Opportunistic Digital Twin: an Edge Intelligence enabler for Smart City

CLAUDIO SAVAGLIO, University of Calabria, Italy
VINCENZO BARBUTO, University of Calabria, Italy
FARAZ MALIK AWAN, Urban Big Data Center, UK
ROBERTO MINERVA, Telecom SudParis, Institut Polytechnique de Paris, France
NOEL CRESPI, Telecom SudParis, Institut Polytechnique de Paris, France
GIANCARLO FORTINO, University of Calabria, Italy

Although Digital Twins (DTs) became very popular in industry, nowadays they represent a pre-requisite of many systems across different domains, by taking advantage of the disrupting digital technologies such as Artificial Intelligence (AI), Edge Computing and Internet of Things (IoT). In this paper we present our "opportunistic" interpretation, which advances the traditional DT concept and provides a valid support for enabling next-generation solutions in dynamic, distributed and large scale scenarios as smart cities. Indeed, by collecting simple data from the environment and by opportunistically elaborating them through AI techniques directly at the network edge (also referred to as Edge Intelligence), a digital version of a physical object can be built from the bottom up as well as dynamically manipulated and operated in a data-driven manner, thus enabling prompt responses to external stimuli and effective command actuation. To demonstrate the viability of our Opportunistic Digital Twin (ODT) a real use case focused on a traffic prediction task has been incrementally developed and presented, showing improved inference performance and reduced network latency, bandwidth and power consumption.

CCS Concepts: • **Information systems → Computing platforms**; • **Computing methodologies** → *Distributed computing methodologies*; • **Computer systems organization → Embedded and cyber-physical systems**.

Additional Key Words and Phrases: Digital Twins, Edge Intelligence, Internet of Things, Synthetic Sensing

1 INTRODUCTION

Modern cities, before being smart, need to be measurable, representable by accurate models and actionable. In most cases, this dictates the availability of a large data infrastructure, an extensive sensing and actuation basis made out of thousands of sensors, and an effective distributed software infrastructure capable of collecting data and use them to model, monitor, and act upon programmable functionalities. Only in such a way, indeed, modern cities can reflect the complexity of the phenomena occurring in the urban environment and provide cyber-physical, next-generation services, thus eventually evolving into truly smart cities. However, the design of a smart city poses multi-faceted engineering challenges for the modeling of the relevant phenomena occurring in

Authors' addresses: Claudio Savaglio, csavaglio@dimes.unical.it, University of Calabria, Rende, Italy; Vincenzo Barbuto, vincenzo.barbuto@dimes.unical.it, University of Calabria, Rende, Italy; Faraz Malik Awan, farazmalik.awan@glasgow.ac.uk, Urban Big Data Center, Glasgow, UK; Roberto Minerva, roberto.minerva@telecom-sudparis.eu, Telecom SudParis, Institut Polytechnique de Paris, Evry, France; Noel Crespi, noel.crespi@mines-telecom.fr, Telecom SudParis, Institut Polytechnique de Paris, Evry, France; Giancarlo Fortino, g.fortino@unical.it, University of Calabria, Rende, Italy.

1550-4859/2023/8-ART $15.00
https://doi.org/10.1145/3616014

the environment as well as for the difficulty of implementing, deploying, and managing the needed resources and functionalities: this is where the Digital Twins (DTs) come on stage.

Indeed, in recent years, advanced virtualization techniques and distributed Edge Intelligence (EI) [6, 48] have emerged as key elements to enable the development of the outlined smart cities platforms and related services in a decentralized way and in spite of their inherent complexity. On the one hand, virtualization allows for better exploitation of computing, storage and sensing resources, leading to the possibility of creating a very capillary infrastructure able to monitor and control local resources. On the other hand, cloud computing is now complemented by edge computing in the attempt to deal locally with emerging behaviors and promptly react to phenomena that occur in the urban environment: in particular, EI guarantees the possibility of limiting the need of centralizing data and functionalities in the cloud by executing relevant intelligent functionalities where they are needed [28]. The synergistic exploitation of virtualization and EI techniques, therefore, promises to ease both the representation and the programming of complex environments, and the DT definition exactly serves this scope, by providing straightforward metaphors for virtualizing smart cities infrastructures as well as a sound basis for injecting intelligence into its operating systems. As a matter of fact, the application of the DT paradigm in the city is acquiring a certain interest from the academic and industrial communities, motivated by the interest towards a continuous and evolving representation of the status of the urban environment[1, 2, 3, 4, 5].

Along this direction, in this manuscript we conceptualize the *Opportunistic Digital Twin (ODT)* definition as a novel, interdisciplinary approach having the objective to ease the (re)engineering of large-scale distributed smart systems, exactly like smart cities, by maximizing the exploitation of their infrastructure and resources. In particular, we define as "Opportunistic" a DT, created from scratch or upon other existing DTs, modeling a single physical object (PO), an ensemble of POs or a phenomenon through their distinctive information formalized within a repository called signature. The signature is purposely created for the ODT from a selection of the available (mainly sensory, but also historical) data, gathered from a single source or from multiple ones for being post-elaborated through AI techniques, as the synthetic-sensing theory describes. The elaboration of the signature data can have place on the cloud or, as desirable, close to the virtualized POs/phenomena: as result, synthetic features can be obtained to enrich the representation of DT/PO as well as to rapidly impact on their operations. Indeed, the signature data enable a better understanding of the DT/PO status and of the surrounding context and, therefore, PO's operation can be dynamically undertaken or finely tuned (e.g., the sampling rate of a temperature sensor) in a context-aware manner and according to the individual/collective needs of a given situation. Such a data-driven and bottom-up approach is *(i)* original with respect to the state-of-the-art since, typically, the DT representation is statically and a priori defined to exactly mirror the actual features of a PO; *(ii)* extremely profitable for the smart city context, where there is availability of many general-purpose sensors while dedicated devices are costly, less versatile and more invasive; *(iii)* beneficial for injecting, through the DT platform, more smartness, context-awareness and dynamics to the smart cities' services, even on large scale and in a distributed way, and for supporting simulations. As a matter of fact, we want to investigate if the exploitation of ODTs can deeply impact on current and future smart city design by moving complexity from hardware infrastructure to software layer, and, in general, eventually unleash the great potential of the Internet of Things (IoT). As proof-of-concept for our approach, we present a research path towards the incremental implementation of ODTs in terms of a real use case focused on one typical activity of smart city, namely the traffic prediction. In particular, we focus on the bottom-up and data-driven creation of an ODT of a road by showing the modeling of its synthetic sensing-enhanced signature, its EI-oriented design and its performance evaluation.

[1]https://ru.muenchen.de/2018/194/Digitaler-Zwilling-Bessere-Luft-durch-intelligente-Mobilitaet-80933
[2]https://www.hel.fi/static/liitteet-2019/Kaupunginkanslia/Helsinki3D_Kalasatama_Digital_Twins.pdf
[3]https://virtual.corp.at/200609/virtualcorp2020_20200609_van_der_Heijden.pdf
[4]https://www.3ds.com/insights/customer-stories/rennes-metropole
[5]https://www.3ds.com/insights/customer-stories/virtual-singapore

To sum up, the main contributions of this work are the presentation of the ODT at the confluence of DT, synthetic sensing and EI, and its exemplification through the aforementioned use case. The rest of the manuscript is organized as follows. In Section 2 we provide some background concepts about the building blocks of our approach, namely synthetic sensing, DT and EI. In Section 3 we extensively present the ODT vision, its fundamentals, benefits, challenges and limitations, with a particular focus on the smart city domain which hosts the traffic prediction use case reported in Section 4. Future research directions and final remarks conclude the manuscript in Section 5.

2 BACKGROUND

2.1 Towards Synthetic Sensing: General-Purpose Sensing plus AI

The emerging trend among sensing techniques consists a full-flexible utilization of devices, by populating them with several elementary sensors. The resulting boards are said to be "general-purpose", since they are able to collect a wide range of environmental facets. This new technique is proposed in the literature as *"general-purpose sensing"* [26], in contrast with the "single-purpose" approach, in which one sensor is used for gathering just one measure. Although using a single sensor network tends to be a robust technique (especially when only a few aspects of the environment are monitored) it loses effectiveness when the features to monitor increase, thus requiring the deployment of a larger set of sensing infrastructure. General-purpose sensing, instead, helps to improve environments' digitization process by cutting down deployment, hardware components and maintenance costs, by diminishing the aesthetic and social impacts [7], and, last but not least, by moving the complexity from hardware to software, enhancing empowering sensory capabilities with AI techniques.

With respect to the latter point, the huge amount of data gathered from the environment through the general-purpose devices can feed Machine Learning (ML) models in order to understand stimuli coming from the environment and, accordingly, to act based on them. Such a full exploitation of the versatility of general-purpose devices, presented in [26] as "synthetic sensing", allows transforming heterogeneous low-level sensory data into "semantically relevant representations". Recently fueled thanks to the wider dissemination of "general-purpose" devices, the "synthetic sensing" is not only a new sensing method to *listen* to the environment, but also an enabler to power-up sensory features with AI by moving complexity to the software layer. For instance, Trifan, et al. [42] showed how it is possible to use smartphones' embedded sensors (GPS, accelerometer, gyroscope, etc.) and their behavioral patterns (app usage, social interactions, activities log) to passive monitoring users' physical and mental health. Also authors of [17] recognized that, thanks to improvement in video-based sensing, home-assistant devices, tablets and, in general, boards equipped with cameras are also close to achieve the objectives set by general-purpose sensing: through techniques such as object detection and image classification, they could use video streams along with other data acquired by sensors to predict complex events that fully describe the environment of interest. Platforms of both Remote Health Monitoring [33] and Human Activity Recognition [2] can also benefit from the comprehensive approach of synthetic sensing as well as smart system focused on automation could leverage on enriched data to feed their complex ML algorithms [22].

2.2 Digital Twins in the Smart City context

According to Kritzinger, et al. [24] there are plenty of definitions of DT and there is not a reference one, since it depends on the research's focus area. The term "Digital Twin" – as a digital equivalent to a PO – was first introduced by Grieves in 2003 [21] within the context of Product Lifecycle Management (PLM). Grieves coined this term in reference to a three-dimensional concept model that encompasses key elements: the PO existing in the physical space, its virtual counterpart i.e., the logical object (LO) in the virtual space, and the data/information connection between them. Then, in 2012, NASA and the U.S. Air Force defined a DT as an integrated multiphysics, multiscale, probabilistic simulation of an as-built system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin [20]. More recently, in the industry world, the DT has been defined as an executable virtual model of a physical thing or system [23, 45, 46], aiming to highlight that the key actor of DT technology is the all-embracing connectivity between the real and virtualized world [13] enabled by the developments in the IoT field – such as remote sensing, data ubiquity, automation, and actuation capabilities [35]. As it is pointed out in [32], DTs present several properties including *(i)* representativeness, *(ii)* contextualization, *(iii)* modeling, *(iv)* reflection and *(v)* entanglement. Each of them contributes to describing the fundamental features of the DTs.

Several industries in many sectors, especially the ones related to large-scale systems featured by huge amount of components (hardware and software) and interactions such as smart grids [44], smart manufacturing [34], smart healthcare [16], just to name a few, are pushing to invest in DT to boost productivity and efficiency. Out of all the aforementioned application domains, the DT technology stands out as being particularly intertwined with smart cities [18], whose resource management has become challenging due to the growth of economic activities and population in metropolitan areas as well as the pressing concern of the climate crisis. DTs offer a solution to these challenges by serving as virtual, evolving models of the physical world, enabling intelligent management and development of complex virtual spaces. This, in turn, facilitates the transition towards digitization that smart cities are striving for and opens up the opportunity of realizing groundbreaking and eco-friendly outcomes aligned with UN's Global Goal 11 [19]. Along this line, a particularly intriguing aspect is the synergistic application of DT and Intelligent Transport System (ITS). For instance, Andrey Rudskoy et al. [38], provide a reference model of services for simulation purposes on predictive analytics which exploits ML techniques along with DT, aimed at reducing the human error inside traffic control centers by favoring operators activities automation. Instead, Sathish A. P. Kumar et al. [25] presented a new DT-centric approach for reducing traffic congestion by performing driver intention prediction: the idea is to gather a massive amount of real-time data coming from cameras and sensors to build the DT of vehicles, i.e., Virtual Vehicle model, and, along with drivers' historical data, feed ML and deep learning (DL) models to gain and predict driver intention. Although the aforementioned systems present novel and innovative approaches, they do not exploit the full potential of IoT technology. In fact, they require both a costly specific-sensing and computing infrastructure since the majority of AI and ML algorithms are performed in the cloud, that is too far from data sources.

2.3 Bring AI to the Edge

The cloud-based ecosystem has largely established itself as a formidable tool in delivering AI-powered applications, thanks to the high-performance computing capabilities offered by remote servers. However, as the demand for real-time and low latency applications increases, cloud technology alone may not be enough to meet these requirements. This is particularly true for applications where the field-to-cloud data transfer time is unacceptable, such as autonomous driving or some forms of healthcare. In these life-critical scenarios, data processing needs to be done as close as possible to data sources to ensure low latency and fast decision-making. Additionally, uploading data to remote servers necessitates a constant internet connection and heightened security measures to prevent privacy breaches. To tackle the limitations faced by cloud-based AI applications, edge computing has been introduced as a groundbreaking solution that seeks to bridge the gap between the cloud and IoT devices. Edge computing brings the power of cloud computing where information originates and, therefore, advocates for relocating computation instead of data, leading to faster processing times and minimized communication lags. With the ultimate goal of extending AI throughout the edge ecosystem, EI [6, 48] a.k.a. Edge AI has risen as a crucial research area. Defined as the marriage of edge computing and AI [29], EI strives to bring intelligence in proximity to the data source, namely at the edge layer of the network. This layer is largely populated by edge and mobile devices, which, although limited by resources and power, possess a wealth of information and can perform real-time analysis. To fully exploit the potential of EI, it is necessary to optimize ML models to effectively run within the limitations of edge and mobile devices. This is often achieved through techniques such as model compression and partitioning, which reduce the model size and computation time, as well as power consumption, while still maintaining desired accuracy. However, optimizing ML models for EI is not a trivial task [15] and requires a deep understanding of the underlying hardware, software, and algorithms as well as of deployment's requirements in terms of network availability, privacy, and security. Overall, a trade-off analysis between performance, accuracy, and resource utilization is a key step preliminary to the EI implementation, making this a challenging but essential area of research.

3 ODT: TOWARDS EDGE INTELLIGENCE BEYOND THE PHYSICAL OBJECT MIRRORING

Hereinafter we present our ODT definition, born at the intersection of Opportunistic Computing, Synthetic Sensing, and EI and targeted at the DT-aided development of large-scale distributed smart systems. We introduce the general ODT approach and relationships with the DT original one in Section 3.1; we discuss foundations, benefits, limitations and challenges in Section 3.2, and finally, we discuss the problem domain of the smart city in Section 3.3, thus anticipating some challenges addressed in the use case of Section 4.

3.1 Integration of Synthetic Sensing and Digital Twin with the "Opportunistic" approach

Opportunism, intended as the exploitation of any chance that brings entities closer to their goals, is a philosophy that deeply impacted the ICT world in the last years [12]. In particular, the recent paradigms of opportunistic networking and opportunistic computing are both data-driven approaches dealing with uncertainty (e.g., topological information is not precise at all) and evolvability (desired output might not-be defined a priori) through the ad-hoc and on-demand exploitation of local and remote resources. Opportunism is also notably relevant for the IoT services, which are supposed to be dynamically created, context-aware, co-located and transient [8] despite their heterogeneity [3].

With these principles in mind, we propose the ODT definition as an evolution of the original DT concept, in which *opportunism* impacts on both the description of a DT and its operation. Typically, the undertaken approach to the DT design consists in shaping its representation exclusively based upon the sensed or available data, according to the vertical notion of a DT reflecting the physical and immutable properties of industrial equipment. A subset of these data representative of the DT characteristic features composes its signature, a repository essential to enable the identification, monitoring, and contextualization of the DT itself with respect to the PO, other DTs and the surrounding environment. Now, suppose more features of the DT signature need to be introduced for enabling novel functionalities. In such a case, more specialized sensors need to be deployed, embedded in the legacy infrastructure and finally integrated in DT representation with an ad-hoc modeling. This kind of approach has obviously limitations in terms of usability and scalability, especially given the expansion of smart city and the increasingly frequent availability of new hardware and services. Therefore, more recently, the introduction of synthetic sensing [26] and its extension to the DT [31] have added the possibility to use AI capabilities aiming to derive additional data from the one directly obtained with "general-purpose" sensing, thus shaping the description of POs into evolvable and opportunistic DT. Indeed, in the ODT vision the representation of the target PO as well as the list of its operations are no longer a priori and statically defined, but they can be incrementally improved thanks to synthetic values and contextual information, purposely elaborated with AI techniques. In such a way, a novel feature can be added to the DT and, leveraging on all the *available* data, the ODT becomes context-aware and able to provide advanced functionalities. We intentionally referred to "available" and not to "sensed" data, since our ODT is designed to interact and collaborate with other DTs and information sources, whether they belong or not to the same organization. The idea is to favor openness instead of isolation to build scalable and elastic ODT, which are crucial properties, especially in large environments. Moreover, the ODT can be purposely created upon existing DTs (e.g., to augment a single DT or to aggregate other ones) for specific, even time-limited, scopes, so that their creation and deletion can be dynamically managed without affecting (also, from the cost viewpoint) the infrastructure. Foremost, by definition our ODT is able to acquire a huge amount of relevant data which, if elaborated through EI techniques, can drive the management and the operations of the ODT itself, with benefits in terms of responsiveness as well as of privacy, energy, bandwidth saving. Hence, EI and ODT results to be complementary and gives the birth to a strong cooperation for pushing intelligence beyond the cloud's boundaries till end-devices. To wrap up, unlike the original approach that strictly limits DT and its functionalities solely to the PO and its sensed data, the ODT offers the flexibility to harness the full potential of general-purpose devices and synthetic sensing, enabling the virtualization of acquired low-level data into semantically meaningful information. Figure 1 exemplify our approach with respect to the ODT of a vehicle. We begin by taking the PO and its directly measurable quantities (such as GPS, fuel levels, and pollutant emissions, among others), and proceed to construct its digital representation, i.e., its DT. A subset of these features is opportunistically selected to assemble its signature (e.g., temperature, CO_2, speed) and synthetically elaborated for specific purposes, e.g., inferring events like traffic congestion if the vehicle alternates between short trips and frequent stops or analyzing the engine status on the basis of its emissions and internal temperature. If necessary, the signature could be opportunistically enriched (e.g., with the addition of fuel levels and GPS) and a new service dynamically enabled (e.g., prediction of fuel consumption based on the analysis of the driver's driving style).

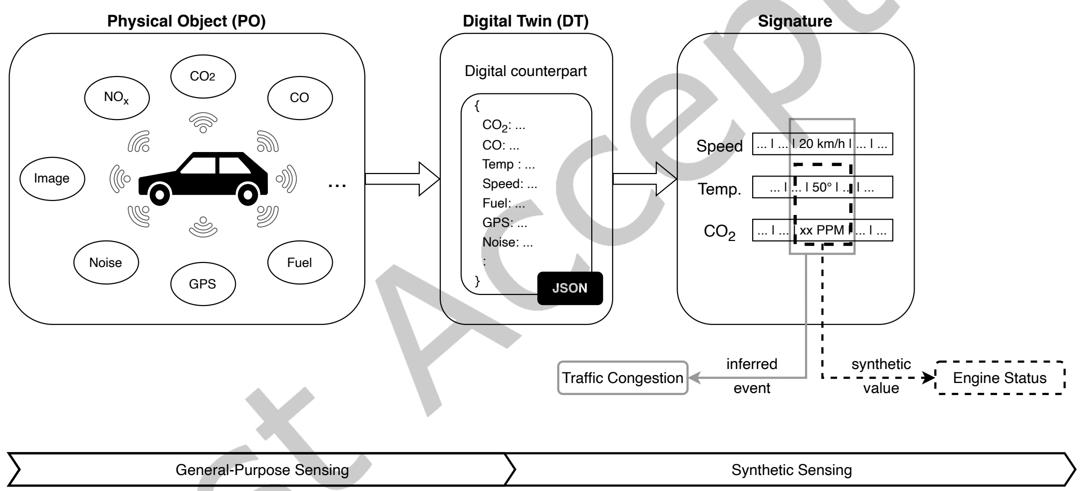

Fig. 1. Proposed approach: from the PO – and its DT – to its signature, in which features are "opportunistically" selected and synthetically elaborated

On these premises, the capabilities of classical DT are surpassed and its aforementioned five distinctive properties improved by the "opportunistic" feature selection envisaged by the ODT approach. In particular, the synthetic elaboration of the signature elevates the DT capacities for representativeness, contextualization and reflection; moreover, if the PO/system is programmable, the ODT can be used to control, change or optimize its behavior, thus also enhancing the entanglement between the real entity and its virtual alias. Then, as for the Digital Shadow approach [39], the representation provided by the ODT is not symmetrical, in the sense that the virtual counterpart is not necessarily communicating with the PO. However, the virtual representation is incrementally built by analysing the behavior of the object in the context and deriving rules or patterns of its actions in the context. There is not a priori knowledge of the PO nor a model describing it: the challenge (and somehow the advantage) of the ODT compared to a predefined descriptive model of the behavior of a DT [36] or Digital Shadow is the ability to "learn" the behavior from experience. One additional step would be to make the representation conform to a model understandable and reusable. Finally, with respect to another important feature of a general DT which is simulation, the ODT can still be used for this purpose. There is a need to train and learn about the expected behavior of the PO and how it relates to the specific operational context but, after this process, the ODT can be used to predict the behavior or, if a set of initial conditions and some time series of data are provided, or even a set of signatures, the prediction capability could be used to simulate the behavior of the PO.

3.2 Foundations, Benefits, and Challenges

The current trend in IoT is to deploy considerable infrastructure for collecting specific measurements and data related to "vertical" phenomena occurring in an environment. The conventional method involves selecting the relevant phenomena and determining the most suitable sensing and actuation infrastructures. This results in a comprehensive vertical infrastructure that generates a wealth of specific data pertaining to the subject. The next step is to format the collected data into well-formed data models and to use them to foster the data interoperability [10]. This approach has the merit of harmonizing data and promote the usage through different environments and use cases. Applications can be built on the common ground of standardized data models. This could be seen as a three stages process that brings from a sensed environment to a programmable environment by a measurable environment. The *sensed* environment represents the collection of "raw" data that the sensors are producing and providing to the entire infrastructure. The *measurable* environment, instead, encompasses the entirety of formatted, curated, and organized sensed data, including their real-time updates. Finally, the *programmable* environment comprises the applications, services, and functions used to provide added value to the final user. These applications and services are developed exploiting the common data infrastructure and specific middleware platforms (e.g., Fiware [10]). In this perspective, DTs can be modelled by using the available data structures that fit the specific model and its purposes. On this basis, it is possible to build smart and compelling services [11]. However, one aspect that is strongly emerging is the need/possibility of considering to what extent the sensed data can be used for inferring and deriving additional data. Data collected and formatted can be used to derive signatures or for inferring new data by correlating different data models. This is an example of application of synthetic sensing [26] to smart cities supported by a generalized representation of DTs [31]. As described in [9], synthetic sensing still has to find its own consolidation and identification of limits and applicability. As a challenge for synthetic sensing and ODT, an additional stage could be introduced between the measurable and the programmable environments. The *inferred* environment, in fact, could represent the set of additional measures that could be derived from available ones as well as the identification, classification, and incremental learning about objects and phenomena operating in the sensed environment. Figure 2 represents the three and four stages approaches and highlights how the DTs definition and modeling could be different. Synthetic sensing poses a number of challenges and questions: from its applicability and scalability to the limit of the inferring capabilities of algorithms. It also poses the challenge of the interaction between reasoning and ML approach. A significant and fundamental question arises regarding the ability to determine when the available sensing infrastructure is adequate to infer and derive the required information. It is crucial to establish a mathematical foundation for comprehending whether a sensing environment is sufficiently comprehensive to provide the necessary data for a specific application. While this theoretical problem is not addressed here, our current focus is on evaluating the practical implementation and value provided by synthetic sensing and the ODT. The next step in this incremental verification of the viability of the approach is to challenge the ODT to represent and understand the environment in which it is embedded. The goal is to investigate whether contextual information can aid in discerning the ongoing activities, understanding the behavior of POs within the environment, and narrowing down the current situation. In this case, an environment can be modeled by defining a minimal set of

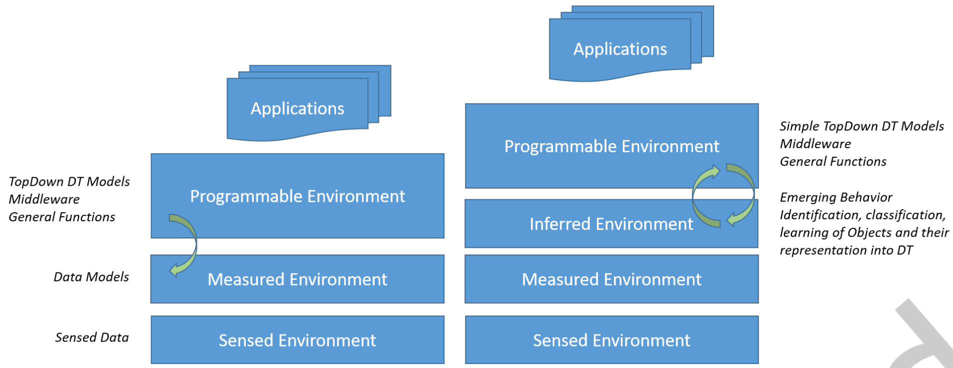

Fig. 2. Three-stage vs. four-stage approach: the impact of Synthetic Sensing on DT definition and modeling.

features and then using ML and reasoning techniques to incrementally grow the representation and description of PO affecting the environment. This will lead to the identification of “normal behaviors” and of objects (and their related ODT) deviating from the expected learned behavior. An example could clarify the approach using the crossroad example. Typically, the flow of traffic in the crossroad can be considered as normal (e.g., no accidents or roadblocks). On a period of time, the ODT approach will learn how to predict the behavior of POs in the normal status. The occurrence of accidents is rare, but after some time, the ODT could progressively “learn” to detect anomalies and detect the behavior of POs in these circumstances. However, the learning of abnormalities will require more time to apprehend and for certain rare occurrences, the ODT could only signal that something different is going on.

As discussed in previous sections, general-purpose sensing can be used to monitor the environment and to create ODT representing the PO operating in the environment. The next challenge is to understand how general-purpose sensing can be used to “measure” and detect events that can be classified as “intended” or “unintended”. The ODT should detect deviations from the learned normal behavior, represent them, and derive additional information to classify them as outliers, triggering specific treatments or warnings. The modeling of the environment can be either “complete”, encompassing all the relevant features, rules, and characteristics for the applications, or minimal yet incorporating ML capabilities for inferring normal functioning. The Situation Awareness will be, in both cases, a representation of objects that are operating accordingly to the expectations and those that are outliers in terms of behavior. The challenge is twofold: to identify the minimal set of sensor capabilities needed to infer the largest possible number of events occurring in the modelled environment; and to determine the minimal set of features that model the environment and support an incremental understanding of the situation. Obviously, adding additional sensing capabilities (e.g., thermal cameras) can result in better event recognition. However, adding new sensors will increase the costs and probably will not substantially improve the results. Another important feature that will be experimented with is the coordination and exploitation of edge cloud continuum [30]. The described processing is expected to take place at the edge. In some cases, the edge capabilities can be instrumental to capture a flow of information and to send it to cloud based resources that can apply more sophisticated algorithms that require more processing power.

As a sort of recap, the choice to model the context (e.g., a crossroad) as ODT instead of individual POs (cars) is due to the fact that the approach will progress from the prediction is a collective and aggregated phenomenon (e.g., traffic) and it will incrementally learn how to spot and determine individual behavior and the effects on the aggregate one. In this manner, the ODT shows a double value in modeling collective behavior (traffic in a certain area) and the ability to identify individual objects’ features and characteristics (vehicle engine malfunction).

3.2.1 Some envisaged limitations. The ODT approach is based on a bottom-up aggregation of information to incrementally create a consolidated representation of POs and their behavior. As such, it is intrinsically based on the comprehension and description of the operating context. This imposes an initial definition of some descriptions and rules related to the context in which POs will act. The more the context is specialized and formalized and the more the description and the capture of objects' behaviors will be precise. This, however, poses two questions: the shift in the initial modeling from the PO to its context, and the specialization of the context description. The modeling of the context can be seen as similar in scope to the description of the PO, but there are some differences: the context is described in terms of expected general rules ("dos and don'ts") that are a bit different from the modeling of a PO's behavior. The bottom-up approach can suffer from adaptation to specific conditions and could result in a loss of generality. Also, the signature could be prone to these generalizations: there is the to understand the applicability of findings in a specific context to a more generalized level. Unfortunately, the lack of modeling for individual POs and the learned models without too much explainability will not help in exporting the knowledge to all possible situations. On another level, security may be a major concern since the ODT approach can be prone to fake input and susceptible to errors induced by external players. In fact, general-purpose sensing can be fallacious or easily turned around by expert software developers. Obfuscation techniques could for instance used to hide or change the signatures and to avoid some POs being observed in the specific context. One possibility for the ODT is to apply counter-measures based on malware detection to mitigate these problems.

3.3 Problem domain: Smart Cities and Future applications of ODT

Our ODT approach is application-domain agnostic, but it results particularly suitable for those large scale scenarios in which the number and typology of both devices and services is ever-growing, exactly as smart cities. There, the availability of sensors/actuators and the request of services is unpredictable and, therefore, an ad-hoc modeling of the overall scenario would result actually unfeasible, as well as interoperability, usability, and integration issues would be intractable on large scale. If the results of our ODT approach will be confirmed, it can lead to a change in how large scale environments should be sensed and equipped. The effect on smart cities could be noteworthy, the infrastructure could be used for deriving more information and signature without requiring necessarily the deployment of additional sensors. The existing infrastructure could be updated in such a way to complement the needed sensing capabilities to the existing ones, in order to achieve the optimal mix of general-purpose sensing features to provide the maximum possible information. In spite of these premises, the work presented tries to be on the practical and reusable side, adopting a step by step strategy for the development of the solutions in order not to waste considerable resources and effort.

An illustrative example within the context of a traffic monitoring system exemplifies this approach. Figure 3 represents a roundabout with vehicles or bikes passing by. This roundabout can be modeled in terms of some characteristics such as allowed spaces and related borders, number of lanes, speed limits, intended directions, capacity in terms of vehicles, and other information. This model can be used to create a contextualized DT of the environment that represents the behavior of the roundabout. It can be used to monitor and to understand how vehicles are using the roundabout and what critical situations may occur. One option is to offer a succinct depiction of the roundabout while employing training techniques to recognize the typical behavior of objects within the surrounding area. General-purpose sensing can catch data about the direction of vehicles, their usual paths, the average, and maximum intensity of traffic. The experiments can be carried out in such a way to start from a minimal description of the roundabout and to determine the "accuracy" of detection of outliers during time. The minimal description, the better. Data models for describing the environment can be built based on the Fiware Data Models and/or the integration of BIM models for the description of urban environments. The figure represents vehicles engaging the roundabout. Each of them can be represented by its own ODT. Some vehicles have a normal behavior, and they are within the limits of the lanes or sectors. The system can follow them, classify them and count them in order to estimate the intensity of traffic, but also to detect anomaly situations. Indeed, some vehicles are clearly behaving in a non-intended manner, e.g., vehicles A and B are out of the allowed spaces, while vehicle C is marching in the wrong direction. Within the context of the roundabout (the yellow contour), it is possible to spot the "outlier" vehicles that are engaging the DT of the roundabout in unintended manners.

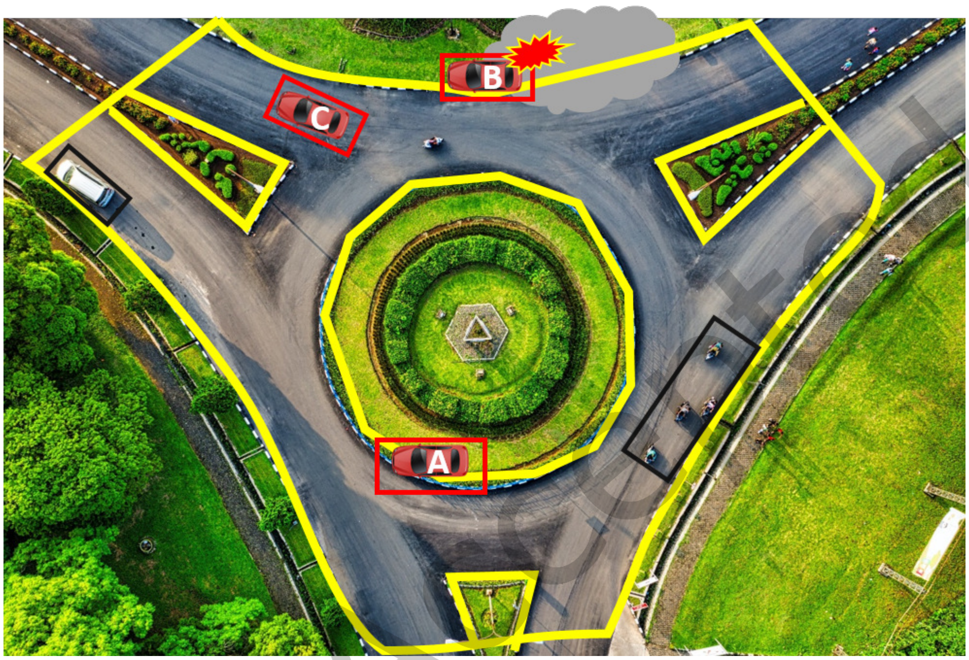

Fig. 3. Contextualized image for a roundabout

The general-purpose sensing equipment for this roundabout comprises cameras (with vehicle counting and movement detection), noise, and pollution sensors. As seen in previous use cases, vehicles A and B can be recognized and classified as "cars". Using specific algorithms, it is possible to determine if the vehicles are moving (and their speed) or if they are stopped. Note that depending on the distance from camera and noise sensor, the combination of their measurements can be used to understand if the vehicle halted and if the engine is running (considering it is not electric). If the vehicle is moving, and it is passing in a contour zone, then an alarm could be generated, and a better recognition of the vehicle itself could be attempted (or simply, the recording of the images can be passed to a better system for further analysis aimed at recognizing the vehicle plate number). If the vehicle stopped, its location could be determined and a warning could be emitted because it is in a dangerous zone. The pollution sensors could then be used to determine if a higher level of pollution is generated (e.g., a higher level of PM values). In this case, a warning about the possibility of a fire or other similar accident could be triggered. Similar mechanisms can be used to understand whether vehicle C is taking a not allowed direction (the DT of the roundabout could have been trained to recognize the habitual sense of direction). The use of noise sensors can be useful in detecting the presence of "noisy" vehicles. This could have two application facets, on the one hand (as experimented in some Indian cities), for regulation of traffic lights of nearby traffic lights in a longer wait for 'noisy' lanes. On the other side, the noise could be a distinctive feature of special vehicles like ambulances or police/fireman vehicles. In this case, warning or alarms could be triggered in order to control the situation (e.g., nearby traffic lights could be programmed to switch to red, facilitating the smooth passage of these vehicles.).

In the following sections, we exploit our ODT approach for a key task typical of a traffic monitoring system within the smart city context, namely traffic prediction. We hence developed a use case in which it is addressed first the ODT modeling (namely the process leading to its signature definition) in Section 4.1, then its design in Section 4.2, and finally its deployment and performance evaluation in Section 4.3.

4 USE CASE: TRAFFIC PREDICTION THROUGH THE ODT APPROACH

To exemplify our approach, we present a Traffic Management System (TMS) based on the ODT of a road. The main goal is to predict road traffic using AI. In such direction, three steps are needed: 1) the modeling of the ODT (namely the process leading to its signature definition), 2) its design (i.e., hardware and software components to be interconnected and deployed), and 3) its performance evaluation (with a comparative analysis to assess the difference with respect to conventional cloud-based solution).

4.1 ODT Modeling: looking for signature

The key activity in the modeling of an ODT is assembling its signature. This is relevant in several applications domains, for instance, vehicles, vessels, and planes can be recognized by their radar signature while home appliances have recognizable electric signatures that can be used in order to determine when one of these devices is operating in the home. Obviously, signature comprehends features which are correlated with each other and, possibly, easy to be measured. Therefore, it is very important to disclose which are those important features that exhibit *correlation* with traffic, if they are easily *measurable* and if they serve a *functional* purpose in predicting road traffic itself. Historic traffic intensity data and the info about types of vehicles (cars, trucks, motorcycles, etc.) monitored in a certain area represent a baseline, however there is a room for improvement in traffic prediction. For example, features, such as air pollution and atmospheric variables, are highly correlated with traffic[6]. In many studies, traffic intensity data have been used to predict the concentration of air pollution [37, 40]. However, conversely, not much work has been done in the opposite direction.

To proceed with the approach, first we assessed the correlation between air pollution and road traffic by making use of the open data portal provided by Madrid City Council [7] featuring data from around 4000 road traffic sensors and 24 APM stations (Figure 4a). For this analysis, we used NO_2 as this air pollutant is highly associated with the traffic emission in the literature. NO_2 data were collected from one of the APM stations, situated at 105m distance from the chosen traffic intensity sensor (Figure 4b). These data were collected over the period of one month (from 01-05-2019 to 31-05-2019).

Figure 5 represents the Pearson correlation [14] between NO_2 and traffic flow for the month of May. A moderate to strong correlation can be seen between traffic and NO_2 during 23 days of the month, whereas during the rest of the 8 days, weak to very weak relationship has been observed. In order to analyze this weak relationship, we studied well-known Gaussian air pollutant dispersion model [1] and we found out that wind speed is one of the most important factors influencing the behavior of air pollution. Hence, to investigate if wind speed is affecting the correlation between air pollution and traffic, we performed the correlation analysis between wind speed and correlation values (r) of traffic and air pollutant NO_2. In Table 1, weak influence of wind speed on traffic-NO_2 correlation can be seen during the days when strong traffic-NO_2 correlation was observed. Contrary, when we performed the correlation analysis between wind speed and traffic-NO_2 correlation values for the days when weak correlation was observed, we found wind speed affecting the correlation significantly with the r value of -0.69, which states that greater the wind speed values, smaller the traffic-NO_2 correlation value and vice versa.

[6]https://naei.beis.gov.uk/resources/Primary_NO2_Emission_Factors_for_Road_Vehicles_NAEI_Base_2021_v3.pdf
[7]https://datos.madrid.es/portal/site/egob

Sensors.PNG


(a). Traffic sensors (blue) and APM stations (red)

(b). Chosen APM station (red) and traffic sensor (blue) for data collection

Fig. 4. Sensor Network in Madrid

Table 1. Correlation analysis of wind speed and r value of NO_2-Traffic correlation

Variable-1	Variable-2	r
Wind Speed during the days when strong correlation was observed between traffic and NO_2	Correlation value of air pollution and traffic during the days when strong correlation was observed between traffic and NO_2	0.32
Wind Speed during the days when weak correlation was observed between traffic and NO_2	Correlation value of air pollution and traffic during the days when weak correlation was observed between traffic and NO_2	-0.69

Many studies, such as [47], explain the inverse relationship between the wind speed and the concentration of air pollution. Which implies the fact that even if the traffic is high, sensors may observe low concentration of pollutants because wind speed has dispersed them, causing the absence of correlation between air pollutants and traffic. Other than that, in Figure 5, opposite to our expectations, we noticed a strong negative correlation between traffic and NO_2 on the day 11. In order to understand it, we plotted traffic and NO_2 graph for May 11 (Figure 6). It can be seen in the highlighted region of the figure that traffic and NO_2 patterns are quite opposite to our expectations. We observed an increase in NO_2 despite the reduction in traffic. Moreover, unlike other hours of the day, wind speed is not appearing to be triggering this unexpected behavior of NO_2. It brought us to the conclusion that there might be some other contributors of NO_2 that need to be taken into account. According to AECOM Greater London Authority report[8], domestic heating is the significant contributors of NO_2 in the air. We did not have household NO_2 emission data from Madrid available. However, as the domestic heating is inversely related with outer temperature (lower the temperature, highest the heating), therefore, to develop a hypothesis, we included temperature in our plot (Figure 7). In the shaded region, it is evident that during the time interval in which NO_2 showed an expected behavior with respect to road traffic, temperature, dropped

[8]https://www.london.gov.uk/sites/default/files/domestic_boiler_emission_testing_report.pdf

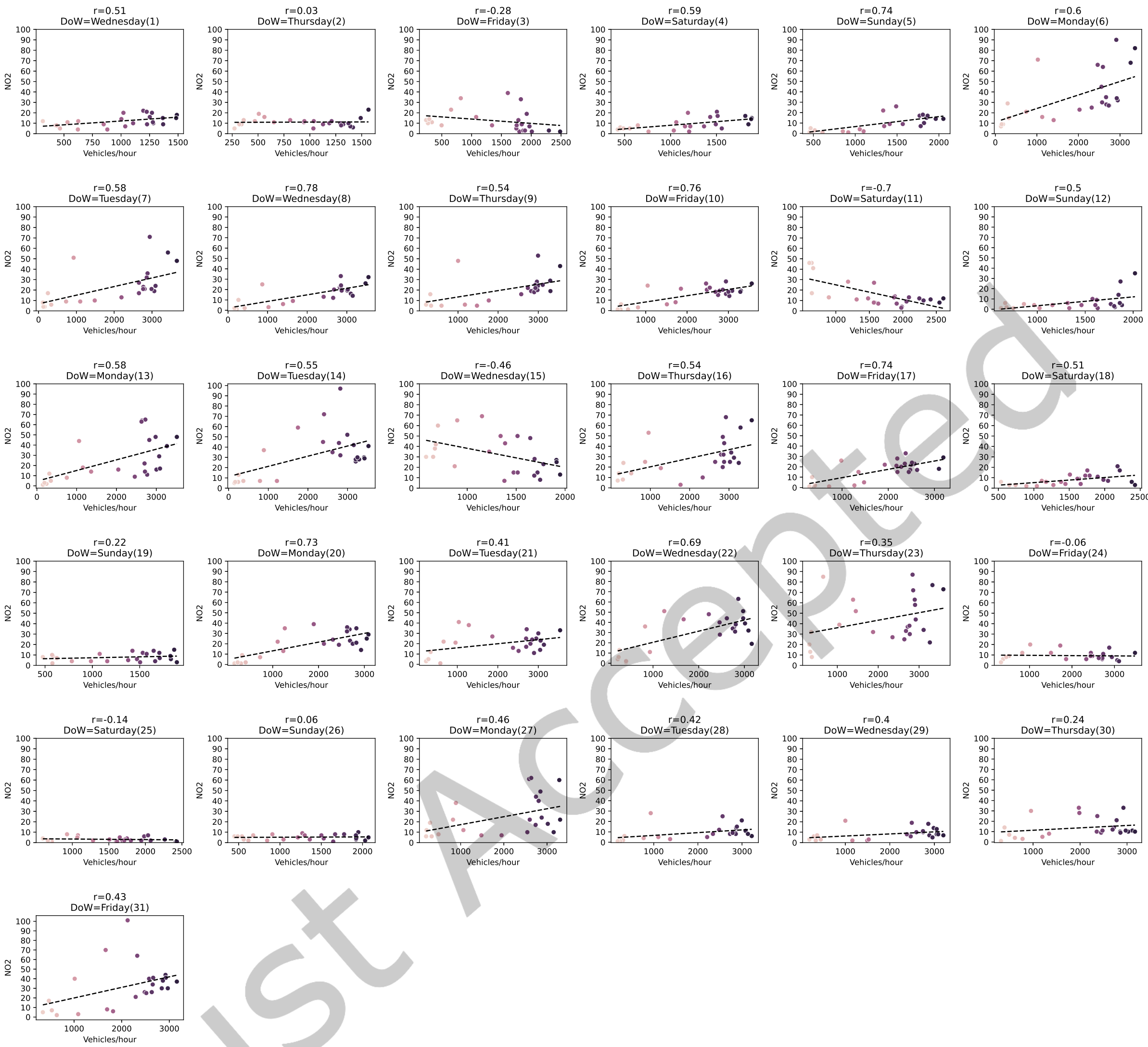

Fig. 5. Pearson correlation between traffic and NO_2 for May 2019

significantly. We hypothesize that the fall in temperature caused the domestic heating to turn on, which caused the peak in NO_2 concentration. To support our hypothesis, we computed the Pearson's correlation coefficient (r) value for NO_2-Temperature correlation, which was found to be -0.8. This indicates a highly significant negative correlation between temperature and NO_2 concentration.

4.1.1 Improving traffic prediction with pollution and meteorological features: Ascertained that winds speed and temperature are some of the most important factors that significantly affect the correlation between traffic and NO_2, the availability of well-formed data collected by the city of Madrid[7] let us verify the possibility of improving the prediction of traffic intensity by including pollution and meteorological features in the analysis

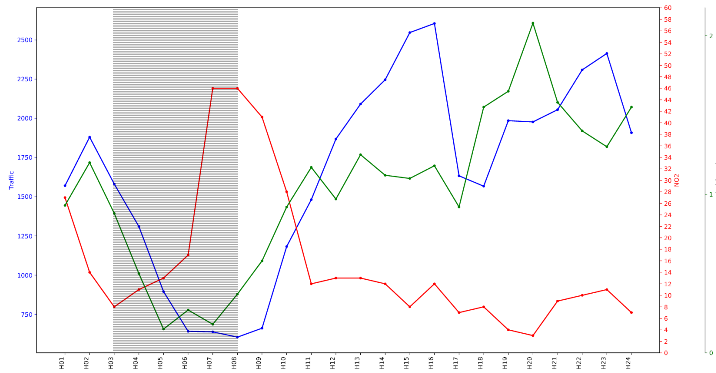

Fig. 6. Traffic, NO_2, and Wind Speed values with respect to different hours of the day on May 11

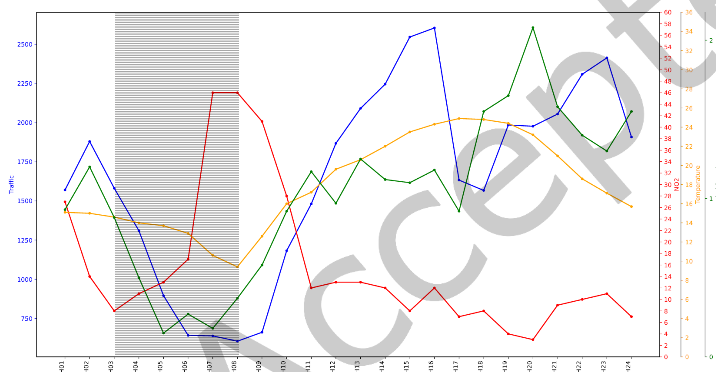

Fig. 7. Traffic, NO_2, Wind Speed, and Temperature values with respect to different hours of the day on May 11

(and, consequently, in the signature of our ODT). To this end, we developed an LSTM Recurrent Neural Network with 3 hidden layers, a dropout layer, and early stopping enabled (to keep the model from going into overfitting). Unlike conventional feedforward neural networks, the usage of memory units instead of hidden neurons helps the LSTM RNN to keep feedback connection and to use the sequence of n number of previous outputs to provide an output at time t. The choice of a so designed LSTM RNN is due to the fact that the nature of traffic, pollution, and meteorological data is time-series and, having the capability of considering previous time steps, LSTM RNN is a very popular choice for predictions that are based on the time-series data (further details about its architecture and configuration can be found in [4]).

In this work, the considered features for training the LSTM RNN were: a) traffic intensity data, collected from traffic sensors located near (less than 500 m) to an APM station; b) air pollution data, i.e., (CO, NO, NO_2, NO_x, O_3; and c) Atmospheric data, i.e, wind speed, wind direction, pressure, temperature. These data were collected over the period of 1 year (January 2019-December 2019), which consisted of nearly 9,000 records to be used in the experiment. Following the convention, 67% allocated for training purposes and the remaining 33% designated for testing. We compared the performance of the model with the performance of a baseline model, which was trained using only traffic time-series data. Mean absolute error and mean squared error were used as evaluation metrics. On comparing the performances, we found out that the addition of air pollutants and atmospheric data improved the forecasting accuracy average around 40% in terms of mean absolute error [4].

4.1.2 Improving traffic prediction with noise pollution: As an extension of our previous work [4], we trained the LSTM RNN model using noise pollution level and traffic time-series data. Hourly traffic and noise pollution data were collected from a traffic sensor and a noise pollution sensor, situated at around 25m distance from the traffic sensor. Following the previous work, we compared the performance of the model with a baseline model which was trained with only traffic time-series data, and we achieved around 13.48% improvement in terms of mean absolute error [5]. Unfortunately, since the two datasets were not integrable, noise and air pollutants have not been considered simultaneously with atmospheric data.

To wrap-up, ascertained that traffic time-series data can be empowered by correlated phenomena (measurable with low cost and general-purpose sensory infrastructure) such as NO_2, winds speed, and temperature for enabling an accurate traffic prediction, we decided to include them in the signature of our ODT (see Figure 8), whose development process is reported next.

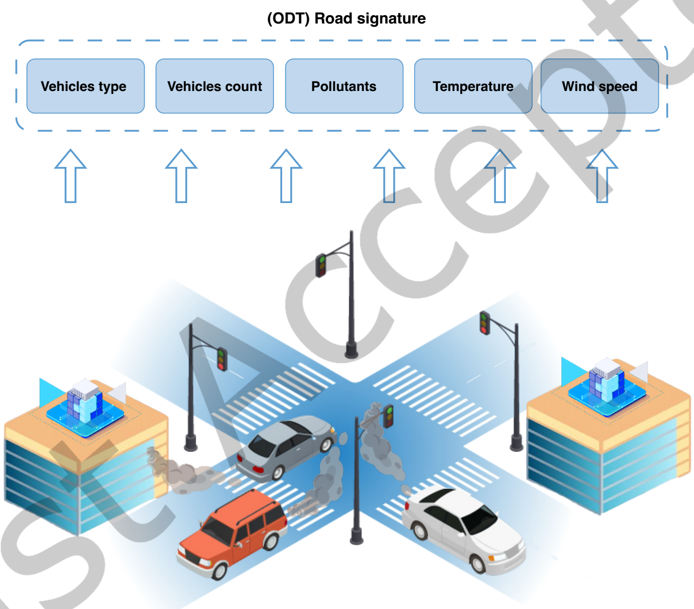

Fig. 8. Modeled ODT Road and its signature

4.2 ODT Development

In order to concretely develop a TMS carrying out traffic prediction, the ODT demands for an EI system able to gather traffic, pollution and meteorological data from the environment, to process them with low-latency and low-cost communication technology, and to store relative information in the database to ensure persistence of data for further analysis. The proposed system consists of several hardware components (shown in Figure 9) that work together to achieve the desired goal:

- one general-purpose board: we opted for a *Google Coral Dev Board*, whose TPU performs up to 4 trillion operations per second [9] and, hence, is particularly suitable for applying even complex EI techniques at the network edge;
- two sensor boards: we opted for *Gravity MEMS Gas Sensor Board* for gathering pollutants data and Google Environmental Sensor Board for gathering environmental telemetries;
- one camera module: it could be either *Google Coral Camera* or a normal USB Camera for gathering real time video streams about the vehicles passing by the observed road;
- one microcontroller unit (MCU): we opted for a *ESP32 WROOM 32D Board* for interfacing with the Gas Sensor Board.

Data from ESP32 and Coral Dev board are forwarded toward an *MQTT Broker* (implemented using Eclipse Mosquitto[10]) to the core module of the system, in the form of structured data. This has been implemented using *Node-red*[11] for wiring between hardware devices and software modules (dispatching function) as well as for managing rules, thresholds, and conditions which define the backend logic of the system (controlling function). In details, the Node-red module can *(i)* either fetch or update DTs status information via the REST API of *Eclipse Ditto*[12], which is encapsulated in a *Docker*[13] container along with its software dependencies; *(ii)* manage data persistence through two different NoSQL databases, i.e., *MongoDB*[14] for the DT status and *InfluxDB*[15] for telemetries; and *(iii)* support data visualization by making use of *Grafana*[16]'s user-friendly dashboards. An overview of these software modules with their deployment is depicted in Figure 10. In particular, it should be noted that the Google Coral Dev Board and its Coral Camera, the Environmental Sensor Board as well as the Gas Sensor board and the ESP32, are deployed at the edge of the network in line with the EI principles, to gather the video and real-time telemetries needed directly from the field and there apply real-time inference with AI algorithms. However, as for the other modules of backend logic (Node-red), communication services (MQTT Broker), and data storage (InfluxDB, MongoDB) and visualization (Grafana), they all could be distributed outside the deep Edge, for example on the Cloud, thus keeping the same operation but with a notable impact on the overall performance, as discussed in the following.

4.3 Performance Evaluation

EI allows processing data locally, i.e., without sending them to the remote servers, with several benefits in terms of latency, bandwidth consumption and energy dissipation, as well as with implication in terms of cost, privacy, and security. Therefore, to quantify how EI could impact on our TMS and, particularly, on the time-sensitive task of vehicle recognition, we first evaluate the performance (in terms of accuracy and latency) of different DL models in Section 4.3.1 and, then, we provide a comparison with respect to a conventional cloud-based solution in Section .

4.3.1 Models performance test. After the survey of the available models for vehicle recognition, we found that the most promising ones are:

[9] https://coral.ai/docs/edgetpu/benchmarks/
[10] https://mosquitto.org/
[11] https://nodered.org/
[12] https://www.eclipse.org/ditto/
[13] https://www.docker.com/
[14] https://www.mongodb.com/
[15] https://www.influxdata.com/
[16] https://grafana.com/

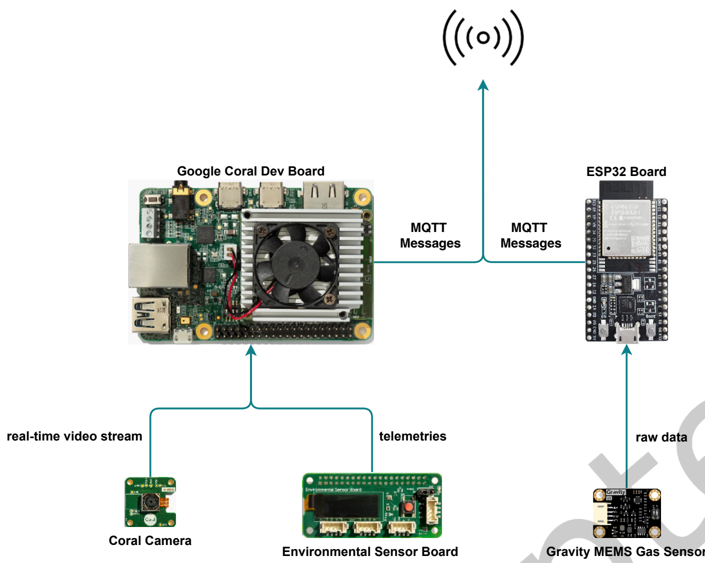

Fig. 9. Overview of the hardware components of the ODT-based TMS

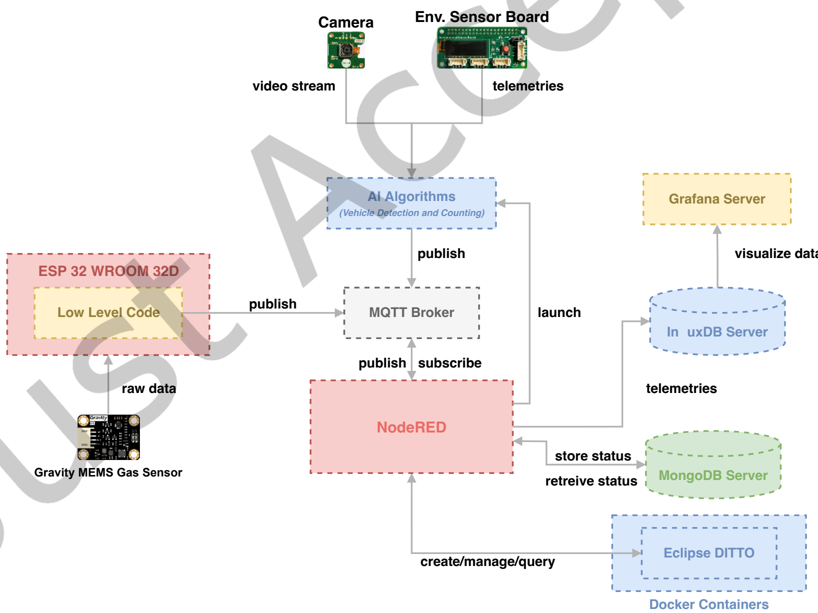

Fig. 10. Overview of the software modules of the ODT-based TMS

- Three pre-trained models taken from TensorFlow Hub [17]], strongly suggested by Google Coral's community, i.e., *SSD MobileNetV1*, *SSD MobileNetV2* and *SSDLite MobileDet* [43].

[17]https://tfhub.dev/s?deployment-format=lite&module-type=image-object-detection

- Two models trained with custom Datasets using Transfer Learning technique on EfficientDet architecture [41] through TensorFlow Lite Model Maker [18]: *(i)* MTD Model, trained using *Mini Traffic Detection* (MTD) dataset[19] and *(ii)* TI Model, trained using *Traffic Images* (TI) dataset[20].

In order to provide an accurate analysis, each model has been tested over our Google Coral Dev Board with the same video stream and its performance measured in terms of *mean average precision* (*mAP*) and latency. The *mAP* is the primary metric according to COCO evaluation metrics [27] and is based on *(i)* Confusion Matrix; *(ii)* Intersection over Union (IoU); *(iii)* Recall; and *(iv)* Precision. Each model is assigned its own *mAP* which represents the accuracy score averaged over all detection's categories. The term latency, instead, refers to the time required for just one inference: the higher this value, the slower is the overall inference applied by the model. The testing session summarized in Figure 11 pointed out that, in our case, models' accuracy and latency are inversely proportional: if the accuracy increases, the latency decreases and vice versa.

The pre-trained models perform very fast, both with real-time and recorded videos. In fact, none of the three exceed 30ms per frame as inference time. The fastest one, according to the tests taken, is *SSD MobileNet V1*, which has a latency of 12.6ms. On the other hand, whatever they earn in latency, they lose it in terms of accuracy. In fact, all the three models didn't perform very well in detecting classes, sometimes they weren't able to detect the object even if it was on the frame. Conversely, while the re-trained *MTD Model* and *TI Model* exhibit higher precision, they may sacrifice some degree of responsiveness: indeed, both the re-trained models have almost the same latency (around 70 ms per frame), which more than the double of the lowest pre-trained model. However, the neural network with the highest *mAP* between the two of them is the *MTD Model* (85.5%), trained using zoomed images of vehicles (*MTD dataset*[19]) and, therefore, more suited for situations in which the camera is not too far from vehicles.

Providing the best trade-off between accuracy and latency, we have decided to use the MTD Model for our TMS, and a further performance analysis of the overall system is reported as follows.

4.3.2 Performance comparison with Cloud-based deployment. In order to point out the added value produced by the EI-based deployment based on the Google Dev Coral Board, the chosen *MTD Model* has been tested with the same video stream on a typical Cloud instance (i.e., Intel® Core™ i7-6820HQ, 2.70GHz x 8), aiming to compare the performance of these two configurations. The results achieved provide very interesting insights from different viewpoints, and are reported in Table 2. First, with respect to inference time per frame, the Edge TPU is 1,6 times faster than the Cloud's CPU (*70.40ms vs. 112.32 ms*) thanks to the ML accelerator of the Google Coral Board. But it also notably improves the whole system performances in terms of power consumption (the slower the model, the more is the power dissipated by the device) and memory saving (by reducing the model size). Beyond guaranteeing a lower time, the EI implementation avoids additional overhead coming from the networking: if using a cloud-centric approach for computer vision tasks (such as the vehicle detection one), the frames captured from the real-time video stream at the edge need to be forwarded to the cloud as input for the inference task. In our case, each frame captured by the camera module through a Python algorithm has an approximate file size of *1,936Mb*. Such a data traffic impact on the network latency for around *43ms* for frame (exploiting the Coral Dev Board 802.11ac interface at 2.4GHz for a greater coverage, the realistic data rate is up to 450Mbps[9]) but, it simultaneously affects the power dissipation of the edge device and, more importantly, the bandwidth consumption. Indeed, taking into account one-day interaction between edge device and cloud server with a 20fps (frame per second) bit rate, 1,729,000 frames would be eventually sent, for a total of *3.34Tb* of daily data traffic. Conversely, if we locally process the frame and forward only results to a remote server through an MQTT message, a small packet of *67 bytes* (including packet ID, control headers and payload) is needed, four orders of

[18]https://www.tensorflow.org/lite/models/modify/model_maker
[19]https://www.kaggle.com/datasets/zoltanszekely/mini-traffic-detection-dataset
[20]https://universe.roboflow.com/traffic/label-yeebg

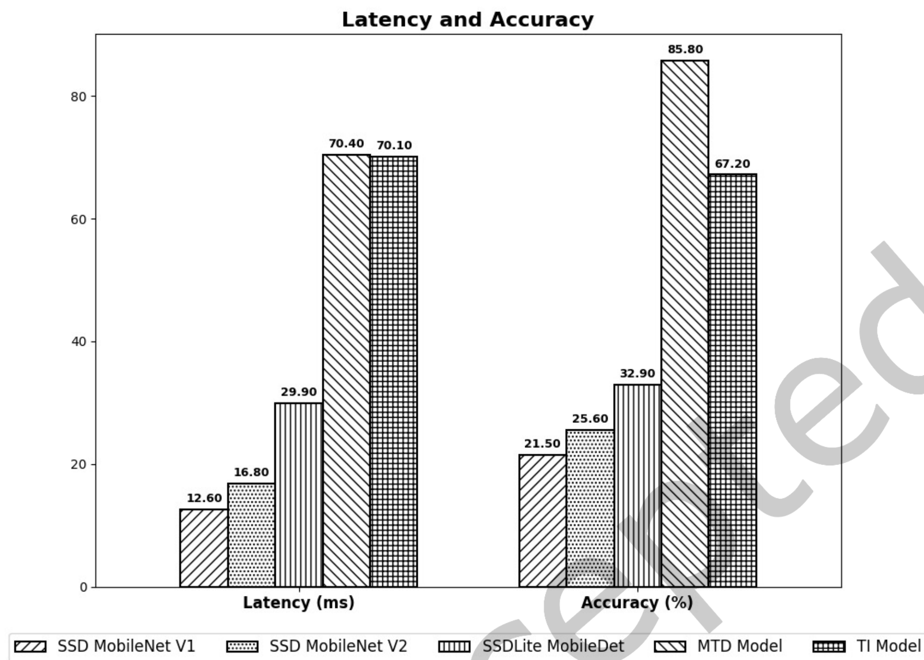

Fig. 11. Benchmark in terms of Latency and Accuracy of both pre-trained and re-trained models

Table 2. Performance Evaluation: Edge- vs Cloud-based deployment

	Edge	**Cloud**
Model Latency (per inference)	70.40ms	112.32ms
Network Latency (per frame)	0	34ms
Data Traffic (per frame)	0.54Kb	1.936Mb
Data Traffic (per day)	0.872Gb	3.34Tb

magnitude lower with respect to the previous case, thus saving a huge data traffic every day (*only 0.872Gb vs. 3.34Tb*).

Summarizing, the benefits coming from the edge-based deployments are manifold (latency, bandwidth, energy, privacy, etc.) and the continuous advances in the field of general-purpose devices equipped with AI accelerators promise to further push for an always wider adoption of EI, with positive impacts on those approaches, exactly like the ODT, which aims to support the service provision locally and in a data-driven manner.

5 CONCLUSION AND FUTURE WORKS

The need of infrastructures focused on specific sensing tasks and powerful enough for promptly processing big amount of data hinder the development of key IoT services, especially those particularly complex and large scaled typical of the smart cities. In this direction, we have presented the definition and the first implementation of

ODT, a novel concept at the confluence of DT, synthetic sensing and EI aimed to simplify the (re)engineering of large-scale distributed smart systems by moving complexity from hardware infrastructure to software layer. We have first introduced the key building blocks of our approach which is highly innovative since bottom-up, data-driven and multidisciplinary; then, we have presented a use case related to a ODT-based TMS focused on traffic prediction, built from the preliminary modelling to the implementation and the performance evaluation. The obtained results through real data are indeed very positive, and the overall approach promises to be more efficient than the state-of-the-art (mainly consisting in conventional cloud-based solutions).

In line with the limitations and challenges discussed in Section 3.2.1, we propose a threefold research direction for future work: *(i)* enrich the current ODT with all the functionalities presented in Section 3.3, for example by providing the ODT with additional event detection capabilities (e.g., traffic anomalies); *(ii)* develop additional large-scale use cases in other IoT scenarios beyond the smart city domain, where it has been successfully exploited, i.e., the industry in the context of the Horizon Europe MLSysOps [21]; and *(iii)* finally, provide a systematization of the development practices presented in this work under the form of a full-fledged methodology. In order to fully use and move the ODT to real deployment, there is also the need for a systematic study about the security and privacy issues that this approach may have. The issues can range from the introduction of false or fake signatures to trick or mislead the results, to the possibility of determining specific behavioral patterns of a single person. In addition, anomalies can be purposely injected into the system in order to derive wrong output and reactions. In many cases, these security issues can be similar to those that IoT system cope with in the deployment scenarios. The major security concerns are those related to the injection of signatures that do not really represent the actual behavior of real objects. The "camouflage effect" could be considered another important challenge to the approach.

REFERENCES

[1] Adel A Abdel-Rahman. 2008. On the atmospheric dispersion and Gaussian plume model. In *Proceedings of the 2nd International Conference on Waste Management, Water Pollution, Air Pollution, Indoor Climate, Corfu, Greece*, Vol. 26.

[2] Fayez Alharbi, Lahcen Ouarbya, and Jamie A Ward. 2020. Synthetic Sensor Data for Human Activity Recognition. In *2020 International Joint Conference on Neural Networks (IJCNN)*. 1–9. https://doi.org/10.1109/IJCNN48605.2020.9206624

[3] Gianluca Aloi, Giuseppe Caliciuri, Giancarlo Fortino, Raffaele Gravina, Pasquale Pace, Wilma Russo, and Claudio Savaglio. 2017. Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. *Journal of Network and Computer Applications* 81 (2017), 74–84.

[4] Faraz Malik Awan, Roberto Minerva, and Noel Crespi. 2020. Improving road traffic forecasting using air pollution and atmospheric data: Experiments based on LSTM recurrent neural networks. *Sensors* 20, 13 (2020), 3749.

[5] Faraz Malik Awan, Roberto Minerva, and Noel Crespi. 2021. Using Noise Pollution Data for Traffic Prediction in Smart Cities: Experiments Based on LSTM Recurrent Neural Networks. *IEEE Sensors Journal* 21, 18 (2021), 20722–20729.

[6] Vincenzo Barbuto, Claudio Savaglio, Min Chen, and Giancarlo Fortino. 2023. Disclosing edge intelligence: A systematic meta-survey. *Big Data and Cognitive Computing* 7, 1 (2023), 44.

[7] Chris Beckmann, Sunny Consolvo, and Anthony LaMarca. 2004. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. In *International Conference on Ubiquitous Computing*. Springer, 107–124.

[8] Roberto Casadei, Giancarlo Fortino, Danilo Pianini, Wilma Russo, Claudio Savaglio, and Mirko Viroli. 2019. A development approach for collective opportunistic Edge-of-Things services. *Information Sciences* 498 (2019), 154–169.

[9] Tahiya Chowdhury, Murtadha Aldeer, Justin Yu, Joseph Florentine, Amber Haynes, and Jorge Ortiz. 2021. Is General Purpose Sensing a Pipe Dream? A Case Study in Ambient Multi-sensing for Human Activity Recognition. In *Proceedings of the First International Workshop on Cyber-Physical-Human System Design and Implementation*. 21–26.

[10] Flavio Cirillo, Gürkan Solmaz, Everton Luís Berz, Martin Bauer, Bin Cheng, and Ernoe Kovacs. 2019. A standard-based open source IoT platform: FIWARE. *IEEE Internet of Things Magazine* 2, 3 (2019), 12–18.

[11] Javier Conde, Andres Munoz-Arcentales, Alvaro Alonso, Sonsoles Lopez-Pernas, and Joaquin Salvachua. 2021. Modeling digital twin data and architecture: A building guide with fiware as enabling technology. *IEEE Internet Computing* (2021).

[21] https:/mlsysops.eu/

[12] Marco Conti, Silvia Giordano, Martin May, and Andrea Passarella. 2010. From opportunistic networks to opportunistic computing. *IEEE Communications Magazine* 48, 9 (2010), 126–139.
[13] Flavio Corradini, Arianna Fedeli, Fabrizio Fornari, Andrea Polini, and Barbara Re. 2022. DTMN a modelling notation for digital twins. In *International Conference on Enterprise Design, Operations, and Computing*. Springer, 63–78.
[14] Gualter Couto and Ricardo Faria. 2010. PSI-20 fluctuation: correlation of the portuguese stock market with major global capital markets. *CEEAplA-A-Working Paper Series* (2010), 1–18.
[15] Shuiguang Deng, Hailiang Zhao, Weijia Fang, Jianwei Yin, Schahram Dustdar, and Albert Y Zomaya. 2020. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal* 7, 8 (2020), 7457–7469.
[16] Haya Elayan, Moayad Aloqaily, and Mohsen Guizani. 2021. Digital twin for intelligent context-aware IoT healthcare systems. *IEEE Internet of Things Journal* 8, 23 (2021), 16749–16757.
[17] Laith M Fawzi, Sideeq Y Ameen, Salih M Alqaraawi, and Shefa A Dawwd. 2018. Embedded real-time video surveillance system based on multi-sensor and visual tracking. *Appl. Math. Infor. Sci* 12 (2018), 345–359.
[18] Giancarlo Fortino and Claudio Savaglio. 2023. *Integration of Digital Twins & Internet of Things*. Springer International Publishing, Cham, 205–225. https://doi.org/10.1007/978-3-031-21343-4_8
[19] Steffen Fritz, Linda See, Tyler Carlson, Mordechai Muki Haklay, Jessie L Oliver, Dilek Fraisl, Rosy Mondardini, Martin Brocklehurst, Lea A Shanley, Sven Schade, et al. 2019. Citizen science and the United Nations sustainable development goals. *Nature Sustainability* 2, 10 (2019), 922–930.
[20] Edward Glaessgen and David Stargel. 2012. The digital twin paradigm for future NASA and US Air Force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*. 1818.
[21] Michael Grieves. 2014. Digital twin: manufacturing excellence through virtual factory replication. *White paper* 1, 2014 (2014), 1–7.
[22] Gerhard P Hancke, Bruno de Carvalho e Silva, and Gerhard P Hancke Jr. 2012. The role of advanced sensing in smart cities. *Sensors* 13, 1 (2012), 393–425.
[23] A Hughes. 2018. Forging the Digital Twin in Discrete Manufacturing, A Vision for Unity in the Virtual and Real Worlds. *LNS Research e-book* (2018).
[24] Werner Kritzinger, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. 2018. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* 51, 11 (2018), 1016–1022.
[25] Sathish AP Kumar, R Madhumathi, Pethuru Raj Chelliah, Lei Tao, and Shangguang Wang. 2018. A novel digital twin-centric approach for driver intention prediction and traffic congestion avoidance. *Journal of Reliable Intelligent Environments* 4, 4 (2018), 199–209.
[26] Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic Sensors: Towards General-Purpose Sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3986–3999. https://doi.org/10.1145/3025453.3025773
[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
[28] Zhihan Lv, Dongliang Chen, Ranran Lou, and Qingjun Wang. 2021. Intelligent edge computing based on machine learning for smart city. *Future Generation Computer Systems* 115 (2021), 90–99.
[29] Javier Mendez, Kay Bierzynski, MP Cuéllar, and Diego P Morales. 2022. Edge Intelligence: Concepts, Architectures, Applications, and Future Directions. *ACM Transactions on Embedded Computing Systems (TECS)* 21, 5 (2022), 1–41.
[30] Dejan Milojicic. 2020. The edge-to-cloud continuum. *Computer* 53, 11 (2020), 16–25.
[31] Roberto Minerva, Faraz Malik Awan, and Noel Crespi. 2021. Exploiting digital twin as enablers for synthetic sensing. *IEEE Internet Computing* (2021).
[32] Roberto Minerva, Gyu Myoung Lee, and Noel Crespi. 2020. Digital twin in the IoT context: a survey on technical features, scenarios, and architectural models. *Proc. IEEE* 108, 10 (2020), 1785–1824.
[33] Skyler Norgaard, Ramyar Saeedi, Keyvan Sasani, and Assefaw H. Gebremedhin. 2018. Synthetic Sensor Data Generation for Health Applications: A Supervised Deep Learning Approach. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 1164–1167. https://doi.org/10.1109/EMBC.2018.8512470
[34] Jamie O'Sullivan, Dominic O'Sullivan, and Ken Bruton. 2020. A case-study in the introduction of a digital twin in a large-scale smart manufacturing facility. *Procedia Manufacturing* 51 (2020), 1523–1530.
[35] A Parrot and L Warshaw. 2019. Industry 4.0 and Digital Twin, Manufacturing Meets Its Match 2017. *Retrieved on* 10 (2019).
[36] Barbara Re. 2023. DTMN a Modelling Notation for Digital Twins. In *Enterprise Design, Operations, and Computing. EDOC 2022 Workshops: IDAMS, SoEA4EE, TEAR, EDOC Forum, Demonstrations Track and Doctoral Consortium, Bozen-Bolzano, Italy, October 4–7, 2022, Revised Selected Papers*, Vol. 466. Springer Nature, 63.
[37] Mats Rosenlund, Francesco Forastiere, Massimo Stafoggia, Daniela Porta, Mara Perucci, Andrea Ranzi, Fabio Nussio, and Carlo A Perucci. 2008. Comparison of regression models with land-use and emissions data to predict the spatial distribution of traffic-related air pollution in Rome. *Journal of exposure science & environmental epidemiology* 18, 2 (2008), 192–199.

[38] Andrey Rudskoy, Igor Ilin, and Andrey Prokhorov. 2021. Digital Twins in the Intelligent Transport Systems. *Transportation Research Procedia* 54 (2021), 927–935. https://doi.org/10.1016/j.trpro.2021.02.152 International Scientific Siberian Transport Forum - TransSiberia 2020.

[39] Samad ME Sepasgozar. 2021. Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment. *Buildings* 11, 4 (2021), 151.

[40] Jason G Su, Michael Jerrett, Bernardo Beckerman, Michelle Wilhelm, Jo Kay Ghosh, and Beate Ritz. 2009. Predicting traffic-related air pollution in Los Angeles using a distance decay regression selection strategy. *Environmental research* 109, 6 (2009), 657–670.

[41] Mingxing Tan, Ruoming Pang, and Quoc V Le. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10781–10790.

[42] Alina Trifan, Maryse Oliveira, José Luís Oliveira, et al. 2019. Passive sensing of health outcomes through smartphones: systematic review of current solutions and possible limitations. *JMIR mHealth and uHealth* 7, 8 (2019), e12649.

[43] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. 2021. Mobiledets: Searching for object detection architectures for mobile accelerators. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3825–3834.

[44] Mike Zhou, Jianfeng Yan, and Donghao Feng. 2019. Digital twin framework and its application to power grid online analysis. *CSEE Journal of Power and Energy Systems* 5, 3 (2019), 391–398. https://doi.org/10.17775/CSEEJPES.2018.01460

[45] Xiaokang Zhou, Wei Liang, Ke Yan, Weimin Li, I Kevin, Kai Wang, Jianhua Ma, and Qun Jin. 2022. Edge-enabled two-stage scheduling based on deep reinforcement learning for internet of everything. *IEEE Internet of Things Journal* 10, 4 (2022), 3295–3304.

[46] Xiaokang Zhou, Xuesong Xu, Wei Liang, Zhi Zeng, Shohei Shimizu, Laurence T Yang, and Qun Jin. 2021. Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics* 18, 2 (2021), 1377–1386.

[47] Ying Zhou and Jonathan I Levy. 2007. Factors influencing the spatial extent of mobile source air pollution impacts: a meta-analysis. *BMC public health* 7, 1 (2007), 1–11.

[48] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. 2019. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* 107, 8 (2019), 1738–1762.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009