Application of Machine Learning and Artificial Intelligence for Automatic Layout Generation for Smart Cities

Snehee Chheda
B.Tech Data Science
MPSTME, NMIMS
Mumbai, India
ID: 0000 0002 4183 0372

Anish Gharat
B.Tech Data Science
MPSTME, NMIMS
Mumbai, India
ID: 0000 0001 9496 5505

Dr. Siba Panda
Depatment of Data Science
MPSTME, NMIMS
Mumbai, India
Siba.Panda@nmims.edu

***Abstract*— Over the years, cities have developed significantly and major urbanisation has taken place. New roads, highways, and transportation routes are being worked on everyday all over the world. With the increase of population and human needs, quality of life has become very important.**

An area is considered to be a smart city when the city's operations are not only smooth but also extremely efficient, leading to a significant increase of ease in the people's lives.

Our research aims to use machine learning and basic artificial intelligence concepts to enable people, especially policymakers and organisations, to generate unique, quick and efficient layouts for a smart city. Python libraries such as NumPy, SciPy, Pandas, and Matplotlib were used to develop a tool to optimise urban planning by creating layouts consisting of various zones. Constraint based algorithm concepts along with basic artificial intelligence concepts such as optimization, distance and heuristics and random sampling are used.

The generator considers 16 zones, each individually colour coded, while also keeping in mind transportation networks, environment and amenities. The solution offers a dynamic and customisable tool for scenario management and future development planning.

***Keywords*: *Smart City Infrastructure, Urban Planning, Zoning Regulations, Sustainable Development, Artificial Intelligence*.**

I. INTRODUCTION

As of now, India itself has around 100 smart cities. The Government of India launched the Smart Cities Mission on 25 June 2015 with the objective to promote sustainable and inclusive cities that provide core infrastructure and give a decent quality of life to its citizens, a clean and sustainable environment and application of 'Smart' Solutions [1].

Our code generates a simulated grid by randomly placing various zones according to predefined percentages and constraints using Euclidean distance. In Euclidean space, the Euclidean distance is the length of a segment of a straight line that connects two points [2]. It then visualises the entire layout. The random nature of actual cities is considered while creating the algorithm. There are 16 zones covering all aspects i.e residential, healthcare, educational, commercial and industrial. Natural features such as parks, forests and water bodies have also been considered. A basic structure for roads has been depicted. Other than this, airports, highways and railway lines are added for travelling connectivity. Further zones have been added to account for fixed spaces and a few blocks can be kept empty, for future development. A lot of constraints have been considered such as dense green areas are placed next to industrial areas to reduce their carbon footprint. Industrial areas have been visualised keeping renewable energy in mind. Waste management and energy consumption is supervised. Industrial areas have been placed as far as possible from the city centre. Residential zones are placed in clusters just like actual cities. To avoid mishaps, buildings close to the airports must have a lesser number of levels. A highway runs throughout the city, connecting the city to the city centre. Basic connectivity is also shown between the closed industries to commercial zones and residential zones to hospitals. Additionally, fixed zones can be added for the coordinates that the user wants undisturbed. Current work done with respect to smart cities focuses on using data to make sustainable changes to current cities for betterment. Data is gathered for studies using multiple sensors, Geographic Information Systems (GIS), etc. A gap was seen in terms of making a layout in an already existing city and hence, this research was undertaken.

II. LITERATURE REVIEW

The paper by Ojo et al. (n.d.) proposes the Smart City Initiative Design (SCID) framework model for smart city development purposes. More than the layout, it discusses socio-economic factors such as governance, technology, civic engagement etc. to create a good urban environment. It emphasises on the importance of data driven decision making, while highlighting the need for scalability, adaptability, and resilience in smart city designs to meet evolving urban challenges. The approach deals with issues at multiple levels instead of just a single one [3]. The article by Gao et al. (2023) discusses the concept of sustainable smart cities, which utilise information technology to boost urban development. While exploring the challenges faced by modern cities, the authors aim to introduce a sustainable solution for urban development. Some of the technical features of the article include Internet of Things (IoT), City

Information Modelling (CIM), and GIS [4]. The article by Sharifi et al. (2021) discusses bibliometric analysis and the usage of science mapping techniques to identify the thematic focus of over 5000 articles indexed in the Web of Science since the year of 1991. In this research, it was concluded that present research is dominated by either conceptual issues or underlying technical aspects and future research must be done on the actual implementation of the smart city [5].

It is found that even if smart city technologies increase civic engagement and efficiency, there are still issues with matching sensor data to actual conditions. Monitoring public space usage, maximising urban shape, and pinpointing crime-ridden regions are some of the key applications. The primary problem lies in validating data against real-world environmental conditions and user behaviour [6]. The use of smart maps shows another level of innovation within the urban planning details. The smart maps integrate real-time data within a GIS framework, and this time comes up with some dynamic visualisations over urban areas. In this respect, the map could be used to monitor traffic flows, environmental conditions, and other critical urban metrics. In a way, the report had identified how smart maps can make urban planning processes a bit more responsive and adaptive [7]. The use of AI, atmospheric science and social science to improve the air quality in a city has been studied. With the special data available traffic flows, construction of new routes and population dynamics is optimized, ultimately with the goal to optimize sustainable urban planning solutions [8].

III. Methodology

The proposed methodology is a Python-based urban planning simulation tool designed to optimize city layouts by assigning various land use zones while ensuring compliance with zoning regulations and accessibility standards. Fig. 1 depicts the flowchart of the entire algorithm of our solution. The algorithm was designed keeping in mind, that a layout is generated following certain norms but also adding a variable of randomness giving a unique layout every time. The code generates a simulated grid of areas specified by the user by randomly placing various zones like residential, industries, healthcare, etc. while ensuring spatial constraints. This grid is later visualized for a better understanding. The user also has an option of generating a default grid or a customized one. In a customized grid you can block or reserve spots on the grid if needed before generating the layout. This feature is useful in cases where there are pre-existing structures or features present in the area. These pre-existing features could be topographical features or monuments as well.

A. Steps of proposed algorithm

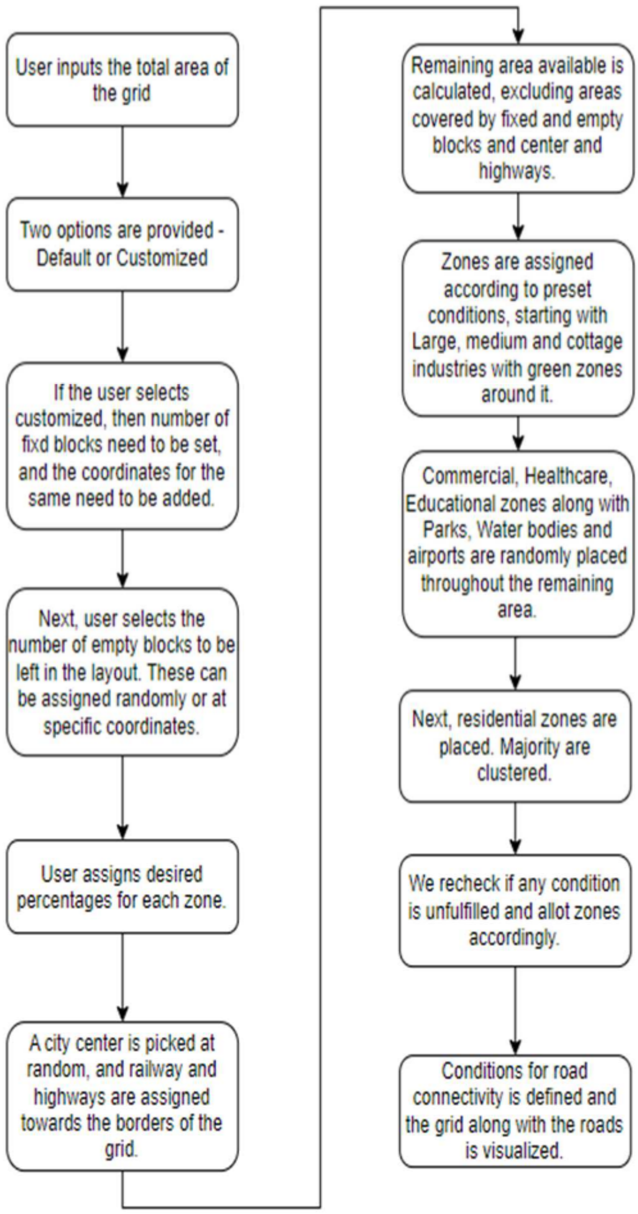


Fig. 1. Flowchart of Proposed Algorithm

After the entire grid is made, accuracy score is calculated by comparing the actual allocations and the desired percentages for each area/zone. This gives us an estimate of how rigidly can the current constraints be followed. The number of fixed and empty blocks and the area covered by each zone affect the accuracy score in this case.

B. *Pseudocode of the solution*

```
START

  // Step 1: Initialize Grid and Input
    INPUT total_area  // Total area of the city
    CALCULATE grid_size = sqrt(total_area)  // Assuming a square grid
    INITIALIZE grid[grid_size][grid_size]  // Create empty grid

  // Step 2: User Chooses Default or Customized Grid
    DISPLAY "Do you want a default grid or a customized grid?"
    INPUT grid_type  // Either "default" or "customized"

    IF grid_type == "default" THEN
      // Default grid, no fixed or empty blocks
      CONTINUE with zone allocation

    ELSE IF grid_type == "customized" THEN
      // Customized grid, assign fixed and empty blocks

      INPUT fixed_blocks[]  // User defines fixed zones (like parks, buildings, etc.)
      INPUT empty_blocks[]  // User defines empty zones (like undeveloped land)

  // Step 3: Set Fixed and Empty Blocks
    FOR each block in fixed_blocks
      SET grid[block] = "Fixed"
    END FOR

    FOR each block in empty_blocks
        SET grid[block] = "Empty"
      END FOR
    END IF
  // Step 4: Allocate Zones Based on Percentages
    INPUT zone_percentages[]  // User inputs percentage for each zone type
    VALIDATE that zone_percentages sum to 100

  FOR each zone_type in zone_percentages
      CALCULATE blocks_needed for zone_type based on percentage
      WHILE blocks_allocated < blocks_needed
        FIND an empty block in the grid
        ASSIGN zone_type to the block based on its nature
        IF zone_type is "Large Industry"
          PLACE block far from the center
        END IF
      END WHILE
    END FOR

  // Step 5: Validation and Connectivity
    FOR each residential block
      FIND nearest healthcare block
      CALCULATE distance between them and ensure connectivity
    END FOR

  // Step 6: Visualization
    DISPLAY the final grid with color-coded zones
    PRINT summary of blocks allocated for each zone type

END
```

Fig. 2. Pseudocode of the solution

Fig. 2 depicts the internal working of the solution in even more detail. The pseudocode talks about the order in which zones are placed and what conditions or restraints are considered before planning those zones. From this it can also be understood that goal of the solution is not only placing the zones but also validating its placement after all zones are fixed. Prevention of overlapping of zones is also checked, proving to be a dependable solution. The algorithm plans the layout to the smallest details, which includes factors such as:

1. Assigning heavy industries away from the city center and placing green zones around them to reduce the carbon footprint.
2. Residential zones are placed in clusters, making it similar to naturally built residential areas present today.
3. Close allocation of zones such as educational, commercial, parks and recreational zones to residential zones is also considered.
4. Airport blocks are to be placed close to each other as they generally need a larger area.
5. Basic road connectivity between important zones, such as residential to hosptials and clinics and industries to commerical zones is also shown.

C. *Inputs*

Table 1. Percentage allocation to each zone

Number	Zone	Percentage Allotted
1	Dense Green	6%
2	Parks & Recreation	4%
3	Residential	33%
4	Large Industries	7%
5	Medium Industries	7%
6	Cottage Industries	6%
7	Large Hospitals	6%
8	Clinics	7%
9	Commercial	11%
10	Airports	2%
11	Educational	7%
12	Water Bodies	4%

The distribution of the zones can be personalized according to the users need. This distribution is taken in terms of percentage of area of be covered by the zone. The results discussed further in the paper use the values mentioned in Table. 1, which shows the 12 zones whose areas can be customized along with the area allocated to each zone. Total area has been taken as 1100.

IV. Results

While implementing the solution the first step is defining the area of the grid where the zones are to be allotted. Fixing or reserving the zones is the next step if needed can be done using the custom layout option. Percentages for each zone are taken as an input from the user and the number of cells to be allocated per zone is calculated based on it. Zones are allocated according to its priority while considering its constraints. A final grid is generated and a matrix representing the zones is created. Each number in the matrix in Fig. 3 denotes a unique zone, this matrix is visualized for a better understanding.

```
[[ 9.   3.   9. 11.   3.   3.   1. 15.   3.   3.   3.]
 [12.   3. 13.   7.   4.   1.   9. 15. 14.   3.   5.]
 [ 7.   3. 14.   3.   5.   3. 10. 15.   3. 14.   1.]
 [15. 15. 15. 15. 15. 15. 15. 16. 15. 15. 15.]
 [ 2.   3.   1.   3. 11.   7.   4. 15.   9. 12.   9.]
 [ 9.   8.   8.   3.   5. 11.   3. 15.   3.   7.   3.]
 [ 3.   3.   9.   3.   8.   3. 10. 15.   3.   3. 12.]
 [12. 11.   7.   2. 11.   3.   3. 15.   2.   3. 11.]
 [ 8.   3.   3.   6.   4. 11.   5. 15.   9.   3.   8.]
 [ 1.   1.   5.   6.   4.   8.   3. 15.   3.   4.   2.]
 [ 4.   6.   5.   4.   4.   9.   8. 15.   7.   9.   9.]]
```

Fig. 3. Matrix showing how the zones are allocated in the code

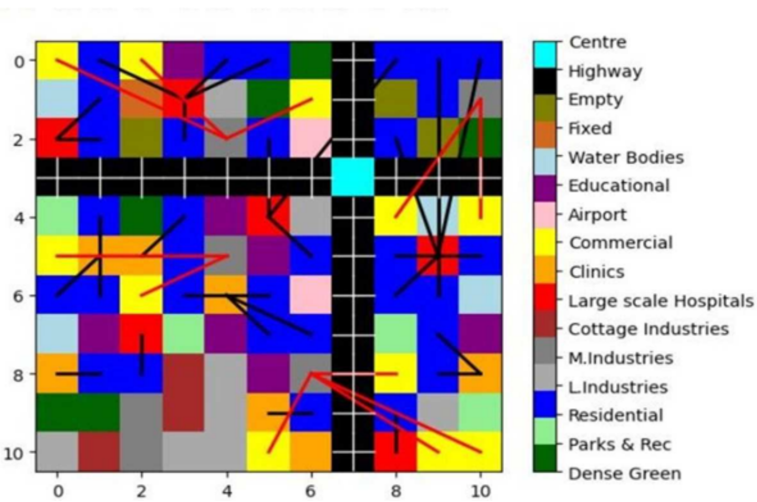


Fig. 4. Layout of the city visualised

Fig. 4 show the layout created from the matrix and visualized on a 2-D plane, which can be generated multiple times, or be modified according to the requirements. A map can be drawn from this 2-dimensional map and software could be used to create the 3-dimensional model for the same [9]. This layout satisfies the constraints with an accuracy of 97%.

Fig. 5. Layout of the city visualised as a 3-D model

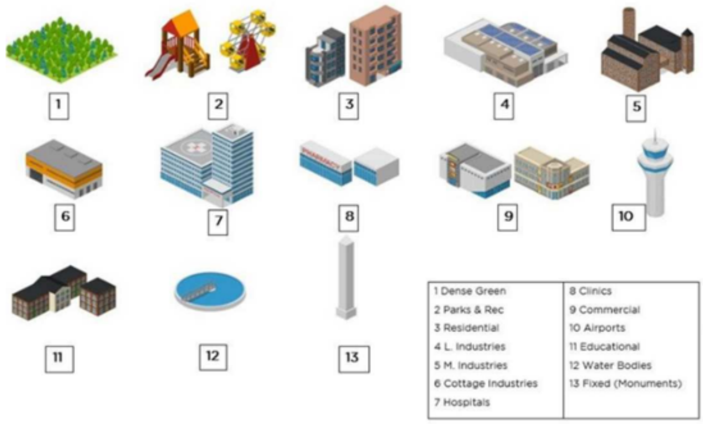


Fig. 6. Legend for the 3D city visualised above

Fig. 5 is 3-D representation of the city generated and shown in Fig. 4, giving us a better idea of the visual aspect of the city. Fig. 6 is the legend depicting the zones and the icons used for the same.

Other than this the highways have separate lanes for public transport and other modes of transport such as underground metro systems for better connectivity. Renewable sources of energy can be used for services such as street lighting and surveillance cameras.

The constraints for generating basic road connectivity can be changed as needed by the user. This solution all in all improves the current manual method of designing smart cities. The constraints can be improved or modified based on the need of the user, the use of empty and fixed blocks further enhances the chances of customizing the layout. Even if not 100% desirable, multiple layouts can be generated. This process is simple, cost effective and way faster than the current methods.

The advantage of the algorithm is that the same code can be re-run to get a new layout each time. If a user wants a change or is not satisfied with the current output in terms of accuracy, the code can be re-run to generate a fresh, more accurate result as well. Keeping the constraints, the algorithm was executed again to generate the output as Fig. 7 and Fig. 8.

```
[[ 4.  8. 10.  9. 15.  3.  3.  6.  3.  8.  7.]
 [ 4.  3.  3. 11. 15.  3.  3.  3. 12.  9.  8.]
 [ 3.  9.  3. 11. 15. 11.  3.  9.  5.  2.  8.]
 [15. 15. 15. 15. 16. 15. 15. 15. 15. 15. 15.]
 [ 6.  9.  3.  9. 15.  8.  2.  3.  3.  3.  4.]
 [ 3.  9.  3. 12. 15.  5.  5.  2.  3.  7.  3.]
 [ 9.  5.  3.  3. 15.  3.  6.  3.  7.  9. 12.]
 [10.  7.  3.  8. 15. 11.  3.  9.  5.  3.  3.]
 [11.  3.  8.  7. 15.  1.  2.  6.  4.  5.  5.]
 [12. 11.  3. 11. 15.  4.  4.  1.  9.  1.  3.]
 [ 3.  7.  3.  3. 15.  4.  6.  1.  1.  1.  4.]]
```

Fig. 7. Matrix of the city visualised by executing the code again

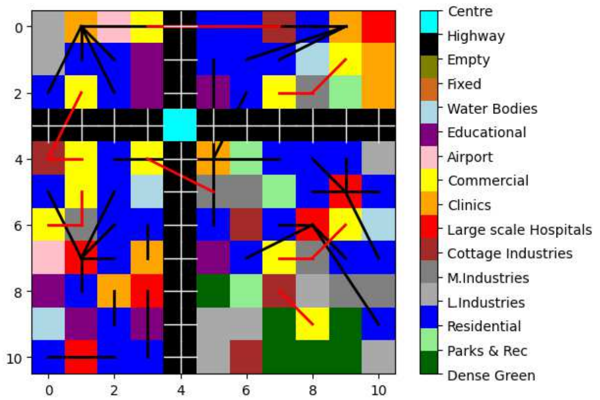


Fig. 8. Layout of the city visualised by executing the code again

The algorithm, for this re-run returned an accuracy of 98% in terms of satsfying contraints.

Future changes need to be done along the lines of layout generation for smart cities keeping topography, population, shape and needs of the city in mind. The use cases of the algorithm are not limited to smart cities and can be used to plan transport lines or construction zones in existing cities as well. It can be used while transforming rural areas to cities and smart cities as well.

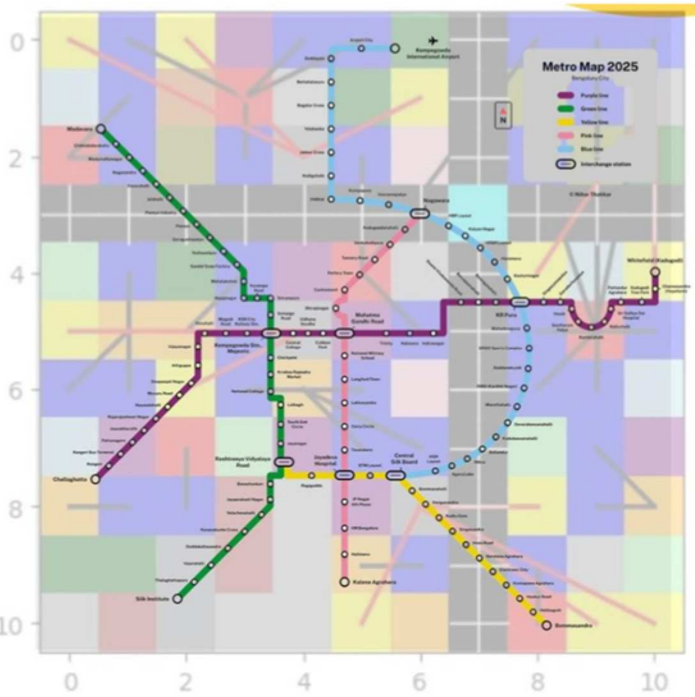
V. Future Scope

1. Our current layout is in a square shape, cities need not have a fixed shape. Modifications for shapes need to be done.
2. The current layout gives a more macroscopic view of the city, but planning minor aspects also plays a major role while planning a smart city.
3. The road infrastructure is a basic idea and can be scaled as desired.
4. Amenities such as bus stops, charging stations, surveillance systems, etc can be added. Other factors such as Terrain differences can be scaled further.
5. As it is a smart city, renewable energy methods have been incorporated, but changes can be made.
6. Additional data can be gathered and models could be trained to decide the percentages of each zone.
7. Multiple technologies such as GIS and IoT can be incorporated to take the terrain of the location in consideration while generating the layout.

VI. Use Case

If a party wants to convert a city to a smart one, he or she needs to overcome the problems due to space and resource allocation, transport management and sustainability. Our solution helps them to generate their desired layout. In our plan, fixed locations and empty spaces have been accounted for. Other than this, sustainability is maintained as industries are powered by renewable energy and forestation is placed right next to it. For transport, we suggest Metros, Railways, Wider Highways and basic roadways. Metro layouts can be adapted according to existing city maps or future plans, for example: Bangalore and Delhi.

Fig. 9 is one such example where the 2-D layout generated is placed upon the current metro infrastructure of Bangalore. As the layout can practically be generated an infinite number



of

Fig. 9. Map of Bangalore Metro on our layout showing possible connectivity [10]

times, various layouts can be generated until a desired layout is generated. In this case a desired layout would be which incorporates important parts of the current infrastructure (i.e. Bangalore metro in this example) while also assigning zones, driving it towards becoming a smart city.

Another use case for this solution is, if the people or government want to shift the city from a predominantly residential city to an industrial hub or vice versa. This transformation if not monitored properly which may seem insignificant in the start, starts to build up in the future causing imbalances. While shifting to a more industrial development, the cities original features should not be affected and A layout can be generated based on the current features which need to be preserved, empty spots can be allocated for areas where have to developed in the future.

Using this solution also ensures that there is a balance maintained in the city and is not overpowered by the development of the industries. The percentage of green zones to allocated can be increased if high number industries are planned to be developed in the city. This is leads to a more planned monitored and stable approach of developing an existing city.

VII. Conclusion

This Python based solution is a significant addition in making city layouts extremely efficient through computational methods. By using a 2-D grid and 16 zones, the tool is flexible and adaptable to aid in city planning, allowing visualisation and consideration of multiple land zones while keeping zonal constraints in mind. As cities continue to evolve, this tool provides a valuable resource for generating efficient, sustainable, and adaptable urban layouts, ultimately contributing to the development of smart cities. The possibility of adaptation with various technologies makes this code extremely adaptable and flexible for all kinds of use cases. Practices such as reduced carbon footprint, renewable energy methods ensure that the layout is made keeping the future in mind.

In majority cases, the algorithm satisfies constraints with above 90% accuracy. The use cases can be explored as much as the user desires and is not limited to smart cities. The code can be revised to change constraints of the zone but this requires prior knowledge of Python.

The algorithm can help people generate proposals based on any kind of problem statement where layout generation is needed.

This project not only addresses the immediate needs of urban planning but also lays the groundwork for future enhancements. The incorporation of machine learning and artificial intelligence concepts offers the potential for more dynamic and responsive city planning solutions.

VIII. References

[1] Smart Cities : Ministry of Housing and Urban Affairs, Goverment of India. (n.d.). https://mohua.gov.in/cms/smart-cities.php

[2] The Editors of Encyclopaedia Britannica. (2023, April 14). *Euclidean distance | Definition, Formula, & Facts*. Encyclopedia Britannica. https://www.britannica.com/science/Euclidean-distance

[3] Ojo, A., Curry, E., Janowski, T., & Dzhusupova, Z. (n.d.). Designing next generation Smart City Initiatives: the SCID Framework. In *Transforming City Governments for successful Smart Cities*. https://doi.org/10.1007/978-3-319-03167-5_4

[4] Gao, C., Wang, F., Hu, X., & Martinez, J. (2023). Research on sustainable design of smart cities based on the internet of things and ecosystems. *Sustainability*, *15*(8), 6546. https://doi.org/10.3390/su15086546

[5] Sharifi, A., Allam, Z., Feizizadeh, B., & Ghamari, H. (2021). Three decades of research on smart Cities: mapping knowledge structure and Trends. *Sustainability*, *13*(13), 7140. https://doi.org/10.3390/su13137140

[6] Lee, J. (2021). Smart city in urban design. *www.sbt-durabi.org*. https://doi.org/10.22712/susb.20210031

[7] *SMART MAPS FOR INDIA*. (2015). https://smartnet.niua.org/sites/default/files/resources/20150427_Google-Smart-Maps-Report_v10_NO-MAP-copy_0.pdf

[8] J. C. Demmler *et al.*, "A novel approach of creating sustainable urban planning solutions that optimise the local air quality and environmental equity in Helsinki, Finland: The CouSCOUS study protocol," *PLoS ONE*, vol. 16, no. 12, p. e0260009, Dec. 2021, doi: 10.1371/journal.pone.0260009.

[9] *Icograms 3D Map Online Designer - create maps, diagrams, infographics and illustrations from isometric icons*. (n.d.). https://icograms.com/designer

[10] *File:Metro Map 2025 - Bengaluru City.pdf - Wikimedia Commons*. (2023, March 31). https://commons.wikimedia.org/wiki/File:Metro_Map_2025_-_Bengaluru_City.pdf