Since its first introduction in the late 1960s, Geographic Information System (GIS)-based suitability analysis continues playing a significant role in contemporary practices of land use planning (Collins et al., 2001; Malczewski, 2004; McDowell et al., 2018; Seyedmohammadi et al., 2019). It is particularly favored by land-use practitioners because the technique can effectively synthesize spatial analytics, expert knowledge, and community values, all of which, from a planning perspective, are critical factors to consider when making land-use decisions. In practice, suitability analysis is usually performed using off-the-shelf (OTS) GIS software via a graphical user interface (GUI), such as ArcGIS and QGIS (Abdullahi et al., 2015; Mesgaran et al., 2017). And since both software applications provide a Python site package (namely ArcPy and PyQGIS) as an interface to their respective functions, suitability analysis can also be performed through scripting. Because it allows researchers to automate and customize processes, the second approach is preferable when dealing with complex spatial models.

However, suitability modeling frameworks offered by desktop GIS are either constrained to only run inside the application or requiring of the main application installed on the desktop computer (Steiniger and Hunter, 2013). Relying solely on a desktop GIS also creates obstacles for suitability mapping to become an integral module of a larger framework that involves analyses, for example, transportation and hydrologic modeling, completed by third-party software. For developing countries, where land use suitability mapping is arguably more useful and needs to be extensively applied, proprietary GIS software, such as ArcGIS (43% share of the global market), is cost-prohibitive for those countries who have not yet received any financial aid on purchasing licenses (Chang et al., 2009; Esri, 2015). Additionally, ArcGIS is only available on Windows platform, which prevents modeling land use suitability (for a large territory) on a supercomputer since most of them run on a Linux-based OS (Strohmaier et al., 2020; Tang and Matyas, 2018).

Instead, a programming library could be an alternative solution to desktop GIS, that can help lower the mentioned barriers. At the time of writing, there exist a wide variety of programming libraries (written in Python) that are developed to study urban-related questions, with examples being *UrbanSim*—simulating urban real estate markets, *OSMnx*—analyzing street networks, and *UrbanAccess*—measuring transit accessibility (Waddell, 2010; Boeing, 2017; Blanchard and Waddell, 2017). Despite being an influential technique for land use planning, there lacks a domain-specific programming library for land

* Corresponding author. Florida Institute for Built Environment Resilience, University of Florida, 720 SW 2nd Ave, Gainesville, 32611, FL, United States.
E-mail address: chj.chen@ufl.edu (C. Chen).

https://doi.org/10.1016/j.envsoft.2022.105362
Received 16 September 2021; Received in revised form 9 February 2022; Accepted 14 February 2022
Available online 3 March 2022

use suitability analysis. To fill in this gap, we present a free and open-source software (FOSS) package—PyLUSAT: Python for Land-Use Suitability Analysis Tools. PyLUSAT provides a wide-ranging set of tools that is capable of carrying out spatial and aspatial operations entailed by suitability analysis. Moreover, PyLUSAT is cross-platform and can run on a supercomputer to take advantage of High-performance computing (HPC).

In this paper, Section 2 reviews the literature and explains the choice of adopting a vector-based GIS approach. Section 3 describes three main categories of PyLUSAT functions including geospatial, transformation, and aggregation functions. Section 4 presents validation and performance evaluation of PyLUSAT functions. Finally, Section 5 summarizes the paper and briefly discusses future research.

2. Literature review

2.1. Land use suitability analysis

Malczewski (2004), defined land use suitability analysis as the analysis aiming at "identifying the most appropriate spatial pattern for future land uses." The procedure involving superimposing semi-transparent maps proposed by McHarg (1969) is often considered the precursor of GIS-based land use suitability analysis that are known today (Steinitz, 1976; Collins et al., 2001). The analysis has very distinct applications at different spatial scales. At macro (regional) scales, where land is seen as a resource (not attached to specific uses), applications of suitability analysis are more focused on addressing food production (Mesgaran et al., 2017; Scopesi et al., 2020; Yohannes and Soromessa, 2018), ecological function (Marull et al., 2007; Owusu et al., 2017), and regional management (Bolleter et al., 2021; Ingmire and Patri, 1971; Steiner et al., 2000). On the other end of the spectrum, i.e., micro (landscape) scales, the connotation shifts to the human employment of land parcels, and consequently applications of suitability analysis focuses more addressing site search (Xiao et al., 2002), and urban planning issues (Abdullahi et al., 2015; Berry and BenDor, 2015).

From a historical perspective, methodological advancement in land use suitability analysis went in parallel with the development and evolution of GIS (Malczewski, 2004). Computer-assisted overlay was the predominant method for suitability analysis in the early years of GIS between 1950s and 1970s (Hopkins, 1977; Steinitz, 1976). Example software at this stage includes SYMAP and GRID systems developed by the Harvard Labortory for Computer Graphics and Spatial Analysis (Lyle and Stutz, 1983). At this stage, GIS software, like computers, are still only available to large institutions (Coppock and Rhind, 1991). With the advancement in computing hardware, the 1980s have witnessed perhaps the greatest improvement in GIS software, in that general-purpose GIS that prevails today started to emerge (Collins et al., 2001). The primary method used for suitability mapping has also changed to *Map Algebra* and Weighted Linear Combination (WLC) in this period (Tomlin, 1990). Example software includes ARC/INFO, Idrisi, GRASS (Eastman, 1997; Neteler et al., 2012). After 1990s, general-purpose GIS has becomes more mature and has gradually become the dominant tool used for suitability modeling (Collins et al., 2001; Malczewski, 2004). In this period, the Multi-Criteia Decision Analysis (MCDA) has been introduced to land use suitability analysis and was considered having a stronger theoretical basis than the WLC method (Thill, 1999; Seyedmohammadi et al., 2019). Fuzzy logic (Jiang and Ronald Eastman, 2010) and evolutionary programming (Xiao et al., 2002; Cao et al., 2012) are some other methodologies applied to evaluating land use suitability in more recent literature. However, these methods are out of the scope of this paper, hence not discussed in details here.

Consistent improvements in general-purpose GIS have made such systems popular for suitability analysis in recent years (Berry and BenDor, 2015; Bolleter et al., 2021; Girmay et al., 2018; Mesgaran et al., 2017; Seyedmohammadi et al., 2019; Scopesi et al., 2020). For example, several GIS offer graphical modeling environments, such as ArcGIS ModelBuilder, QGIS Graphical Modeler, and Idris Macro Modeler, to automate complex process in suitaiblity modeling, which led to the developments of ILARIS (Jones and Grant, 2007) and LUCIS (Carr and Zwick, 2007) models. Despite these successful applications, there are still a few limitations, as mentioned in Section 1, in today's GIS software, that can be further complemented by a portable, free, cross-platform, and domain-specific package such as PyLUSAT.

2.2. Vector-based GIS approach

In GIS literature, many studies have discussed which data format—vector (a view based on discrete objects) or raster (a view based on continuous fields)—leads to a better representation of our world. For example, Egenhofer and Frank (1987) proposed an object-oriented data model to address the deficiencies of storing, manipulating, and querying spatial data in conventional relational database management systems (RDBMS). However, Goodchild (1989) argued that the object view is a continuation of a tradition inherited from *Cartography*, and the field representation is more realistic and accurate. Bian (2007) went a step further to generalize all environmental phenomena into three categories and discussed each category's applicability of adopting the object representation. Vector and raster representations operate on two distinct sets of logic, but they both stem from and partially exhibit human perceptions of the world. Instead of viewing them as competing or conflicting, we should deem vector and raster as complementary representations of the real world as suggested by Couclelis (1992) "people manipulate objects, but cultivate fields." Today, we find the two representations and their corresponding analytical methods co-exist in harmony in GIS applications, in spite of the attempts to develop a "general theory" of geographic representation (Liu et al., 2008; Goodchild et al., 2007; Winter and Frank, 2000).

Vector-based GIS routines were implemented in PyLUSAT primarily because it fits well conceptually with the object-oriented nature of Python, as the same choice made by other land-use modeling programs (Barreira-González, Gómez- Delgado, and Aguilera-Benavente, 2015; Bolte et al., 2007). Furthermore, such approach has two additional advantages over the raster representation. First, vector-based GIS tools bypass the *modifiable areal unit problem* (MAUP) since measurements or statistics are not derived from a raster grid whose cell size is arbitrarily determined (Jelinski and Wu, 1996), but directly from individual objects. Secondly, a vector-based land-use model is more politically relevant since objects, e.g., property parcels, reflect and honor the land ownership. Just as Couclelis (1992, p. 67) argued that "it is at this lowest level of real estate …, that we find the cultural grounding of the notion of space as objects." As PyLUSAT deals with suitability analysis for urban (land use) planning, that is, the human employment of land parcels on micro (landscape) scales, vector-based GIS is a more appropriate choice.

2.3. Developing open-source geospatial tools in Python

We used Python to develop PyLUSAT package. At the conceptual stage, we considered using R (R Core Team, 2021) because it integrates well with existing GIS software and has several established packages dedicated to spatial analysis e.g., *sp: Classes and Methods for Spatial Data* and *sf: Simple Features for R* (Bivand et al., 2013; Pebesma, 2018). Although R is strong in statistical computing, Ma et al. (2020) showed that in a land use regression (LUR) application, a Python-based package, PyLUR, offers greater processing efficiency and software stability comapring to RLUR, a package with similar functions implemented in R. In addition, Python is a general-purpose language, a preferable feature to software development (*custom programing*), whereas R more commonly is used to apply existing methods of analysis (Muenchen, 2017).

There is an abundant resource of open-source geospatial packages in Python (Carreira, 2016). For I/O-related tasks, `Fiona` (Gillies and

Others, 2011), an API of the OpenGIS Simple Features Reference Implementation (OGR), is capable of handling a variety of forms of vector data from local *Shapefiles* to data on stored on a PostGIS/PostgreSQL database. Shapely (Gillies and Others, 2007), a Python API of the Geometry Engine Open Source (GEOS), provides methods related to set-theoretic analysis and manipulation of planar features. GeoPandas (Jordahl et al., 2019) is arguably the most powerful open-source geospatial package when it comes to vector GIS, in that it merges the functionalities of the two packages above, plus the data structures of pandas (McKinney, 2010), a fast, flexible, high-level building block for data analysis in Python. For raster data inputs, Rasterio (Gillies and Others, 2013) based on the Geospatial Data Abstraction Library (GDAL) can deal with I/O related tasks as well as converting from (or to) vector data.

3. PyLUSAT functions for suitability analysis

Functions in PyLUSAT adhere to vector-based GIS routines, i.e., taking vector input and generating vector output after performing one or more geospatial operations. In most cases, a PyLUSAT function takes a collection of polygons representing land units as input, in which a single uniform land-use decision can be made. For example, these polygons can be property parcel data or an output of a series of GIS operations using multiple datasets, such as the *Integrated Decision Units* (IDUs) (Bolte et al., 2007; Wu et al., 2015).

The classic suitability analysis framework contains three steps: (1) evaluate land units based on identified criteria, (2) transform the measurements to a uniform suitability scale, and (3) combine the results to generate a single suitability score for each land unit (Steiner et al., 2000; Marull et al., 2007). The three tasks of suitability analysis can be carried out respectively by the three types of PyLUSAT functions: geospatial functions, transformation functions, and aggregation functions. The rest of this section explains in greater detail how the three categories of PyLUSAT's functions operate.

3.1. Geospatial functions

The suitability of a parcel of land for a given land use depends greatly on its spatial relationship with the amenities, institutions, service providers, and natural features in the area. Therefore, the geospatial functions are crucial to understanding land-use suitability. PyLUSAT provides a variety of functions to evaluate spatial relationships among vector objects, such as calculating distance and density, examining topological predicates (Strobl, 2008), and interpolation.

3.1.1. Nearest Neighbor search

Distance, a direct measurement of proximity, is one of the most fundamental factors determining land-use suitability. For example, due to agglomeration effects, land parcels in the vicinity of a central business district (CBD) (Ottaviano and Thisse, 2004) or urban sub-centers (Yang et al., 2019) are highly suitable for commercial uses, whereas residential uses often favor parcels remote from nuisances, such as quarries or poultry farms. In these examples, land parcels/units can be conceived as a source set paired with a second set, we call *targets*, where distance from a source to the nearest target affects suitability. Thus, calculating distance amounts to a search for the Nearest Neighbor (NN). To generalize, given a set of m sources, i.e., $S = \{\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_m}\}$, and a set of n targets, i.e., $T = \{\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_n}\}$, the distance between $\mathbf{s_i}$ and $\mathbf{t_j}$, the corresponding NN of $\mathbf{s_i}$ in T, determines $\mathbf{s_i}$'s suitability, from a proximity standpoint. We use $d_e(\mathbf{s_i})$ and $d_m(\mathbf{s_i})$ for the *Euclidean* and *Manhattan* distances, respectively, as defined by the following equations.

$$d_e\left(\mathbf{s_i}\right) = \min_j \|\mathbf{s_i} - \mathbf{t_j}\|_2, \text{for} j = 1, 2, \ldots, n \tag{1}$$

$$d_m\left(\mathbf{s_i}\right) = \min_j \|\mathbf{s_i} - \mathbf{t_j}\|_1, \text{for} j = 1, 2, \ldots, n \tag{2}$$

, where $\mathbf{s_i} = (s_{i1}, s_{i2})'$ and $\mathbf{t_j} = (t_{j1}, t_{j2})'$ are in $\mathbb{R}^2$; $\|\|_2$ denotes the Euclidean (or l_2) norm, i.e., $\|\mathbf{s_i} - \mathbf{t_j}\|_2 = [(s_{i1} - t_{j1})^2 + (s_{i2} - t_{j2})^2]^{1/2}$; and $\|\|_1$ denotes the Absolute-value (or l_1) norm, i.e., $\|\mathbf{s_i} - \mathbf{t_j}\|_1 = (|s_{i1} - t_{j1}| + |s_{i2} - t_{j2}|)$. Note that, here, we use a single representative point, usually a centroid, to represent a land unit. Fig. 1 provides an illustration of the NN search based on the Euclidean distance.

Evidently, the computational complexity for this process will be $\mathcal{O}(mn)$, if a brute-force search was conducted (Xiao and George, 2016). However, it would be overly expensive when the magnitude of mn is large. The function pylusat.distance.to_point uses scipy.spatial.ckdtree which implements a *sliding midpoint* method to construct *KDtree* (K-dimensional tree) objects to search for the NN more efficiently (Virtanen et al., 2020). Like regular KDtree construction, the sliding midpoint method attempts to split the data at the median (midpoint) on each axis first, but the plane will then slide to the closest point if a trivial (all points on one side of the plane) split occurs (Maneewongvatana and Mount, 1999). Such a process results in a KDtree whose height need not to be $\mathcal{O}(log(n))$, the height of a regular KDtree, which in turn makes the construction of KDtree less computationally expensive. Maneewongvatana and Mount (1999) have shown that this implementation offers a better performance in NN search, especially when data are clustered along one axis of $\mathbb{R}^2$.

3.1.2. Affine transformation

When the target set is comprised of line features, an efficient approach to compute distances is to transform the line features from individual vector shapes to a raster grid. The function pylusat.distance.to_line implements this approach by using the *Affine transformation* defined by the following equation.

$$\begin{bmatrix} v_x \\ v_y \\ 1 \end{bmatrix} = \begin{bmatrix} c & 0 & l \\ 0 & -c & t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ 1 \end{bmatrix} \tag{3}$$

, where c is the cell size used to rasterize the 2-D plane shaped by the extent of the line dataset; v_x and v_y are the (x, y) coordinates of the

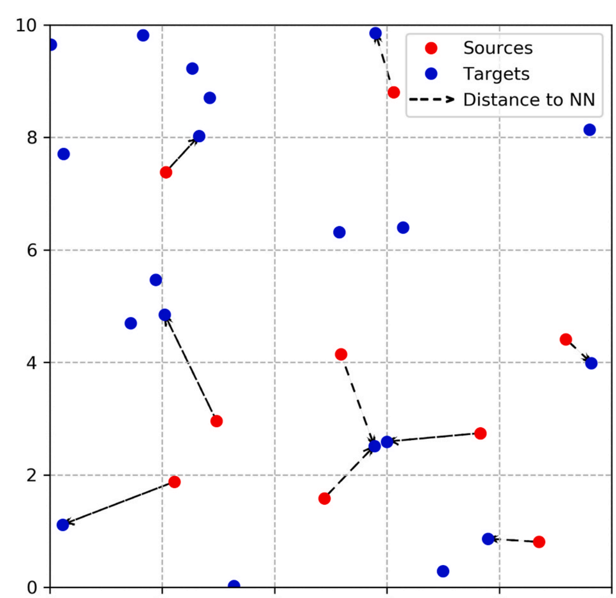

Fig. 1. The Nearest Neighbor, in terms of Euclidean distance, of each source in the target set.

vertices of the line features; l and t are the left and top bound of the 2-D plane; and, finally, r_x and r_y represents the row and column number of the cell located on the transformed raster grid. The affine transformation matrix (ATM), i.e., the augmented matrix in the equation, performs a linear transformation (a scaling and a rotation) followed by a translation (shifting the origin), which preserves the relative spatial relationship among the line features (Wheaton et al., 2012).

After applying an Affine transformation, the problem of calculating distance to line features reverts to a search for NN since lines are pixelated into a definite amount of cells, wherever any part of any line exists. Note that the precision of the calculated distances will depend on c, the cell size of the converted raster grid. Fig. 2 shows an example of the transformation whose ATM's entries are $c = 100$, $l = 5000$, and $t = 4500$.

3.1.3. *Density and Zonal Statistics*

Density is an important instrument in suitability analysis since it directly, from a 2-D perspective, measures the intensity of land-use related activities/phenomena on a landscape. For example, the density of single-family dwelling units in a given region reveals characteristics of the neighborhood, and the density of a road network reflects the level of accessibility it provides to vehicles. Because of its vector-based characteristic, PyLUSAT's density module calculates density within a user-defined set of input zones (polygons). Using an analogy of source and targets in the distance definition, the density of targets in a given source zone i is defined as $\rho_i = (\sum_{j=1}^{m} t_j v_j)/A_i$, where $t_j = 1$, if target j is within the area of i (A_i); otherwise $t_j = 0$, and v_j—the value corresponding to target j—equals to 1, if not specified.

To evaluate spatial containment of point targets, we used the function geopandas.sjoin, which supports three types of topological predicates: *intersect*, *contain*, and *within*. When targets are line features, PyLUSAT will first apply an Affine Transformation to convert them to a raster grid, and then use rasterstats.zonal_stats to calculate the total amount of cells within each source zone. Fig. 3 illustrates the case of calculating line density.

3.1.4. *Interpolation*

Interpolation methods are used to estimate values of unknown data points using the known ones, of which the *Inverse Distance Weighted* (IDW) interpolation is widely applied in GIS-based suitability analysis (Tercan and Dereli, 2020; Varatharajan et al., 2018; Yao et al., 2013). IDW, as a GIS technique, is one of the classic realizations of the so-called *first law of geography*: "everything is related to everything else, but near things are more related than distant things" (Tobler, 1970, p. 236). The most commonly adopted version of IDW is Shepard's (1968) method defined below.

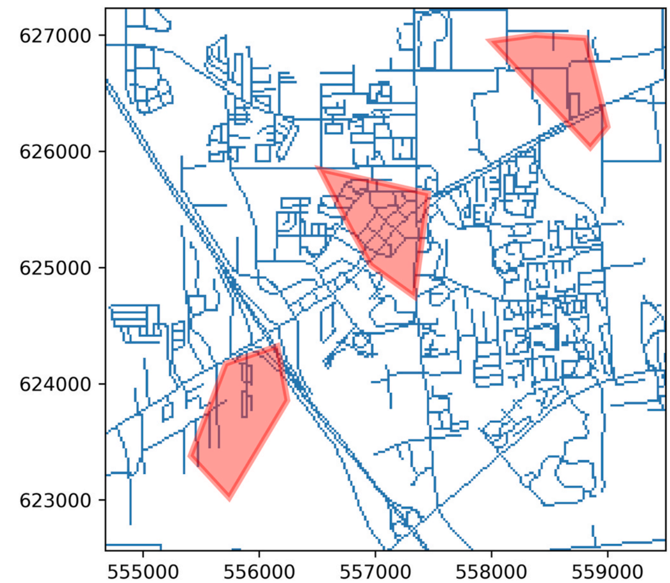

Fig. 3. Density of line features within input zones (after Affine Transformation applied).

$$f(P) = \begin{cases} \dfrac{\sum_{i=1}^{N} (d_i)^{-u} z_i}{\sum_{i=1}^{N} (d_i)^{-u}} & \text{if } d_i \neq 0 \text{ for all } i = 1, 2, \ldots, N. \\ z_i & \text{if } d_i = 0 \text{ for some } i = 1, 2, \ldots, N. \end{cases} \tag{4}$$

, where P is the point of interest, z_i and d_i are, respectively, the value at the i-th known data point and its Euclidean distance to P, and u is a predefined positive number also known as the *power parameter*. Note that, the negative sign before u makes the i-th point's weight inversely proportional to its distance to P in the estimation, hence how the method got its name. As distance increases, the known values' influences on the estimation of P declines faster as u gets bigger, which shifts the algorithm from a global model to a local model. The process of determining the best value of u is relatively deterministic. PyLUSAT provides a function, pylusat.interpolate.idw_cv, which allows users to pick a proper value for u through *Cross Validation*.

3.2. *Transformation function*

The goal of transformation functions is to translate measurements based on various criteria into a standardized "suitability" scale, which is

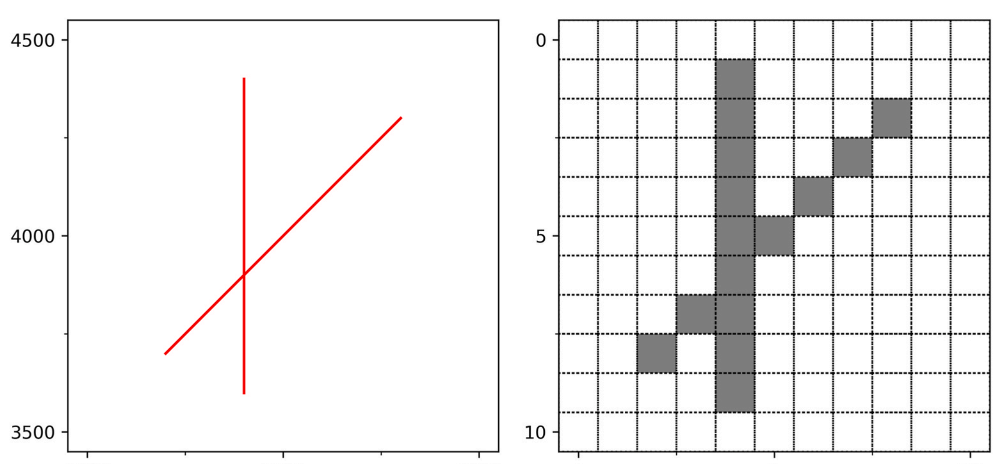

Fig. 2. An Affine Transformation that converts lines features (left) to a raster grid (right).

arbitrarily chosen and then consistently applied throughout the analysis, for example, a scale of 1–9 (the lowest to highest suitability) adopted by the Land Use Conflict Identification Strategy (LUCIS) (Carr and Zwick, 2007). In general, there are three mechanisms to define a transformation, which are by (a) unique categories, (b) range of classes, and (c) continuous functions. The first mechanism typically deals with nominal and ordinal data, in which individual values are associated with different degrees of suitability. The second method handles interval or ratio data and is more flexible, in that values fall into a certain range represent the same degree of suitability. PyLUSAT's rescale.reclassify function allows users to do both transformations by leveraging the pandas.DataFrame object's highly efficient indexing/slicing capability.

Besides pre-defined ranges, ranges can be derived from the data as well. Jenks Jenks (1977) developed the *natural breaks* algorithm, originally as a choropleth mapping technique, which was widely employed in land-use suitability analysis (Abdullahi et al., 2015; Berry and BenDor, 2015; Owusu et al., 2017). Natural breaks seek to simultaneously minimize the differences within classes and maximize the differences between classes. The total variance in the data, also known as the squared deviation from array mean (SDAM), is defined as $\sum_{i=1}^{n}(x_i - \bar{x})^2$, on the other hand, the within-class difference is captured by the squared deviation from the class mean (SDCM), that is $\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_i - \bar{x}_j)^2$, where n_j is the number of elements in the j-th class, and k is a pre-defined number of classes. The algorithm iterates through all possible breaks and computes the Goodness of Variance Fit (GVF), i.e., $\frac{SDAM-SDCM}{SDAM}$. The minimum GVF obtained corresponds to the so-called "optimal" range of classes.

PyLUSAT supports linear transformation, also known as the *min-max feature scaling*, for continuous data transformation. This method is the most common technique to standardize a single criterion and has been used for feature engineering technique in many machine learning applications (Malczewski, 2004; Tang et al., 2018). Besides its intuitiveness, another advantage that makes the method appealing in suitability analysis is that it preserves the distribution of the original variable after the transformation (Cao and Obradovic, 2015). Without loss of generality, the following equation defines a linear transformation of a variable X from its original scale $[x_{min}, x_{max}]$ to an arbitrary scale $[a, b]$.

$$x_i^{'} = \begin{cases} a + \dfrac{x_i - x_{min}}{x_{max} - x_{min}}(b-a) & \text{for } i \in \{1, 2, \ldots n\} \text{ (regular order)}. \\ b - \dfrac{x_i - x_{min}}{x_{max} - x_{min}}(b-a) & \text{for } i \in \{1, 2, \ldots n\} \text{ (inverse order)}. \end{cases} \quad (5)$$

, where n is the total number of observations. Note that, both cases in equation (5) are relevant to measuring land-use suitability. As with the two examples in Section 3.1.1, a relatively large value of the same measurement, i.e., distance, may be valued as either pros or cons, depending on the suitability criteria.

3.3. Aggregation function

In GIS-based suitability analysis, transformed measurements of various criteria are combined to make a land-use decision, which commonly is done by assigning weights to individual criteria, based on expert knowledge, and then summing the results (Kalogirou, 2002). PyLUSAT provides a utility function pylusat.util.weighted_sum for such operation. However, professionals or stakeholders (agents) often find themselves in a situation where a consensus on the weighting cannot be reached. Multi-criteria decision analysis (MCDA) can help resolve the differences. PyLUSAT offers a function, pylusat.utils.ahp, to conduct the Analytic Hierarchy Process (AHP) developed by Saaty (1990).

According to cognitive pyschologist Blumenthal (1977), to make a judgement, either absolute or comparative, "a person must compare an immediate impression with impression in memory of similar stuimuli." This premise that people can deal with only a few facts simultaneously is thoughtfully employed in AHP (Miller, 1956; Saaty, 1990). The MCDA technique converts a decision of multiple criteria into a series of pair-wise comparisons, with the result quantified using a scale from 1 to 9. If item A is equally important to item B, the result is 1. And, if item A is extremely important than item B, the result is 9. The integers in between correspond to different levels of pair-wise importance comparisons. Moreover, the reciprocals of these values are used if one swaps the comparates, i.e., if A to B is 5($w_A/w_B = 5$), then $w_B/w_A = 1/5$. According to this setup, AHP first creates a reciprocal matrix using results of the pair-wise comparisons. Then, it solves the eigenvalue equation, i.e., $A\boldsymbol{v} = \lambda\boldsymbol{v}$, and retains the primary eigenvalue (the largest one among all eigenvalues) and the corresponding primary eigenvector. Finally, it normalize the primary eigenvector (dividing individual elements of the vector by their sum), to obtain the priority vector. Each element of the priority vector represents the weight of an initial criterion involved in the analysis, which reflects its relative importance in the final decision.

AHP also involves a mechanism, *Consistency Ratio* (CR), to validate whether the decisions of the pair-wise comparisons are consistent, e.g., given $w_A/w_B = 7$ and $w_B/w_C = 3$, then if $w_A/w_C = 1/5$, we call it an "inconsistency" in the comparisons. In addition to the pylusat.utils.ahp function, PyLUSAT also provides a pylusat.utils.random_ahp function to generate random AHP weights that follows the rule of thumb, that is CR is less than 0.1.

4. Validation and evaluation

Fig. 4 shows two choropleth maps side-by-side, where the left one presents the result of measuring point distances (using ArcMap) between schools and centroids of census block groups (CBG) of Alachua County, Florida; and the right one shows the same phenomenon with the same color scheme but measured by PyLUSAT. The datasets used to create the two maps, the left by a *layout* of ArcMap and the right by the plotting function of GeoPandas and Contextily (for basemap tiles), are included in the GitHub repository mentioned in the first section. As the figure shows, from a cartographic perspective, the two results are identical.

However, to validate tools in PyLUSAT, especially the geospatial functions, more rigorously, we compared outputs of five *geoprocessing tools* in ArcMap 10.4 with the outputs of corresponding functions in PyLUSAT by conducting a series of (two-tailed) paired-sample t-tests. The null hypotheses of these tests are identical, which is there exists no statistically significant difference between outputs from PyLUSAT functions and their ArcGIS counterparts. Table 1 lists (by function) the degree of freedoms (df), observed t statistics, and p-values of these t tests.

In these t tests, the observations are different quantities measured against the 155 CBGs in Alachua County, hence 154 df. We used school and road network datasets in the county for point and line features respectively which, again, are included in the GitHub repository. For IDW, we used the Digital Elevation Model (DEM) as the value raster grid. As indicated by the p-values, none of the tests can reject the null hypothesis, which suggests that we can trust the results of PyLUSAT's geospatial functions with confidence. PyLUSAT is developed with computing speed in mind as well. In contrast to conducting suitability analysis using GIS applications, computational efficiency is mainly gained from two sources: (a) the implementation of *NumPy's* vectorized operation in PyLUSAT and (b) the I/O wait time saved from reading/writing intermediate files (Harris et al., 2020). The latter is non-negligible in that intermediate outputs are usually saved *on disk* when conducting suitability analysis on desktop GIS software, whereas PyLUSAT keeps the study units (e.g., land parcels) *in memory* as a GeoDataFrame object throughout the entire process of suitability analysis. Fig. 5 shows three time cost (wall time measured in seconds) comparisons between PyLUSAT functions and their counterparts in *ArcMap 10.4*.

Note that, pylusat.density.of_line() calculates line density in each input polygon or in a user-defined radius around each polygon's centroid. Since the function is different from ArcMap's *Line Density* tool, which only produces a raster grid with density values, a ModelBuilder

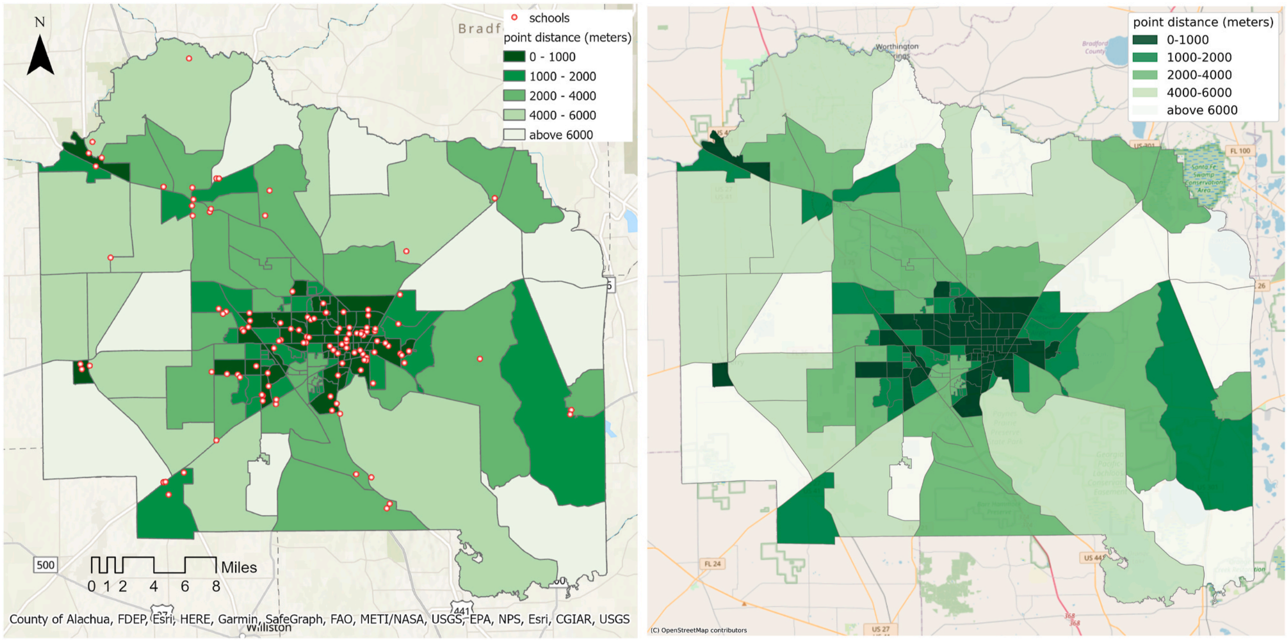

Fig. 4. Maps of measuring point distance. Left: by ArcGIS. Right: by PyLUSAT.

Table 1
Paired-sample *t*-tests between results from PyLUSAT and ArcGIS.

Test function	df	*t* statistic	*p*-value
Distance to point	154	1.052 8	0.294 1
Distance to line	154	0.361 3	0.718 4
IDW	154	0.588 9	0.556 8
Density of point	154	−0.686 3	0.493 6
Density of line	154	1.165 6	0.244 7

model consisting of *Line Density* and *Zonal Statistics* was used in the third comparison. As shown in the figure, PyLUSAT functions take less time to run in all three cases, and such effect become more significant as the total number of measurements increases.

The improved computational efficiency by individual PyLUSAT functions could significantly reduce the total time cost of conducting a suitability analysis. Moreover, this effect can be further amplified by HPC. Since PyLUSAT is cross-platform, it can be installed on a computing cluster (usually running on Linux-based OS) with minimal effort. Thus, it enables urban planners and researchers to rapidly simulate future land-use scenarios to evaluate both the intended and unintended consequences of specific land-use policies under the framework of suitability analysis. This is a main goal of the development of PyLUSAT. Chen Chen (2019) conducted a feasibility study, in which PyLUSAT is used to port the LUCIS model to *HiPerGator*, a supercomputer at the University of Florida (Carr and Zwick, 2007). In this study, ninety-six cores were used to simulate 120 alternative land-use scenarios in Orange County, Florida. The entire process took only slightly over 5 min.

5. Conclusion and future outlook

Open-source software dedicated to GIS-based land use suitability analysis is rarely found in relevant literature. In this paper, we present a Python package—PyLUSAT—representing a promising candidate to help fill in this absence. As an alternative solution to existing GIS applications, PyLUSAT facilitates the customization and automation of suitability analysis while maintaining the process highly scalable and reproducible. The performance of PyLUSAT's functions were evaluated from both accuracy and efficiency perspectives. Five *geospatial functions* in PyLUSAT were selected and conducted paired-sample *t* tests between the outputs from these five functions and outputs from their counterparts in *ArcMap 10.4*. Results of these tests showed that there are no statistically significant difference between the two sets of outputs. Additionally, we benchmarked the time costs (wall time) of three PyLUSAT's geospatial functions and their corresponding tools in ArcMap. Results showed that PyLUSAT functions are noticeably faster.

PyLUSAT has been made available on the Python Package Index (PyPI) and also on GitHub at https://github.com/chjch/pylusat. It offers various tools (functions), allowing the package to handle tasks entailed by GIS-based land use suitability analysis. PyLUSAT can be used not only on a personal computer running on either Windows, Linux, or MacOS, but also on a supercomputer to take advantage of HPC. In addition, PyLUSAT is highly extensible. For example, it has been used to develop a QGIS plugin, *PyLUSATQ*, to support sustainable land management (SLM) in Ghana (Chen et al., 2021). Finally, methods and tools introduced in Section 3 of this paper can be used by developers in the FOSS community who are interested in developing geospatial packages and applications in Python.

As the package is actively being used in applied research projects, continued efforts will be made to maintain the package and respond to feedback and issues reported through the GitHub repository. In addition, we plan to implement a couple of improvements in the forthcoming versions of the software, including 1) a comprehensive list of continuous transforming functions, e.g., gaussian, logistic decay, exponential, etc., 2) further enhancing computational efficiency in PyLUSAT by taking advantage of a just-in-time (JIT) compiler such as *Numba*, and 3) incorporating a parameterized region growth (PRG) function to allow allocation of land uses based on user-defined rules, such as a region's minimum (or maximum) area, minimum average suitability score, and minimum distance between two regions.

Name of software

PyLUSAT.

Developer

Changjie Chen.

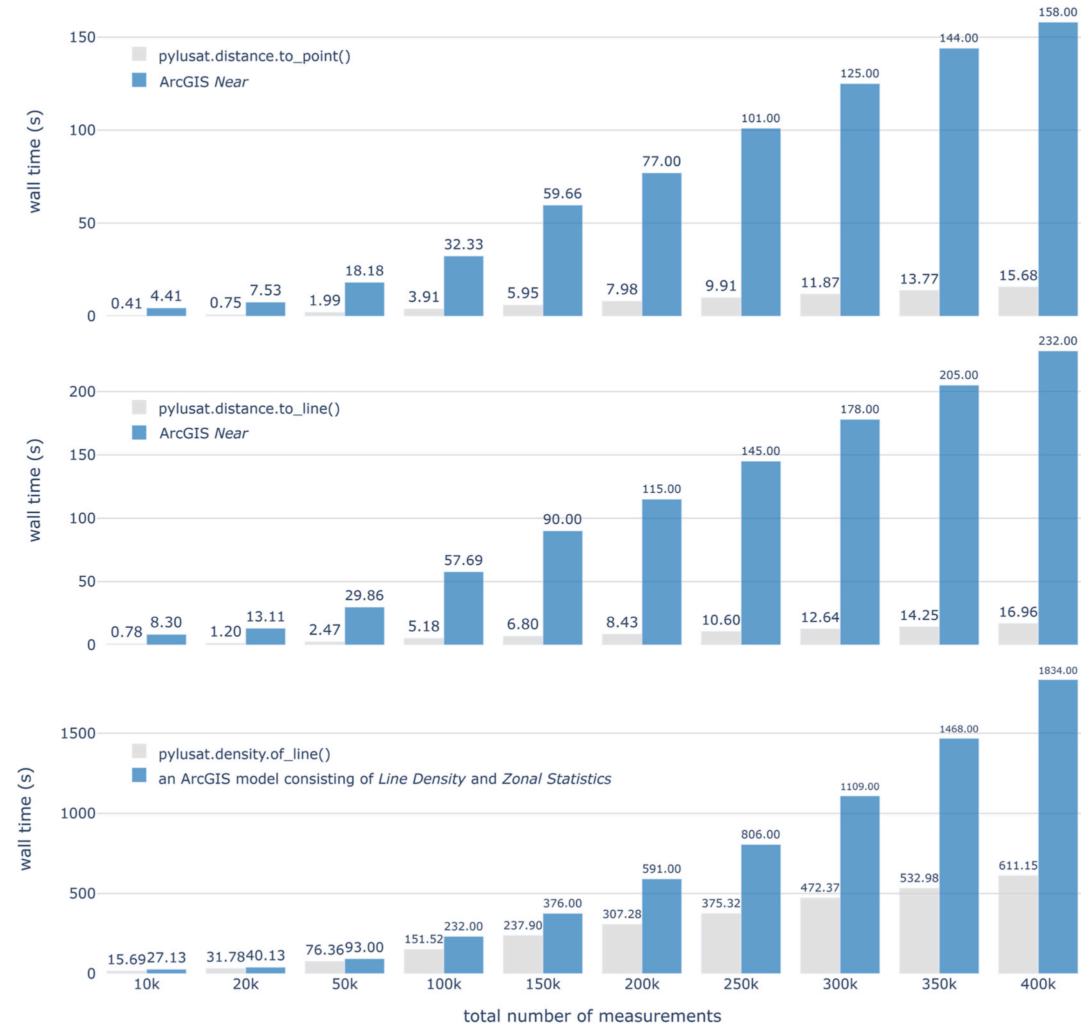

Fig. 5. Running times for PyLUSAT functions compared with their ArcGIS counterparts.

Contact information

chj.chen@ufl.edu.

Software license

BSD.

Year first available

2020.

Program language

Python.

Cost

Free.

Software availability

https://pypi.org/project/pylusat/

Code repository

https://github.com/chjch/pylusat.

Program size

4.2 MB.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank the anonymous reviewers for their time to help improve the manuscript and Professor Paul Zwick from the Department of Urban and Regional Planning at the University of Florida for his valuable inputs at the early stage of PyLUSAT development. In addition, we acknowledge funding from the Florida Institute for Built Environment Resilience (FIBER) and NASA/USAID-SERVIR Program (Award # 80NSSC20K0153) to support this research.

References

Abdullahi, Saleh, et al., 2015. GIS-based modeling for the spatial measurement and evaluation of mixed land use development for a compact city. GI- Sci. Remote Sens. 52 (1), 18–39. https://doi.org/10.1080/15481603.2014.993854.

Barreira-González, Pablo, Gómez-Delgado, Montserrat, Aguilera- Benavente, Francisco, 2015. From raster to vector cellular automata models: a new approach to simulate urban growth with the help of graph theory. Comp. Environ. Urban Sys. 54, 119–131. https://doi.org/10.1016/j.compenvurbsys.2015.07.004.

Berry, Marisa, BenDor, Todd K., 2015. Integrating sea level rise into development suitability analysis. Comp. Environ. Urban Sys. 51, 13–24. https://doi.org/10.1016/j.compenvurbsys.2014.12.004.

Bian, Ling, 2007. Object-oriented representation of environmental phenomena: is everything best represented as an object? Ann. Associat. Am. Geographers 97 (2), 267–281. https://doi.org/10.1111/j.1467-8306.2007.00535.x.

Bivand, Roger S., Pebesma, Edzer, Gómez-Rubio, Virgilio, 2013. Applied Spatial Data Analysis with R, 2nd. Springer, New York, pp. 1–405. https://doi.org/10.1007/978-1-4614-7618-4.

Blanchard, Samuel D., PaulWaddell, 2017. UrbanAccess: generalized methodology for measuring regional accessibility with an integrated pedestrian and transit network. Transport. Res. Record 2653, 35–44. https://doi.org/10.3141/2653-05.

Blumenthal, Arthur L., 1977. The process of cognition. Prentice Hall/Pearson Education. Boeing, Geoff (2017). "OSMnx: new methods for acquiring, constructing, analyzing, and visualizing complex street networks". Comp. Environ. Urban Sys. 65, 126–139. https://doi.org/10.1016/j.compenvurbsys.2017.05.004.

Bolleter, Julian, et al., 2021. Informing future Australian settlement planning through a national-scale suitability analysis. Internation Plann. Studies 27 (1), 18–43. https://doi.org/10.1080/13563475.2021.1899903.

Bolte, John P., et al., 2007. Modeling biocomplexity - actors, landscapes and alternative futures. Environ. Model. Software 22 (5), 570–579. https://doi.org/10.1016/j.envsoft.2005.12.033.

Cao, Kai, et al., 2012. Sustainable land use optimization using boundary-based fast genetic algorithm. Comp. Environ. Urban Sys. 36 (3), 257–269. https://doi.org/10.1016/J.COMPENVURBSYS.2011.08.001.

Cao, Xi Hang, Obradovic, Zoran, 2015. A robust data scaling algorithm for gene expression classification. In: 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE). IEEE, pp. 1–4. https://doi.org/10.1109/BIBE.2015.7367734.

Carr, Margaret H., Paul, D Zwick, 2007. Smart Land-Use Analysis: the LUCIS Model Land-Use Conflict Identification Strategy. Esri Press, Redlands, CA.

Carreira, Pablo, 2016. Geospatial Development by Example with Python. Packt, Birmingham, UK, p. 340.

Chang, Aileen Y., et al., 2009. Combining google earth and GIS mapping technologies in a dengue surveillance system for developing countries. Int. J. Health Geographics 8 (1), 1–11. https://doi.org/10.1186/1476-072X-8-49/FIGURES/7.

Chen, Changjie, 2019. Massive Simulation and Multi-Scale Analysis of Land- Use Suitability Based on Parallel Computing. PhD thesis. University of Florida, Gainesville, FL.

Chen, Changjie, et al., 2021. Building Capacity for Decision Makers in Ghana for Sustainable Land Use Planning Using Earth Observations and Open-Source GIS Tools. https://doi.org/10.1002/ESSOAR.10509617.1.

Collins, Michael G., Steiner, Frederick R., Rushman, MIchael J., 2001. Landuse suitability analysis in the United States: historical development and promising technological achievements. Environ. Manag. 28 (5), 611–621. https://doi.org/10.1007/s002670010247.

Coppock, J Terry, Rhind, David W., 1991. The history of GIS. In: Maguire, David J., Goodchild, Michael F., Rhind, David W. (Eds.), Geographical Information Systems: Principles and Applications. Longman Scientific and Technical, London, UK, pp. 21–43. London, UK.

Couclelis, Helen, 1992. People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In: Frank, Andrew U., Campari, Irene, Formentini, Ubaldo (Eds.), From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning on Theories and Methods of Spatio- Temporal Reasoning in Geographic Space. Springer, London, UK, pp. 65–77. https://doi.org/10.1007/3-540-55966-3_3.

Eastman, J Ronald, 1997. IDRISI for Windows: Advanced Student Manual: Special Topics. Clark University, Clark Labs, version 2.0.

Egenhofer, Max J., Frank, Andrew U., 1987. Object-oriented databases: database requirements for GIS. In: International Geogrpahic Information Systems (IGIS) Symposium: the Research Agenda, 2, pp. 189–211. Arlington, VA.

Esri, 2015. Independent Report Highlights Esri as Leader in Global GIS Market.

Gillies, Sean, Others, 2007. Shapely: manipulation and analysis of geometric objects. url: https://github.com/Toblerity/Shapely.

Gillies, Sean, Others, 2011. Fiona is OGR's neat, nimble, no-nonsense API. url: https://github.com/Toblerity/Fiona.

Gillies, Sean, Others, 2013. Rasterio: Geospatial Raster I/O for Python programmers url: https://github.com/mapbox/rasterio.

Girmay, Gebrehana, et al., 2018. Land capability classification and suitability assessment for selected crops in Gateno watershed, Ethiopia. Cogent Food Agric. 4 (1), 1532863. https://doi.org/10.1080/23311932.2018.1532863.

Goodchild, Michael F., 1989. Modeling error in vectors and fields. In: Goodchild, Michael F., Gopal, Sucharita (Eds.), The Accuracy of Spatial databases. Taylor & Francis, London, UK, pp. 107–113. Chap. 10.

Goodchild, Michael F., Yuan, May, Cova, Thomas J., 2007. Towards a general theory of geographic representation in GIS. Int. J. Geographical Inform.Sci. 21 (3), 239–260. https://doi.org/10.1080/13658810600965271.

Harris, Charles R., et al., 2020. Array programming with NumPy. Nat. 585 (7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2.

Hopkins, Lewis D., 1977. Methods for generating land suitability maps: a comparative evaluation. J. Am. Institute Planners 43 (4), 386–400. https://doi.org/10.1080/01944367708977903.

Ingmire, Thomas J., Patri, Tito, 1971. An early warning system for regional planning. J. Am. Institute Planners 37 (6), 403–410. https://doi.org/10.1080/01944367108977390.

Jelinski, Dennis E., Wu, Jianguo, 1996. The modifiable areal unit problem and implications for landscape ecology. Landscape Ecology 11 (3), 129–140. https://doi.org/10.1007/BF02447512.

Jenks, George, 1977. Optimal Data Classification for Choropleth Maps. University of Kansas, Lawrence Kan.

Jiang, Hong, Ronald Eastman, J., 2010. Application of fuzzy measures in multi-criteria evaluation in GIS. Int.J. Geograph. Inform. Sci. 14 (2), 173–184. https://doi.org/10.1080/136588100240903.

Jones, Grant, R., 2007. Grant Jones/Jones & Jones: ILARIS: the Puget Sound Plan, p. 144.

Jordahl, Kelsey, et al., 2019. Geopandas/Geopandas, 6.3. https://doi.org/10.5281/ZENODO.3946761.

Kalogirou, S., 2002. Expert systems and GIS: an application of land suitability evaluation. Comp. Environ. Urban Sys. 26 (2–3), 89–112. https://doi.org/10.1016/S0198-9715(01)00031-X.

Liu, Y., et al., 2008. Towards a General Field model and its order in GIS. Int. J. Geograph. Inform. Sci. 22 (6), 623–643. https://doi.org/10.1080/13658810701587727.

Lyle, John, Stutz, Frederick P., 1983. Computerised land use suitability mapping. Cartographic J. 20 (1), 39–49. https://doi.org/10.1179/CAJ.1983.20.1.39.

Ma, Xuying, et al., 2020. PyLUR: efficient software for land use regression modeling the spatial distribution of air pollutants using GDAL/OGR library in Python. Front. Environ. Sci. Eng. 14 (3), 1–14. https://doi.org/10.1007/S11783-020-1221-5.

Malczewski, Jacek, 2004. GIS-based land-use suitability analysis: a critical overview. Progress Planning 62 (1), 3–65. https://doi.org/10.1016/j.progress.2003.09.002.

Maneewongvatana, Songrit, Mount, David M., 1999. Analysis of approximate nearest neighbor searching with clustered point sets. In: Goldwasser, Michael H., Johnson, David S. (Eds.), Catherine C. McGeoch., Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges, Proceedings of a DIMACSWorkshop, USA, 1999, 59 DIMACS Series in Discrete Mathematics and Theoretical Computer Science. DIMACS/AMS, pp. 105–123. https://doi.org/10.1090/dimacs/059/06.

Marull, Joan, et al., 2007. A land suitability index for strategic environmental assessment in metropolitan areas. Landscape Urban Plann. 81 (3), 200–212. https://doi.org/10.1016/j.landurbplan.2006.11.005.

McDowell, R.W., et al., 2018. The land use suitability concept: introduction and an application of the concept to inform sustainable productivity within environmental constraints. Ecological Indicators 91, 212–219. https://doi.org/10.1016/J.ECOLIND.2018.03.067.

McHarg, Ian L., 1969. Design with Nature. American Museum of Natural History, Garden City N.Y., p. 197

McKinney, Wes, 2010. Data Structures for Statistical Computing in Python, pp. 51–56.

Mesgaran, Mohsen B., et al., 2017. Iran's land suitability for agriculture. Sci. Rep. 7 (1), 1–12. https://doi.org/10.1038/s41598-017-08066-y.

Miller, George A., 1956. The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol. Rev. 63 (2), 81–97. https://doi.org/10.1037/H0043158.

Muenchen, Bob, 2017. Data Science Job Report 2017: R Passes SAS, but Python Leaves Them Both behind.

Neteler, M., et al., 2012. Grass GIS: a multi-purpose open source GIS. Environ. Model. Software 31, 124–130. https://doi.org/10.1016/j.envsoft.2011.11.014.

Ottaviano, Gianmarco, Thisse, Jacques François, 2004. Agglomeration and economic geography. In: Vernon Henderson, J. (Ed.), Handbook of Regional and Urban Economics, 4. Elsevier, pp. 2563–2608. https://doi.org/10.1016/S1574-0080(04)80015-4. Jacques-François Thisse.

Owusu, Seth, et al., 2017. Assessing land suitability for aquifer storage and recharge in northern Ghana using remote sensing and GIS multi-criteria decision analysis technique. Model. Earth Sys. Environ. 3 (4), 1383–1393. https://doi.org/10.1007/s40808-017-0360-6.

Pebesma, Edzer, 2018. Simple features for R: standardized support for spatial vector data. R J. 10 (1), 439–446. https://doi.org/10.32614/RJ-2018-009.

R Core Team, 2021. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Saaty, Thomas L., 1990. How to make a decision: the analytic hierarchy process. European J. Operation. Res. 48 (1), 9–26, 10.1016/0377-2217(90)90057-I.

Scopesi, C., et al., 2020. Land capability classification of Vernazza catchment, Cinque Terre national Park, Italy. J. Maps 16 (2), 357–362. https://doi.org/10.1080/17445647.2020.1750067.

Seyedmohammadi, Javad, et al., 2019. Development of a model using matter element, AHP and GIS techniques to assess the suitability of land for agriculture. Geoderma 352, 80–95. https://doi.org/10.1016/J.GEODERMA.2019.05.046.

Shepard, Donald, 1968. A two-dimensional interpolation function for irregularlyspaced data. In: Proceedings of the 1968 23rd ACM National Conference. Association for Computing Machinery, New York, NY, pp. 517–524. https://doi.org/10.1145/800186.810616.

Steiner, Frederick, McSherry, Laurel, Cohen, Jill, 2000. Land suitability analysis for the upper Gila River watershed. Landscape Urban Plann. 50 (4), 199–214. https://doi.org/10.1016/S0169-2046(00)00093-1.

Steiniger, Stefan, Hunter, Andrew J.S., 2013. The 2012 free and open source GIS software map – a guide to facilitate research, development, and adoption. Comp. Environ. Urban Sys. 39, 136–150. https://doi.org/10.1016/J.COMPENVURBSYS.2012.10.003.
Steinitz, Carl, 1976. Hand-drawn overlays: their history and prospective uses. Landscape architecture 66, 444–455.
Strobl, Christian, 2008. Encyclopedia of GIS. In: Shekhar, Shashi, Xiong, Hui (Eds.), Encyclopedia of GIS. Springer US, Boston, MA, pp. 240–245. https://doi.org/10.1007/978-0-387-35973-1.
Strohmaier, Erich, et al., 2020. Operating System Family/Linux. https://www.top500.org/statistics/details/osfam/1/.
Tang, Jingyin, Matyas, Corene J., 2018. Arc4nix: a cross-platform geospatial analytical library for cluster and cloud computing. Comput. Geosci. 111, 159–166. https://doi.org/10.1016/J.CAGEO.2017.11.011.
Tang, Xiaotang, Tan, Sheldon X.-D., Chen, Hai-Bao, 2018. SVM based intrusion detection using nonlinear scaling scheme. In: 2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT). IEEE, pp. 1–4. https://doi.org/10.1109/ICSICT.2018.8565736.
Tercan, Emre, Ali Dereli, Mehmet, 2020. Development of a land suitability model for citrus cultivation using GIS and multi-criteria assessment techniques in Antalya province of Turkey. Ecol. Indicators 117, 106549. https://doi.org/10.1016/j.ecolind.2020.106549.
Thill, Jean-Claude, 1999. Spatial Multicriteria Decision Making and Analysis. In: MAKING-ANALYSIS-JEAN-CLAUDE-THILL. Routledge, p. 393. https://doi.org/10.4324/9780429436628/SPATIAL-MULTICRITERIADECISION-.
Tobler, Waldo R., 1970. A computer movie simulating urban growth in the Detroit region. Econ. Geograp. 46, 234–240. https://doi.org/10.2307/143141.
Tomlin, C. Dana, 1990. Geographic Information Systems and Cartographic Modeling. Prentice Hall, Englewood Cliffs N.J., p. 249
Varatharajan, R., et al., 2018. Visual analysis of geospatial habitat suitability model based on inverse distance weighting with paired comparison analysis. Multimedia Tools Appl. 77 (14), 17573–17593. https://doi.org/10.1007/s11042-017-4768-9.
Virtanen, Pauli, et al., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2.
Waddell, Paul, 2010. Modeling residential location in UrbanSim. In: Pagliara, Francesca, Preston, John (Eds.), Residential Location Choice: Models and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 165–180. https://doi.org/10.1007/978-3-642-12788-5_8. David Simmonds.
Wheaton, Joseph M., et al., 2012. A simple, interactive GIS tool for transforming assumed total station surveys to real world coordinates - the CHaMP transformation tool. Comput. Geosci. 42, 28–36. https://doi.org/10.1016/j.cageo.2012.02.003.
Winter, Stephan, Frank, Andrew U., 2000. Topology in raster and vector representation. GeoInformatica 4.1, 35–65. https://doi.org/10.1023/A:1009828425380.
Wu, Hong, et al., 2015. A scenario-based approach to integrating flow-ecology research with watershed development planning. Landscape Urban Planning 144, 74–89. https://doi.org/10.1016/j.landurbplan.2015.08.012.
Xiao, Bo, George, Biros, 2016. Parallel algorithms for nearest neighbor search problems in high dimensions. SIAM J. Sci. Comput. 38 (5), S667–S699. https://doi.org/10.1137/15M1026377.
Xiao, Ningchuan, Bennett, David A., Armstrong, Marc P., 2002. Using evolutionary algorithms to generate alternatives for multiobjective site-search problems. Environ. Planning A 34 (4), 639–656. https://doi.org/10.1068/A34109.
Yang, Tianren, et al., 2019. Understanding urban sub-centers with heterogeneity in agglomeration economies—-Where do emerging commercial establishments locate? Cities 86, 25–36. https://doi.org/10.1016/j.cities.2018.12.015.
Yao, X., et al., 2013. Comparison of four spatial interpolation methods for estimating soil moisture in a complex terrain catchment. PLoS ONE 8 (1), 54660. https://doi.org/10.1371/journal.pone.0054660.
Yohannes, Hamere, Soromessa, Teshome, 2018. Land suitability assessment for major crops by using GIS-based multi-criteria approach in Andit Tid watershed, Ethiopia. Cogent Food Agricult. 4 (1), 1470481, 10.1080/23311932.2018.1470481.