# Digital Twins for Smart Cities: Case Study and Visualisation via Mixed Reality


William Piper
*Department of Engineering, Durham University*
Durham, UK
william.j.piper@durham.ac.uk

Hongjian Sun
*Department of Engineering, Durham University*
Durham, UK
hongjian.sun@durham.ac.uk

Jing Jiang
*Department of Mathematics, Physics & Electrical Engineering, Northumbria University*
Newcastle, UK
jing.jiang@northumbria.ac.uk



***Abstract*—Digital twins is an increasingly valuable technology for realising smart cities worldwide. Visualising this technology using mixed reality creates unprecedented opportunities to easily access relevant data and information. In this paper, a digital twins-based system is designed to visualise information from a city's street lighting system. Data is obtained in two ways: from measured parameters of a miniature model street light in real-time, and from real Durham street lighting. Machine learning is used to maximise the efficiency of purchasing electricity from the grid, and to forecast appropriate adaptive street light brightness levels based on city's traffic flow and solar irradiance. An application designed in Unity Pro is deployed on a Microsoft HoloLens 2, and it allows the user to view the processed data and control the model street light. It was found that the application performed as desired, displaying information such as voltage, current, carbon emission, electricity price, battery state of charge and LED mode, while enabling control over the model street light. Moreover, the Deep Q-Network machine learning algorithm successfully scheduled to buy electricity at times of low price and low carbon intensity, while the Long Short-Term Memory algorithm accurately forecasted traffic flow with mean Root-Mean-Square Error and Mean Absolute Percentage Error values of 12.0% and 20.0% respectively.**



***Index Terms*—Digital Twin, Mixed Reality, Augmented Reality, HoloLens 2, Street Lighting, Machine Learning, Adaptive Dimming.**


## I. Introduction

THE early 2000s saw the rise of digital twins, a technology increasingly used in the industrial sector for seamless data transfer between the physical and virtual worlds [1] [2]. Such a technology is valuable for producing analytical data about the twin's physical counterpart. Mixed reality was defined by P. Milgram and F. Kishino as an environment "in which real world and virtual world objects are presented together within a single display, that is, anywhere between the extrema of the virtuality continuum" [3]. Despite augmented reality (AR) existing as a subset of mixed reality, the latter has the notable benefits of enabling the user to walk into and manipulate a scene, while simulating aspects of reality in the virtual environment [4]. There are numerous ways mixed reality can be deployed, namely head-mounted displays, hand-held displays, monitors, projectors, and 2D smart glasses [5]. Although all of these, head-mounted displays are the industry standard for mixed reality due to their ability to produce immersive, stereoscopic displays, and create a seamless transition between real and virtual environments [6].

There exists plenty of research and development on digital twins and mixed reality individually, however the focus of this paper is on the combination of the two technologies, which is a more niche area of study. Although existing literature on the visualisation of digital twins is limited, most sources suggest that manufacturing and maintenance are currently the most relevant applications for the technology in industry, due to the time and money that can be saved through the utilisation of digital twins, and the relatively simple yet critical information that can be displayed using mixed reality [5], [7], [8]. Predictive maintenance is a fundamental aspect of digital twinning that can be applied to both manufacturing and street lighting. Simple techniques such as visual inspection of relevant parameters from the digital twin, and further evolution of these methods such as automated techniques involving advanced signal processing are invaluable to the progression of this technology in industry. Carvalho *et al.* specified in [9] that many different machine learning techniques, such as Support Vector Machines, Random Forests, Artificial Neural Networks, deep learning, and $k$-means, have been successfully applied to design predictive maintenance applications.

Creating digital twins of a city's street lighting system promises profound implications on system improvement: current literature stipulates that the main improvements are LED predictive maintenance and reduced energy consumption through adaptive dimming or dynamic switching [10]–[13]. These concepts fall under the broader category of "Smart City". Machine learning techniques for predictive time-series models can contribute to the implementation of these adaptive dimming technologies for cities' street lighting.

Within the current industry, there lies a research gap in bridging the gap between digitally twinned street lighting and visualisation and control of the digital twin using mixed reality. There currently exists no research on or development of a system that utilises mixed reality to effortlessly view crucial real-time information such as voltage, current, brightness, carbon emission, and electricity price data, while also enabling control of the lights, such as adaptive brightness control. Advanced


This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 872172 TESTBED2 project.

research and technology exists for both mixed reality and digital twins, however combining these two technologies for an application as fundamental and globally relevant as street lighting has the capability to significantly further the state of the art. Such a system could be beneficial to multiple stakeholders, for example:

- For a technician up a ladder fixing a street light, it would minimise the distraction of needing to read a separate display for information.
- When trying to correlate a spreadsheet of data with an entire street of lights, it would diminish the human error of gathering information for the wrong street light.
- It would enable a company manager or a local council head of department to walk through a street and check the lights meet the required specifications, without the need for any technical knowledge or assistance.

The benefits of this system are vast, because ease of access of information is the nucleus of this area of industry. Hence, such a system becomes the focus of this paper.

## II. Methodology and System Operation

### A. *Whole System*

As shown in Fig. 1, the whole system consists of two separate parts: the back end and the front end. The back end entails the digitally twinned street lighting, which involves a combination of a miniature model street light digital twin and a real street lighting system digital twin. Meanwhile, the front end consists of mixed reality visualisation of the digital twins through an application developed on Unity Pro. This app is deployed on a Microsoft HoloLens 2 mixed reality headset.

The essence of a digital twin is a model that is dynamically updated according to an evolving set of data corresponding to the physical twin, which in this case is street lighting. This requires sensors to collect data from the physical twin, and a means of communication between the physical twin and the digital twin.

Part of the system has been developed using a miniature 3D-printed model lamp post with an LED attached to it, as shown in Fig. 1. The purpose of the model is to imitate interaction with a real street light. This is done by monitoring and controlling aspects of the light such as voltage, current and brightness through the digital twin. To obtain the required input data for the digital twin, the LED from the model has been connected to an Adafruit PCF8591 Quad 8-bit ADC + 8-bit DAC, which has been in turn connected to a Raspberry Pi. The Adafruit ADC/DAC enables the voltage of the LED to be both read and set by a Python script integrated with the Adafruit CircuitPython library on the Rasberry Pi [14]. Analysis and organisation of the digital twin data occurs within the Python program.

### B. *Real Street Lighting System Digital Twin*

Creating a comprehensive digital twin of a system requires not only data collection from sensors, but also relevant and analytical processing of this data. This notion is applied by integrating into the system a real Durham city street lighting digital twin, using time-series data attained from both public online databases and Durham County Council. The data points occur at 30 minute intervals and consist of traffic, carbon emission, electricity price and photo-voltaic (PV) data from 2019, with data from other years used for training [15]–[17].

Actual Durham street lights are not connected to batteries, however for the sake of better decarbonising cities, the real street lighting physical twin is considered to be able to source power from either batteries or the grid, depending on the solar irradiance. In this scenario, the connected batteries store surplus power from PV panels during periods of high solar irradiance, and store energy from the grid when the PV panels cannot supply enough power. When the state of charge (SoC) of the batteries is greater than 80%, the batteries are used to power the lights, otherwise the power grid is used and the batteries are charged. From the specified input data and scenario stipulations, various techniques have been deployed to build the digital twin.

*1) DQN Scheduling Algorithm:* Based on the aforementioned input data provided, the digital twin of this system uses machine learning to optimise carbon emission and electricity price, with the intention of creating a Smart City's street lighting system. The primary implemented algorithm is a Deep Q-Network (DQN), which is used to identify and schedule

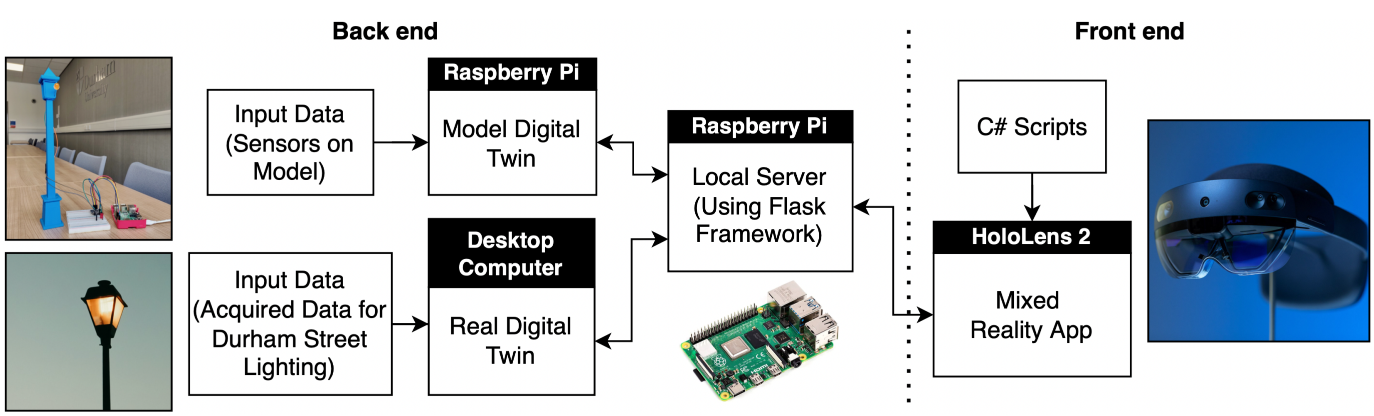


Fig. 1. Simplified System Flowchart Diagram

the optimal time to purchase electricity from the grid, based on carbon emission and electricity price. The algorithm also determines whether to use the batteries or the grid to power the street lights.

*2) Forecasting-based Adaptive Dimming:* Given the importance of displaying relevant information to the user, and the ability to control the street light brightness, a natural area of development is adaptive dimming of the street lighting. The basis of this technology combines forecasted PV data with forecasted traffic data to calculate an appropriate brightness level for the street lighting.

A Long Short-Term Memory (LSTM) network was designed in Python, and is used to periodically forecast traffic flow for the next 12 hours (from the current time of day) based on the previous 4 weeks of traffic data[1]. The provided traffic data exists at 15 minute intervals. The parameters for the LSTM network are dimensionless quantities and shown in Table I.

A separate LSTM algorithm is also utilised to contribute to the DQN scheduling, notably to forecast PV power. It is self-evident that PV power is directly correlated to solar irradiance, and therefore also to the 'brightness' of the sun. Predicted daylight brightness and predicted traffic flow can be combined as shown below

$$Br = Br_0 + \frac{Tr}{Tr_{MAX}} \times \left(1 - \frac{Pv}{Pv_{MAX}}\right) \times (100 - Br_0), \quad (1)$$

where $Br$ is predicted street light brightness for the next half hour, $Br_0$ is base level of brightness, $Tr$ is predicted traffic flow for the next half hour, $Tr_{MAX}$ is maximum traffic flow over the next 12 hours, $Pv$ is predicted PV power for the next half hour, and $Pv_{MAX}$ is the maximum PV power over the next 12 hours. This equation provides a value for predicted brightness with the underlying rule that street light brightness must increase for an increase in traffic flow or a decrease in daylight brightness, and vice versa.

TABLE I
LSTM PARAMETERS

| LSTM Parameter | Value |
|---|---|
| Steps In | 15 data points |
| Steps Out | 48 data points |
| Epochs | 200 |
| Optimiser | Adam |
| Loss Function | Mean Squared Error |
| Features | 1 |

### C. Communications

Paramount to the system is bidirectional communications between the digital twin and the HoloLens app as it enables the processed data to be displayed to the user. The system at hand adopts an Application Programming Interface (API) via a client-server model, where the HoloLens, acting as the client, makes service requests from the Raspberry Pi, which acts as the server. As shown in Fig. 1, the Raspberry Pi Python script is integrated with Flask, a simple but extensible micro web framework that handles Hypertext Transfer Protocol (HTTP) GET requests from the client to the server of the parameters to be displayed in the HoloLens app [18].

Data is transferred to and from the HoloLens app using the JavaScript Object Notation (JSON) data format, which is designed to be fast, human readable, and easy for computers to parse and use [19]. It was therefore deemed a suitable option to send data to the app, since Unity Pro, the app development software, has the appropriate functionality for JSON handling. The JSON file is sent via an in-built Flask function that returns the JSON file as text. Visualising the graph of net power demand is achieved through GET requests within the Flask framework. Prior to the data files being hosted on the server, they are transferred to the Raspberry Pi from the computer via Secure Shell Protocol, which allows file transfer over the local network.

### D. HoloLens App

Unity is one industry leading platform for 3D application development, and has excellent integration for mixed reality development using the Microsoft Mixed Reality Toolkit (MRTK) [20], [21]. The Unity program for this project has been assembled by creating 3D 'game objects' and corresponding programming scripts written in C#. The exported Universal Windows Platform app has been compiled and deployed to the HoloLens using Microsoft Visual Studio. The bedrock of the deployed app is the delivery and visualisation of relevant and dynamically updated information to the user. This is achieved through displaying the computer-aided design (CAD) model of the previously discussed 3D-printed miniature model street light as a game object in Unity, and overlaying the processed digital twin data fetched from the server.

Within the C# script responsible for retrieving the JSON data, the function 'UnityWebRequest' triggers the GET request over the local network, and the 'SimpleJSON' package for Unity unpacks the fetched JSON file, which is subsequently formatted as string variables to be displayed in the application UI based on the current time of day. Moreover, this C# script has functions integrated into the application UI for the brightness control slider, the on/off toggle switch, and the auto-brightness activation switch. The brightness control and on/off functions update the brightness variable upon change in the slider value or button press from the user input, while the auto-brightness activation button changes the brightness variable to the calculated value from the LSTM algorithm.

## III. RESULTS AND DISCUSSION

### A. HoloLens App Integrated with Adaptive Brightness

Fig. 2 shows still one image of the application UI, and a video demonstration of the application is available to view at [22]. It is clear to see the CAD model of the street light and the sphere within it acting as the light itself, while all the information about the status of the light is displayed around it. The 3D virtual CAD model remains stationary in the user's

[1]New Elvet street in Durham (data from 2019 provided by Durham county council)

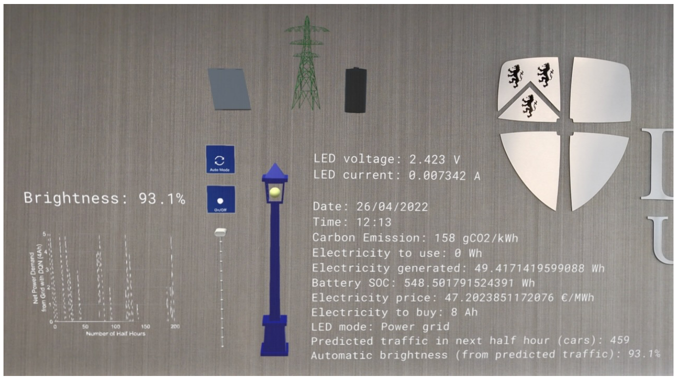


Fig. 2. Digital Twin Visualised in The HoloLens App

surroundings unless the user decides to manipulate the object via gestures such as pinching and dragging to move and resize it, enabled by the MRTK. The figures also show the MRTK interactable objects, notably the brightness slider, the on/off button and the auto-brightness button.

*1) User Interaction:* Upon a change in value from the user input, the brightness slider changes the brightness variable, which not only visibly changes the brightness of the digital light within the app, but also alters the brightness of the miniature model street light LED accordingly. This is achieved through query arguments outlined in section II-C. In a self-explanatory manner, the on/off button sets the digital light and real light brightness level to 0% if it was not previously 0%, and sets it (and the slider) to 100% if it was previously 0%. Regardless of the current state of the street light, the 'Auto Mode' button triggers the light brightness to be set to the predicted level that has been calculated using equation (1).

*2) DQN Scheduling Visualisation:* The 3D icons situated behind the virtual lamp post in Fig. 2 signify part of the DQN scheduling output. Either the power grid icon or the battery icon will light up green, depending where power is being sourced from. It is clear from Fig. 2 that power is being sourced from the grid and therefore the power grid icon is highlighted in green.

## *B. DQN Algorithm*

To achieve useful output results, the DQN scheduling algorithm combines training techniques followed by data processing based on the output of the training. The DNN of the DQN is 30 × 30 in size, with the Levenberg-Marquardt training function. Fig. 3 shows the testing results of the DQN based on the described training. From a randomly selected set of 200 half hour intervals (100 hours), it is clear to see how the carbon emission intensity and electricity price both fluctuate within the bounds of 100-300 $gCO_2/kWh$ and 35-60 $€/MWh$ respectively. The right hand axes of Fig. 3 specify, from the DQN, how much electricity to purchase from the grid, and when to do so. In this case, it is evident from the first sub-figure that the decision to buy electricity was only affirmed when the electricity price was low; for example, intervals around 0-5, 20, 40-55, 90-100, 140 and 190-200. As shown in the second sub-sigure, these intervals also correspond to relatively low carbon emission periods, therefore indicating improved power efficiency due to the DQN.

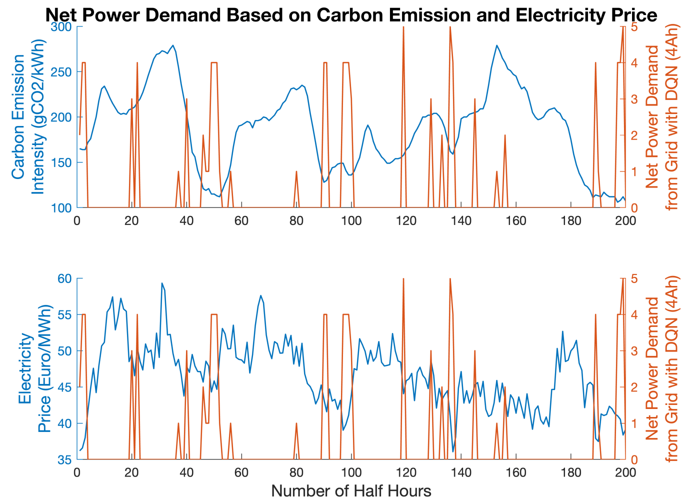


Fig. 3. With Respect to Time: Carbon Emission Intensity, Electricity Price, and Net Power Demand from DQN

## *C. LSTM Algorithm*

To assess the performance of the LSTM forecasting, the method presented in section II-B2 was executed 8 times across different start times. Since the provided traffic data exists for the whole year of 2019, the predicted data for a given period of time has been compared with the corresponding real data. To quantify the error of the predicted data from the real data, values of RMSE and Mean Absolute Percentage Error (MAPE) were calculated using below equation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2},\ MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right|, \tag{2}$$

where $i$ is a given data point for total number of data points $n$, $x_i$ is the real value, and $\hat{x}_i$ is the predicted value.

It was found that across the 8 rounds of testing, the mean RMSE was 0.120 (12.0%) and the mean MAPE was 20.0%, indicating that the predicted values fit the actual data set well. Fig. 4 shows the set of results with the best performance out of the 8 sets calculated, with an RMSE of 0.0985 (9.85%) and a MAPE of 14.8%.

It is clear from Fig. 4 that the algorithm is adept at accurately predicting the general trend of traffic flow, with a correct predicted rise in traffic from around 0-250 minutes, followed by a steady decrease from 250 minutes onward. The algorithm clearly handles fluctuations in traffic well, producing a smooth prediction level at around the average level of oscillations in the real data.

Despite the good performance metrics and the accurate fit of the graph, there is still a slight overshoot in the predicted values between around 150 and 270 minutes, as is evident

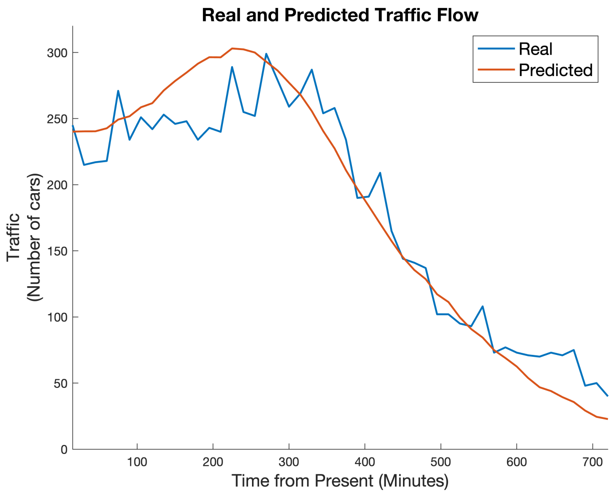


Fig. 4. Real and Predicted Traffic Flow for a Random Time of Day

from Fig. 4. This could be due to an anomalous fall in traffic flow on that particular day, or due to the unpredictability of the fluctuations in the training data, which could instill a slight instability in the predicted values.

Different factors of the LSTM model could be further fine-tuned to attempt to produce more accurate results, such as increasing the size of the training data set or performing more training iterations. However, given the constraints of computing time and power, the current performance is adequate for the application of adaptive dimming.

## IV. Conclusion

The developed digital twin system provides real-time, relevant information to the user. This occurs through bidirectional communication between the digital twin displayed within a HoloLens 2 application, and the physical parts of the street lighting. Consisting of an LED and a 3D-printed lamp post, the miniature model part of the system utilises the Flask micro web framework hosted on a connected Raspberry Pi. Meanwhile, the real street lighting system is based on data from 2019.

The deployed app achieves desired performance goals, notably visualisation of data and control of the physical twin through user input. MRTK gestures successfully allow the user to manipulate the virtual lamp post, alter the linked physical and virtual lights through an on/off button, a brightness slider, and an auto-brightness button.

Furthermore, the digital twin integrates data processing through DQN and LSTM machine learning techniques. These processes decide the optimal time for buying electricity from the grid, and forecast appropriate levels of adaptive dimming due to traffic flow and PV levels. Accurate results were produced by the algorithms for the objective at hand, therefore contributing to a comprehensive and fully functioning system. For future developments, access to real-time traffic, carbon emission, electricity price and PV data would be paramount, and additional machine learning techniques could be used to promote improved predictive maintenance.

## References


[1] A. El Saddik, "Digital twins: The convergence of multimedia technologies," *IEEE multimedia*, vol. 25, no. 2, pp. 87–92, 2018.

[2] L. Wright and S. Davidson, "How to tell the difference between a model and a digital twin," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, no. 1, pp. 1–13, 2020.

[3] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE TRANSACTIONS on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.

[4] M. Speicher, B. D. Hall, and M. Nebeling, "What is mixed reality?" in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–15.

[5] Z. Zhu, C. Liu, and X. Xu, "Visualisation of the digital twin data in manufacturing by using augmented reality," *Procedia Cirp*, vol. 81, pp. 898–903, 2019.

[6] M. Hoover, "An evaluation of the microsoft hololens for a manufacturing-guided assembly task," Ph.D. dissertation, Iowa State University, 2018.

[7] C. Qiu, S. Zhou, Z. Liu, Q. Gao, and J. Tan, "Digital assembly technology based on augmented reality and digital twins: a review," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 6, pp. 597–610, 2019.

[8] S. Rabah, A. Assila, E. Khouri, F. Maier, F. Ababsa, P. Maier, F. Mérienne *et al.*, "Towards improving the future of manufacturing through digital twin and augmented reality technologies," *Procedia Manufacturing*, vol. 17, pp. 460–467, 2018.

[9] T. P. Carvalho, F. A. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.

[10] M. S. Ibrahim, J. Fan, W. K. Yung, A. Prisacaru, W. van Driel, X. Fan, and G. Zhang, "Machine learning and digital twin driven diagnostics and prognostics of light-emitting diodes," *Laser & Photonics Reviews*, vol. 14, no. 12, p. 2000254, 2020.

[11] S. P. Lau, G. V. Merrett, and N. M. White, "Energy-efficient street lighting through embedded adaptive intelligence," in *Proc. IEEE International Conference on Advanced Logistics and Transport*, 2013, pp. 53–58.

[12] M. M. Rathore, S. A. Shah, D. Shukla, E. Bentafat, and S. Bakiras, "The role of AI, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities," *IEEE Access*, vol. 9, pp. 32 030–32 052, 2021.

[13] F. Ramadhani, K. A. Bakar, M. Hussain, O. Erixno, and R. Nazir, "Optimization with traffic-based control for designing standalone streetlight system: A case study," *Renewable energy*, vol. 105, pp. 149–159, 2017.

[14] Adafruit, "Adafruit circuitpython." [Online]. Available: https://learn.adafruit.com/welcome-to-circuitpython

[15] NASA, "The power project." [Online]. Available: https://power.larc.nasa.gov

[16] European Commission, "PV data in Durham." [Online]. Available: https://re.jrc.ec.europa.eu/pvg_tools/en/

[17] National Grid ESO, "Carbon intensity." [Online]. Available: https://carbonintensity.org.uk

[18] Flask, "Welcome to flask." [Online]. Available: https://flask.palletsprojects.com/en/2.0.x/

[19] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of json and xml data interchange formats: a case study." *Caine*, vol. 9, pp. 157–162, 2009.

[20] Unity, "Unity real-time development platform." [Online]. Available: https://unity.com

[21] Microsoft, "What is the mixed reality toolkit," Nov 2021. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05

[22] "App demo." [Online]. Available: https://bit.ly/3EQL2Fr