University of Ottawa

PhD Thesis

Human-Centered Urban Intelligence through Context-Aware Recommendation Using IoT and Digital Twins

By

Saeed Alharthi

Supervisor

Prof. Abdulmotaleb El-Saddik

A thesis presented to the University of Ottawa in fulfillment of the thesis requirements for the degree of Doctorate of Philosophy in Computer Science

uOttawa

School of Electrical Engineering and Computer Science

Faculty of Engineering

Abstract

Rapid urbanization has increased the need for intelligent urban systems that can support context-sensitive decisions for individuals operating under changing environmental and infrastructural conditions. This thesis addresses the lack of an integrated operational framework that connects Internet of Things sensing, artificial intelligence, and Digital Twin representations for personalized services in smart urban environments. The research is motivated by the fragmentation of existing smart-city systems, in which sensing, analytics, recommendation, and simulation are often developed as separate components and are commonly evaluated mainly through accuracy-oriented metrics.

To address this gap, the thesis proposes a unified framework that combines IoT-based context acquisition, structured context modeling, context-aware predictive and recommender models, and interoperable Digital Twins of users, points of interest, and urban environments. The framework transforms heterogeneous urban data into multi-dimensional contextual representations spanning environmental, spatial, temporal, and activity-related factors. It employs a hybrid wide-and-deep learning architecture to support prediction and recommendation under dynamic conditions. Within the proposed pipeline, Digital Twins extend recommendations beyond immediate ranking by enabling simulation-based reasoning over alternative actions and their anticipated consequences.

The evaluation combines offline model assessment with scenario-based and productivity-oriented analysis. Under the reported evaluation setting, the proposed context-aware wide-and-deep model achieved the strongest predictive performance among the evaluated models, attaining an RMSE of 0.3695 and explaining approximately 81.9% of the variance in observed preference scores. Compared with matrix factorization, neural collaborative filtering, and collaborative filtering baselines, the proposed model reduced prediction error and captured contextual variation more effectively. The broader system-level analysis further indicates that contextual recommendation and Digital Twin-based reasoning can support recommendation relevance, activity planning, decision interpretation, and alignment between user goals and available urban resources. The thesis also identifies important constraints related to data quality, sensing coverage, privacy, scalability, and transferability across urban contexts. Overall, the work shows that personalization in smart cities can be framed and assessed as an integrated, human-centered decision-support problem rather than as an isolated recommendation task.

Contents

List of Figures

List of Tables

List of Abbreviations

AI Artificial Intelligence

API Application Programming Interface

CF Collaborative Filtering

DT Digital Twin

EV Explained Variance

IoT Internet of Things

MAE Mean Absolute Error

MSE Mean Squared Error

PoI Point of Interest

PRISMA Preferred Reporting Items for Systematic Reviews and Meta-Analyses

PT Physical Twin

R^2 Coefficient of Determination

RMSE Root Mean Squared Error

W&D Wide-and-Deep

Chapter 1

Introduction

1.1 Motivation

Rapid urbanization has transformed cities into complex socio-technical environments in which transportation, energy, healthcare, education, and public services are increasingly intertwined with digital infrastructure. While this transformation creates economic and social opportunities, it also intensifies challenges related to efficiency, sustainability, and individual productivity. As urban populations grow and daily activities become more constrained by congestion, environmental conditions, and resource limitations, there is an increasing need for intelligent systems that support context-aware decision-making [12, 26, 60].

Smart-city initiatives have sought to address these pressures through Internet of Things (IoT) infrastructures, data analytics, and automation. Distributed sensors can monitor traffic, air quality, energy use, and environmental conditions, enabling operational awareness and optimization [7]. However, many implementations remain infrastructure-centric and organized around isolated services such as transportation or energy management, providing limited support for adaptive and human-centered decision-making [4, 80]. A related limitation appears in recommender and decision-support systems, which often rely on predefined rules, historical averages, or static user profiles. In mobility settings, for example, recommendations commonly privilege shortest paths or historical usage patterns while overlooking transient factors such as environmental comfort, noise exposure, personal constraints, or situational intent [32, 83, 104]. As a result, operational efficiency does not necessarily translate into recommendations that support user well-being and productivity.

This matters because environmental conditions are not peripheral to decision quality. Empirical studies show that noise, temperature, lighting, and air quality influence concentration, comfort, and task effectiveness [65, 90]. At the same time, advances in sensing technologies have made continuous collection of environmental and behavioral data increasingly feasible through embedded urban devices and personal wearables [78]. The challenge is therefore not only to sense urban

conditions, but also to interpret them in ways that remain meaningful for individual users in real time.

Digital Twin (DT) technology offers a promising basis for this shift. A DT is a dynamic digital model of a physical entity that is continuously updated through data exchange with its real-world counterpart [35]. In smart-city research, DTs have mainly been applied to infrastructure management and urban planning, particularly for simulation and predictive analysis [73, 92]. More recent work extends the concept toward human-centered DTs that represent individual behavior, context, and interaction with the environment [110]. This extension is important because it creates the possibility of evaluating alternative actions before they are taken rather than only monitoring conditions after the fact.

Despite progress in IoT sensing, recommender systems, and DT research, these strands remain weakly integrated. IoT infrastructures primarily support data acquisition, recommender systems often depend on abstract interaction data, and many DTs continue to emphasize infrastructure performance rather than human experience [7, 60, 80, 104]. Recent studies also highlight the fragmentation of smart-city recommender systems and the lack of unified architectures capable of reasoning across heterogeneous data sources at city scale [4, 81]. The motivation for this thesis, therefore lies in the need for a more integrated and human-centered approach to urban intelligence, one that accounts for individual context, comfort, and productivity while combining IoT-derived contextual data, adaptive intelligence, and DT-based reasoning for context-aware and personalized decision-making in urban environments [35, 83].

1.2 Problem Statement

The central problem addressed in this thesis is the absence of an integrated operational framework that connects IoT sensing, AI-driven intelligence, and DT representations for personalized services in smart urban environments. Although smart urban systems increasingly rely on sensing infrastructures, data analytics, and intelligent services, these layers are still commonly developed and deployed as separate components. IoT systems primarily support data acquisition, artificial intelligence (AI) models often operate as stand-alone analytical modules, and DTs are typically used for simulation or visualization rather than adaptive decision support [30, 60].

This separation limits how context is represented and used in practice. Most smart-city recommender systems still operate on static or periodically updated datasets rather than continuous context streams. Even when context-aware methods include variables such as time or location, these variables are usually treated as isolated features rather than as parts of an evolving model of the city and its inhabitants [2, 81]. Consequently, recommendations are generated without coherent

representations of how individual behavior, urban dynamics, and system interventions interact over time.

A related problem concerns the role of DTs and personalization models. City-scale DTs support planning, monitoring, and simulation of urban infrastructure and environmental processes [30, 92], but they are rarely embedded in operational recommender systems. In parallel, personalization in smart-city services remains fragmented across domains such as mobility, tourism, energy management, and healthcare, where systems rely on domain-specific user models and features [13, 52]. This prevents knowledge transfer across services and limits the development of coherent city-wide intelligence. Personalization also remains largely reactive, adapting to observed behavior without explicitly modeling user intent, goals, or productivity constraints. The limited integration of Human DTs intensifies this problem because, although human-centered digital representations have been explored in healthcare and activity monitoring, they are rarely embedded within broader smart-city intelligence systems [68]. Without structured models of individuals, systems cannot adequately represent how recommendations influence behavior, how behavioral change affects sensed data, or how these interactions evolve.

The current evaluation practice further compounds the problem. Smart-city recommender systems are commonly assessed using predictive accuracy metrics such as RMSE, precision, recall, or ranking measures [112]. While these metrics are useful for measuring recommendation quality, they provide limited insight into broader outcomes such as productivity, cognitive load, or urban efficiency. From a computational urban science perspective, this leaves unclear whether intelligent urban systems actually support meaningful decision-making and behavioral adaptation in real-world settings [55].

Accordingly, the problem is not only one of technical fragmentation but also one of inadequate evaluation. Addressing it requires a unified framework that connects IoT-based context acquisition, AI-driven prediction and recommendation, and interoperable DTs of humans and urban environments, while also supporting evaluation approaches that assess predictive performance and productivity-oriented decision support under clearly defined experimental and scenario-based conditions.

1.3 Research Objectives

This research aims to design and evaluate a DT-driven framework that integrates IoT sensing and AI models to provide context-aware recommendations and productivity-oriented decision support in smart urban environments. Existing smart-environment and recommender systems frequently rely on static user profiles, limited contextual awareness, or isolated components. The objective of

this thesis is therefore to establish a coherent pipeline that links sensing, modeling, simulation, and recommendation.

The first objective is to develop an IoT-based context acquisition layer for capturing heterogeneous data from users, points-of-interest (PoI), and urban environments. This includes environmental variables such as temperature, humidity, air quality, noise levels, and ultraviolet exposure, together with user-related signals obtained through personal devices. Continuous and localized sensing is necessary because reliable context-aware services depend on environmental and situational observations that many existing systems treat only coarsely or inconsistently [6, 78].

The second objective is to design AI-driven predictive and recommender models that use the collected contextual information together with historical user preferences and behavior. Unlike traditional recommender approaches that emphasize similarity or proximity, the intended models incorporate contextual dimensions directly within the learning process so that recommendations remain responsive to changing urban conditions [1, 24].

The third objective is to model and integrate DTs for users, PoIs, and urban environments within a unified virtual environment. In this framework, the human DT represents user preferences, behavior, and contextual states, while PoI and city DTs represent environmental and urban conditions. Continuous data exchange among these entities supports simulation of potential user-environment interactions before decisions occur in the physical world [35, 45].

The fourth objective is to evaluate the framework in terms of predictive performance and productivity-oriented decision support. This objective focuses on whether contextual sensing, predictive modeling, and DT-based scenario reasoning can support more interpretable and contextually appropriate decisions, using offline predictive metrics and scenario-based decision-support criteria rather than direct measurement of long-term productivity or well-being.

1.4 Research Questions

Previous sections identified key limitations in current smart urban environments, including fragmented system architectures, limited adaptive intelligence, and the absence of evaluation frameworks that capture human productivity and decision quality. Although prior research has examined IoT infrastructures, context-aware recommender systems, and DT technologies, these components are typically studied in isolation. As a result, the literature lacks cohesive frameworks that integrate sensing, intelligence, and digital representations into a unified personalization pipeline [60, 80]. To address these gaps, this thesis formulates three research questions that guide the design, implementation, and evaluation of an integrated human-centered smart-city framework.

RQ1: Does the proposed IoT–AI–DT framework enable the integration of sensing, context modeling, recommendation, and scenario reasoning into a unified decision-support pipeline for personalized smart-urban services?

Current smart-city systems often implement sensing, analytics, recommendation, and simulation layers as loosely connected components, limiting coherent responses to dynamic urban conditions [78]. This question evaluates whether the proposed architecture can operationally connect heterogeneous IoT data acquisition, structured context modeling, predictive recommendation, DT synchronization, and scenario-based reasoning within a single decision-support pipeline. It is addressed through the formal system definition, layered architecture, data-flow specification, and integrated workflow developed in the thesis.

RQ2: Does incorporating IoT-derived environmental, spatial, temporal, activity, and user-context features into a wide-and-deep recommender model improve predictive accuracy compared with collaborative filtering, matrix factorization, and neural collaborative filtering baselines?

Although context-aware recommender systems can improve recommendation relevance, their application within large-scale urban environments remains limited [2, 112]. This question tests whether structured IoT-derived context improves predictive performance when incorporated directly into the learning model. It is evaluated by comparing the proposed context-aware wide-and-deep model against collaborative filtering, matrix factorization, and neural collaborative filtering baselines using repeated experimental runs and predictive accuracy metrics such as RMSE, MAE, R^2, and explained variance.

RQ3: Does DT-based scenario reasoning improve recommendation-based decision support beyond predictive accuracy alone, as measured by contextual relevance, feasibility, access burden, activity-planning support, and productivity-oriented alignment?

Evaluations of smart-city systems commonly focus on technical performance metrics such as accuracy or latency, while decision-support quality and productivity-oriented outcomes receive less attention [67]. This question evaluates whether DT-based scenario reasoning adds decision-support value beyond model prediction scores alone. It is assessed through scenario-based analysis of candidate recommendations using criteria such as contextual relevance, feasibility, access burden, activity-planning support, and alignment between user goals and available urban resources.

Together, these research questions define the scope of the thesis by aligning architectural integration, predictive modeling, and scenario-based evaluation with the goal of adaptive and human-centered personalization in smart urban environments.

1.5 Contributions

This thesis makes three integrated contributions to the design of human-centered urban intelligence systems by unifying IoT-based sensing, context-aware AI, and DT reasoning within a single decision-support framework. Rather than treating data acquisition, recommendation, and simulation as separate problems, the thesis positions them as interdependent components of one urban intelligence pipeline. The novelty of the work lies not in introducing these technologies independently, since each has been studied in prior literature, but in combining them into a coherent architecture for context-aware and productivity-oriented urban decision support.

1.5.1 Unified IoT–AI–DT Architecture

The first contribution is a unified IoT–AI–DT architecture for smart urban environments. Existing literature has examined IoT sensing for urban monitoring, AI models for prediction and recommendation, and DTs for simulation and city management, but these strands are typically developed as separate layers or applied in isolation [23, 30, 60, 92]. In contrast, this thesis proposes an end-to-end architecture in which heterogeneous IoT streams are transformed into structured context, consumed by AI models for predictive and recommender intelligence, and then propagated into DT entities representing users, PoI, and urban areas.

What is new relative to the literature is this explicit architectural unification of sensing, learning, and simulation within a continuous feedback loop. Prior work generally treats IoT as a data source, AI as an analytical component, and DTs as a visualization or scenario-analysis layer. This thesis instead defines their operational integration as a single pipeline for adaptive personalization, where physical sensing updates digital representations, AI outputs inform twin states, and DT reasoning supports downstream decision support. The contribution is therefore not merely that these components coexist, but that they are structured as one interoperable architecture for human-centered urban intelligence.

1.5.2 Context-Aware Wide-and-Deep Model

The second contribution is a context-aware wide-and-deep model for prediction and recommendation in smart urban environments. Prior recommendation literature has established the value of hybrid and deep learning approaches, including wide-and-deep architectures, for combining memorization and generalization in large-scale recommendation settings [28, 51, 112]. However, such models are rarely formulated around IoT-derived urban context as a primary input space, and they are not typically developed to operate on structured representations spanning environmental, spatial, temporal, and user-related dimensions.

The novelty in this thesis lies in adapting the wide-and-deep paradigm to context-aware urban intelligence rather than conventional platform recommendation. The proposed model learns from heterogeneous contextual inputs derived from IoT sensing and context modeling, including environmental conditions, spatial and temporal situations, user state, and PoI-related attributes. In this way, context is not treated as a secondary filter or auxiliary modifier, but as an integral part of the predictive and recommendation function itself. This distinguishes the proposed model from much of the existing literature, where contextual variables are either limited to simple dimensions such as time and location or applied outside the main learning architecture [2]. The contribution is therefore a context-aware wide-and-deep learning formulation tailored to dynamic urban settings and to the decision-support objectives of the thesis.

1.5.3 Digital Twin-Based Decision Simulation

The third contribution is a DT-based decision simulation layer that extends recommendations beyond ranking by enabling scenario-based reasoning over alternative actions. Existing DT literature has demonstrated the value of virtual representations for monitoring, planning, and simulation in urban systems. In contrast, recommender-system literature has focused primarily on ranking relevant items or actions [30, 68, 92]. What remains underdeveloped in the literature is the integration of DT reasoning directly into personalized recommendation workflows.

This thesis contributes a DT-based decision simulation mechanism in which Human DTs, city or area twins, and PoI twins interact to evaluate possible courses of action before they are enacted in the physical world. What is new here is that the DT layer is not treated as a passive representation or a detached urban dashboard. Instead, it functions as an active reasoning layer that supports scenario exploration, comparison of alternatives, and evaluation of likely consequences under current contextual conditions. This extends the role of recommendation from identifying relevant options to supporting simulated decision analysis in a human-centered urban setting.

The contribution therefore lies in repositioning DTs as part of the recommendation logic itself. In the proposed framework, DT simulation complements AI-based ranking by enabling context-sensitive exploration of actions and their anticipated effects on user experience and urban conditions. This provides a form of decision support that goes beyond conventional recommender outputs and is specifically aligned with the thesis objective of productivity-aware urban intelligence [80, 85].

Taken together, the scientific contribution is an integrated decision-support formulation in which IoT-derived context improves prediction, while Digital Twin representations use that context to evaluate recommended alternatives through scenario-based reasoning.

1.6 Publications Arising from This Thesis

Parts of the research presented in this thesis were disseminated through the following peer-reviewed publications:

1. Alharthi, S., Karime, A., and El-Saddik, A. "Advancing Urban Intelligence Through Digital Twins in Smart Cities." In *2025 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pp. 1–6, 2025. doi: 10.1109/SmartNets65254.2025.11106802.

2. Alharthi, S., and El-Saddik, A. "Digital Twin-Enhanced Smart Environments: An IoT-AI Framework for Contextual Personalization." In *2026 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, pp. 1–6, 2026. doi: 10.1109/ACDSA67686.2026.11467934.

3. Alharthi, S., and El-Saddik, A. "Personalization in smart urban environments: a taxonomy and survey of recommender systems." *Computational Urban Science*, vol. 6, no. 1, article 8, 2026. doi: 10.1007/s43762-026-00239-8.

These publications informed the development of the thesis framework, literature review, DT modeling, and context-aware personalization approach.

1.7 Thesis Structure

This thesis is organized into eight chapters that progressively develop the conceptual foundations, system design, implementation, and evaluation of an integrated personalization framework for smart urban environments. The structure connects theoretical motivation, technical components, and empirical analysis into a coherent research narrative.

Chapter 1 introduces the research context and motivation, highlighting the growing complexity of urban environments and the limitations of static smart-city systems in supporting human-centered productivity. It formulates the central research problem concerning the absence of integrated IoT, AI, and DT pipelines for personalized urban services. The chapter also defines the research objectives, research questions, and thesis contributions.

Chapter 2 reviews the background and related work relevant to the thesis. It discusses smart-city concepts, IoT technologies, context-aware systems, DT paradigms, and recommender systems in smart environments. The chapter focuses on concepts and limitations that directly inform the proposed framework and concludes by identifying the research gaps addressed in this thesis.

Chapter 3 presents the overall architecture of the proposed system and its associated data flow. It describes the end-to-end pipeline linking urban sensing with digital intelligence and feedback

mechanisms, including IoT data sources, preprocessing processes, AI models, and DT components. The chapter also outlines system assumptions and constraints related to scalability, data availability, latency, privacy, and governance.

Chapter 4 examines the IoT framework and context modeling layer. It describes the architecture used for context acquisition in smart environments and the modeling of environmental, user, spatial, activity-related, and temporal context. The chapter also discusses data preprocessing strategies for heterogeneous and incomplete data and highlights deployment challenges related to real-world sensing infrastructure.

Chapter 5 focuses on AI models for context-aware prediction and recommendation. It presents the problem formulation, feature representation, model architecture, training configuration, and baseline models used to evaluate the proposed context-aware wide-and-deep approach.

Chapter 6 introduces the DT modeling and simulation layer. It defines the DT entities considered in the thesis, including Human, City or Area, and PoI twins. It explains how these entities interact to support data exchange, scenario simulation, and decision support. Ethical and practical considerations such as privacy, trust, synchronization, and model fidelity are also discussed.

Chapter 7 provides an integrated evaluation of the proposed framework. It reports the training and evaluation methodology, quantitative comparison with baseline and ablation models, statistical validation, scenario-based evaluation, productivity-oriented analysis, and limitations of the reported evidence.

Finally, Chapter 8 concludes the thesis by summarizing the main contributions and findings, identifying research limitations, and outlining directions for future research and potential real-world deployment.

This thesis argues that human-centered urban intelligence requires more than isolated advances in sensing, prediction, or simulation. It requires an integrated framework in which IoT-based context acquisition, context-aware AI models, and DT representations operate as a unified decision-support pipeline for personalized urban services. The central claim is that when heterogeneous urban data are transformed into structured contextual representations and coupled with predictive recommendation and DT-based scenario reasoning, smart-city systems can move beyond static, infrastructure-centered optimization toward adaptive and productivity-aware support for individual decision-making under changing urban conditions.

Chapter 2

Background and Related Work

2.1 Literature Search and Study Selection

The literature review followed a PRISMA-style selection process based on searches in IEEE Xplore, ACM Digital Library, and Google Scholar, supplemented by backward and forward citation tracking. The search focused on recommender systems, IoT, urban computing, and context-aware services, with the final peer-reviewed corpus comprising 98 primary studies. The identification, screening, eligibility, and inclusion process is retained in Fig. 2.1, while additional cited works provide broader conceptual and methodological context for the chapter.

2.2 Smart Urban Intelligence and IoT-Derived Context

2.2.1 Smart Urban Intelligence as a Data-to-Decision Pipeline

Smart cities are commonly framed as cyber-physical systems that integrate sensing infrastructures, communication networks, and computational intelligence to improve efficiency, sustainability, and quality of life. Within this view, IoT enables continuous data acquisition from distributed urban assets, while urban intelligence extends digitization by adding interpretation, reasoning, and decision-support capabilities [23, 60, 105]. Urban intelligence therefore depends not only on collecting data, but also on transforming heterogeneous observations into actionable knowledge through cycles of observation, analysis, and intervention [23].

However, much prior work remains concentrated on data acquisition rather than on integrated data-to-decision pipelines. IoT infrastructures provide fine-grained observations across mobility, environmental conditions, infrastructure, and human activity. However, they also introduce challenges related to data quality, synchronization, semantic interoperability, scale, noise, missing values, and heterogeneous spatial and temporal resolutions [11, 23, 60, 78, 102]. Data availability alone therefore does not produce actionable intelligence. The resulting gap is the weak integration of sensing, data

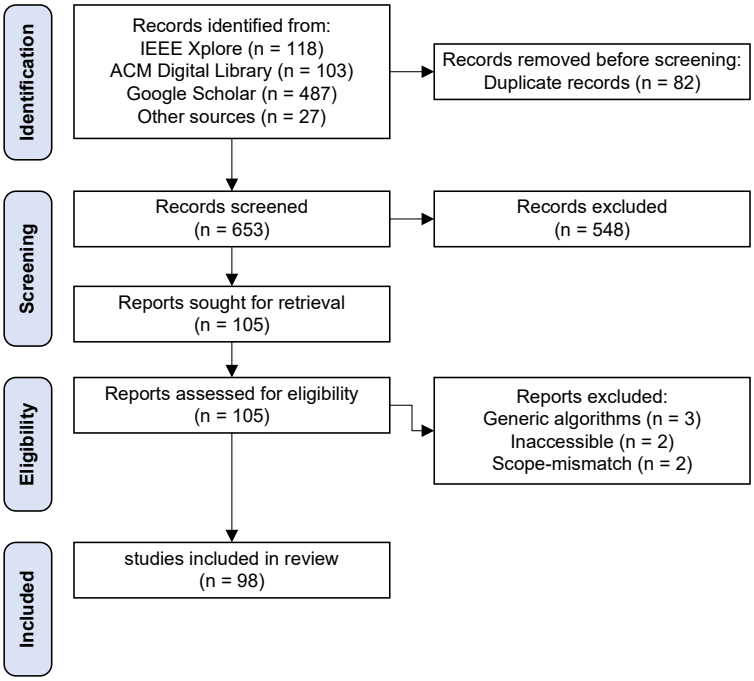

Figure 2.1: PRISMA flow diagram showing study identification, screening, eligibility, and inclusion for smart-city recommender systems

management, context modeling, analytics, and decision support into coherent end-to-end systems. This gap motivates the thesis to focus on a unified pipeline in which IoT sensing is treated as the input to structured context modeling, AI-driven recommendation, and DT-based decision support rather than as an isolated data-acquisition layer.

2.2.2 IoT-Derived Context in Urban Environments

IoT sensing becomes valuable for smart urban intelligence when raw observations are interpreted as context for adaptive system behavior. Context-aware computing allows applications to adapt according to situational information, where context refers to information that characterizes the situation of an entity [22]. In smart-city environments, this includes dynamic conditions such as location, time, activity, environmental state, and user-related behavior. Context-aware recommender systems extend traditional recommendations by incorporating such situational variables into the decision process rather than relying only on historical interactions [2].

Urban context is multidimensional and commonly includes spatial, temporal, environmental, and activity-related dimensions [11]. Spatial context captures location-dependent variation, temporal context represents time-dependent patterns, environmental context reflects physical conditions, and

activity-related context describes user behavior [2, 102]. IoT infrastructures expand the granularity and scope of these contextual signals by supporting persistent data acquisition from environmental stations, traffic sensors, smartphones, wearables, and other connected devices [78, 102]. However, sensor data are not equivalent to context. Raw observations require interpretation, aggregation, abstraction, and alignment before they can support learning, recommendation, or decision-making.

Existing context-aware systems are commonly described through a life cycle of acquisition, modeling, reasoning, and dissemination [78]. Acquisition collects raw data from IoT and digital sources, modeling structures observations for computation, reasoning derives higher-level interpretations, and dissemination makes inferred knowledge available to applications [11, 78]. Although this life-cycle view is useful, many implementations remain loosely coupled and do not specify how these stages should operate within an integrated urban intelligence framework. As a result, context awareness often remains partial, domain-specific, or auxiliary to system design rather than serving as the operational bridge between IoT sensing and downstream decision support [1, 113]. This gap motivates the structured context-modeling layer developed in this thesis, where environmental, spatial, temporal, activity-related, and user-related observations are transformed into unified representations for recommendation and DT synchronization.

2.2.3 Context Management, Real-Time Processing, and Edge-Cloud Support

The transformation of IoT streams into usable context requires context management mechanisms that can handle device heterogeneity, data aggregation, context abstraction, interoperability, and scalable delivery to applications [78]. Middleware can decouple sensing and application layers and support flexible smart-city architectures [60]. However, middleware mainly facilitates abstraction and data flow. It does not by itself provide learning, reasoning, or decision-making. This creates a gap between context management infrastructures and intelligent systems capable of using contextual information for adaptive services.

Real-time operation further complicates context-aware urban intelligence. IoT-enabled smart cities generate continuous high-velocity streams that require streaming-oriented processing rather than delayed batch analysis [38]. Timeliness is important because delayed context inference reduces the relevance of adaptive services and weakens decision support under changing urban conditions [23]. Streaming platforms support incremental processing and continuous context updates as new data arrive [38], but they also introduce trade-offs between responsiveness and analytical depth. Asynchronous data arrival, latency, and synchronization problems can reduce inference reliability when heterogeneous sources are combined.

Edge-cloud architectures address some of these challenges by distributing computation across local and centralized resources. Edge computing enables preliminary processing near data sources to

reduce latency and network load, while cloud infrastructures support storage and advanced analytics [23, 62]. This hierarchy can balance local responsiveness with global integration. Nevertheless, existing approaches often distribute computation without integrating intelligence across layers. Edge and cloud components are frequently treated as separate processing units, leaving unresolved the need for coherent end-to-end context modeling and reasoning. This thesis therefore treats context management, streaming updates, and edge-cloud processing as supporting mechanisms within a larger intelligence pipeline rather than as independent infrastructure concerns.

2.2.4 Decision Support and Governance Constraints

The value of smart urban intelligence lies in its ability to support decisions, not merely to process data. Decision-support systems translate analytical outputs into guidance for users, service providers, or urban administrators. In contrast, recommender systems provide personalized and context-sensitive suggestions across mobility, energy, and public services [5, 27, 80]. Human-centered urban intelligence also requires attention to daily experience, productivity, and well-being, since environmental conditions can affect human performance and comfort [111].

Existing decision-support mechanisms remain largely limited to ranking or prediction. They rarely allow users or decision-makers to explore alternatives, validate outputs, or evaluate likely consequences before action. This creates a gap between predictive intelligence and decision-oriented reasoning. DTs provide dynamic representations that can support simulation and prediction in urban environments [30, 92], but their integration with recommendation frameworks remains limited and often confined to visualization or monitoring. This limitation motivates the need to connect IoT-derived context, AI-driven recommendation, and DT-based reasoning within a unified decision-support pipeline.

Intelligent urban systems also raise governance challenges related to accountability, transparency, privacy, policy alignment, fairness, and inclusivity. Continuous spatio-temporal data collection requires safeguards such as federated learning and decentralized analytics. At the same time, biased data and models may reinforce existing inequalities [17, 42, 81]. Although these concerns are widely acknowledged, current research often separates governance, technical design, and evaluation. Many studies focus on sensing, analytics, or recommendation as individual components, while integrated smart-city recommender systems remain difficult to validate at scale [52, 80]. This fragmentation supports the positioning of this thesis, which treats urban intelligence as an integrated, human-centered pipeline connecting IoT sensing, structured context modeling, AI-driven recommendation, and DT-based reasoning.

2.3 Digital Twins for Urban Decision Support

Digital Twins provide a conceptual and technical basis for representing physical systems as synchronized digital counterparts that support monitoring, analysis, simulation, and decision making. Unlike static digital models, a DT remains connected to its real-world counterpart through ongoing data exchange. In urban contexts, Deng et al. define a DT city as a cyber-physical system in which urban entities, processes, and environments are digitally mirrored to support analysis, simulation, and governance-oriented decision making [30]. This framing positions DTs as operational decision-support systems rather than as visualization artifacts alone. Their defining characteristics include continuous synchronization, model-based abstraction, scenario simulation, and feedback between physical and digital domains [56, 92].

City-scale DTs extend this paradigm from individual assets to complex urban systems. They integrate infrastructure, mobility, environmental processes, land use, and public services within cyber-physical representations that can support spatial analysis, operational monitoring, and scenario evaluation [30, 92]. In these systems, IoT devices provide the sensing basis for synchronization by collecting data from infrastructure, mobile devices, and environmental monitoring systems [60]. Data ingestion and management layers then aggregate heterogeneous real-time and historical data, while urban data pipelines must address scalability, heterogeneity, and temporal alignment [23]. Streaming platforms support real-time processing under latency constraints [38], and edge-cloud architectures can distribute storage and computation to balance responsiveness with analytical capacity [97]. In emergency and infrastructure management, synchronized urban representations can improve situational awareness, risk assessment, and response coordination by allowing alternative interventions to be evaluated before action in the physical system [25].

Human-centric DTs shift the focus from infrastructure and assets to individuals. Laamarti et al. define a Human DT as a virtual entity that mirrors physical and behavioral attributes through continuous data from wearables, sensors, and personal information systems [68]. Such systems can use wearable devices, smartphones, and ambient IoT infrastructure to support monitoring, state estimation, and longitudinal analysis [49]. This human-centered perspective is important for smart urban intelligence because personalized decision support depends not only on the state of infrastructure, but also on user preferences, activities, constraints, and surrounding environmental conditions. AI-based DTs further strengthen this capability by using machine learning to infer latent states, forecast behavior, and update internal representations over time [56]. Deep learning is especially relevant in smart-city settings because it can process high-volume, high-velocity, and heterogeneous urban data [23].

Digital Twins are also applied in smart environments, buildings, energy systems, and urban

utilities, where they model asset behavior, predict performance, and support operational optimization [77, 94]. These applications show the value of combining sensing, modeling, prediction, and intervention. However, they often remain tied to specific assets, sectors, or operational objectives rather than to personalized urban decision support [52].

However, the literature also shows that DTs do not automatically produce reliable or human-centered decision support. City-scale DTs are often constrained by data heterogeneity, uneven data quality, limited model fidelity, and difficulties in integrating diverse urban subsystems [30]. Scholars also warn that urban DTs may oversimplify social processes and create a false sense of control unless uncertainty, assumptions, and institutional constraints are made explicit [44, 71]. Human-centric DTs intensify concerns about privacy, consent, data ownership, and misuse because they rely on continuous monitoring of personal or behavioral data [68]. AI-driven DTs add further concerns about explainability, robustness, and transferability, since data-driven models may improve prediction while obscuring how their outputs are produced or failing when moved to new contexts [56]. These concerns connect directly to broader requirements for transparency, accountability, and respect for human autonomy in AI-supported decision contexts [40].

A further limitation is that existing DT applications are often organized around infrastructure, visualization, monitoring, or domain-specific simulation rather than embedded recommender workflows. City-scale DTs are strong at representing infrastructure and system-level processes, but they provide limited support for personalized, context-aware decision making. Human-centric DTs capture individual states and behavior, but they often remain domain-specific and are not fully integrated with city-scale representations. Smart-environment and energy-oriented DTs support operational optimization, but they are frequently limited to particular assets, buildings, districts, or utility domains [52, 77]. As a result, the literature demonstrates the value of DT reasoning. However, it does not fully connect infrastructure context, human context, AI recommendation, and scenario-based decision support within one operational pipeline.

The resulting gap is that existing DT research does not fully integrate real-time IoT context, AI-driven recommendation, and human-centered scenario reasoning into a single productivity-oriented decision-support framework. DTs are often used to observe, visualize, forecast, or simulate urban systems, while recommender systems typically rank options without using synchronized virtual representations to reason about consequences before action. This separation limits their ability to support decisions that depend simultaneously on changing environmental conditions, user state, activity goals, spatial constraints, and anticipated outcomes.

This thesis addresses this gap by treating the DT as an active decision-support layer within an integrated IoT–AI–DT framework. In the proposed approach, IoT sensing provides real-time environmental and contextual data, AI models generate predictions and recommendations, and DT

representations support scenario-based evaluation beyond ranking alone. The contribution is therefore not the use of DTs as a standalone simulation or visualization tool, but their integration into a productivity-oriented recommender workflow where user context, PoI context, and urban context can be evaluated together before decisions are enacted. This framing positions DTs as advisory and context-aware components of human-centered urban intelligence rather than as autonomous control mechanisms.

2.4 Context-Aware Recommendation in Smart Environments

Recommender systems provide a personalization and decision-support layer in smart environments by translating heterogeneous urban data into user-relevant guidance across domains such as mobility, energy, healthcare, governance, and public services [23, 80]. In smart-city settings, their role extends beyond information filtering because recommendations may influence movement, energy use, healthcare actions, or access to public services, creating feedback between digital intelligence and physical urban processes [80, 81]. Traditional recommender paradigms, including collaborative filtering, content-based filtering, hybrid systems, and deep learning, each contribute useful mechanisms for personalization. Collaborative filtering learns from user-item interactions and is effective when interaction data are dense [13, 66, 87], while content-based and knowledge-based methods use item descriptions, semantic features, and user profiles when interaction histories are limited [27, 70]. Hybrid recommenders combine collaborative, content-based, contextual, and social signals to reduce sparsity and cold start [3, 18, 21, 80]. Deep learning further extends these approaches by learning nonlinear and multimodal patterns from interactions, text, sensor streams, and graphs [23, 112]. Neural collaborative filtering and graph-based models provide examples of this shift toward learned interaction and relational representations [51, 79].

However, these paradigms remain limited when deployed in dynamic smart environments. Collaborative filtering is weakened by cold start, sparsity, and changing city-scale interactions, especially in domains such as energy management [52]. Content-based and knowledge-based methods depend on accurate semantic descriptions and can over-specialize [48]. Hybrid and deep models improve robustness and representation learning. However, they also increase complexity, computational cost, and opacity without necessarily solving real-time sensing, DT synchronization, feedback, or productivity-oriented evaluation. As a result, existing methods are often recommendation-centric rather than system-centric: they improve ranking or prediction but do not consistently connect sensing quality, context construction, AI inference, scenario reasoning, and human-centered outcomes.

Context awareness addresses part of this limitation by incorporating situational variables into the recommendation process. Traditional collaborative and content-based filtering rely mainly on historical interactions or item descriptions, which are insufficient when relevance depends on time, location, activity, device state, environmental conditions, or social setting [1]. Context-aware recommender systems extend the utility function from $u : U \times I \rightarrow \mathbb{R}$ to $u : U \times I \times C \rightarrow \mathbb{R}$, where C represents contextual dimensions [2]. Context can be incorporated through pre-filtering, post-filtering, or contextual modeling [1]. More recent approaches use tensor factorization, embeddings, and deep learning to represent user-item-context interactions [50, 82, 112], while sensor-rich and mobile environments continue to expand the scope of context-aware recommendation in smart environments and healthcare [86].

The limitation is that context is often treated as an added feature, filter, or post-processing condition rather than as part of a continuously updated operational representation. In smart environments, context is dynamic, multidimensional, and dependent on acquisition, representation, reasoning, and lifecycle management mechanisms [11]. IoT infrastructures expand contextual granularity through sensor streams from environmental sensors, smart meters, mobile devices, wearables, and embedded systems [60, 78]. Nevertheless, raw sensor data cannot directly support recommendations. It requires sensing, preprocessing, aggregation, temporal alignment, and semantic enrichment before it becomes usable for recommendation tasks [78]. IoT integration also introduces heterogeneity, uncertainty, synchronization problems, sampling-rate differences, and spatial coverage limitations that may weaken recommendation reliability [102]. Scalable and real-time processing through streaming infrastructures, edge computing, and hybrid edge-cloud architectures can reduce latency and support incremental feature extraction [38, 62, 115]. However, scalable processing alone does not create coherent decision support unless context quality, model inference, DT synchronization, and downstream evaluation are integrated.

Privacy, fairness, transparency, and trust further constrain smart-environment recommendations because these systems may process sensitive behavioral, health, and location data and may influence access to mobility, energy, healthcare, or public services. Federated learning can reduce exposure by keeping raw data local, although communication cost, non-IID data, and convergence remain challenges [17, 100]. Blockchain-based recommender architectures offer secure and auditable coordination, but remain experimental and face scalability and energy limitations [53]. Fairness-aware and explainable recommendations are also needed because biased or opaque systems may reinforce unequal access to urban opportunities [29, 34]. These concerns are important, but in this thesis they are treated as architectural and evaluative constraints rather than as separate research directions. The central gap in the reviewed literature is the limited development of recommender architectures that connect IoT-derived context, AI-based recommendation, DT reasoning, and productivity-oriented

evaluation within one operational pipeline.

Table 2.1 sharpens this gap by comparing representative smart-environment recommendation frameworks against the four elements required by this thesis: IoT support, DT integration, environmental context, and human productivity. The comparison shows that prior work may include IoT sensing, environmental context, or productivity-related outcomes, but generally lacks explicit DT integration and does not combine all four elements within a single framework.

Table 2.1: Comparison of recommendation frameworks in smart environments

Cited Work (Authors, Year)	**IoT**	**DT**	**Environmental Context**	**Productivity-Related Outcome**
Bagci & Karagoz (2015) [8]	No	No	Yes	No
Bouhissi et al. (2023) [15]	Yes	No	Partial	Yes (faster access to services)
Casino et al. (2017) [20]	Yes	No	Yes	Yes (time-efficient healthy routes)
Cheng et al. (2016) [24]	No	No	Not inherent	No
Covington et al. (2016) [28]	No	No	Not inherent	No
Di Martino & Rossi (2016) [32]	Yes	No	Indirect	Yes (reduced travel time)
Liu et al. (2021) [69]	Yes	No	Yes	Yes (faster trips, better mode choice)
Ojagh et al. (2020) [76]	Yes	No	Yes	No
Rafique et al. (2023) [81]	Yes	No	Yes	Yes (service delivery efficiency)
Sayed et al. (2021) [89]	Yes	No	Partial	No
Shambour & Lu (2011) [93]	No	No	No	Yes (decision-time reduction)
Unger et al. (2020) [103]	No	No	Yes	No
Wu et al. (2022) [108]	Yes	No	Yes	Yes (congestion/pollution mitigation)

The resulting research gap is therefore not simply limited recommendation accuracy. The sharper limitation is fragmentation across sensing, context modeling, recommendation, simulation, and evaluation. Existing systems may support context-aware ranking, scalable processing, privacy-preserving learning, or application-specific recommendation, but they rarely treat recommendation as part of a closed-loop urban intelligence pipeline [52, 80, 81]. This thesis addresses that gap by positioning recommendations within an integrated IoT–AI–DT framework. In the proposed approach, IoT-derived environmental, spatial, temporal, user, and PoI-related signals support context-aware AI recommendations. At the same time, DT reasoning extends recommendations beyond immediate ranking toward scenario-based decision support and productivity-oriented evaluation.

2.5 Research Gaps and Positioning

Despite advances in IoT sensing, context-aware recommendation, and DT research, existing smart-city systems still provide limited support for adaptive and human-centered urban intelligence. The reviewed literature shows four related gaps that motivate the framework developed in this thesis.

2.5.1 Gap 1: Fragmented IoT, AI, and DT Pipelines

Smart-city intelligence architectures remain fragmented across sensing, analytics, recommendation, and simulation layers. IoT research emphasizes data acquisition and context management, but often gives limited attention to how sensed data supports personalized decision-making [78]. Recommender-system research focuses on prediction and ranking performance while often abstracting from sensing reliability, real-world context acquisition, and operational deployment [80]. DT research prioritizes visualization, simulation, and governance, frequently treating AI-driven intelligence as an external input rather than as an adaptive component of the twin [30, 92]. As a result, IoT infrastructures, analytics platforms, recommender systems, and DTs are often developed as loosely coupled systems, limiting closed-loop intelligence and operational relevance [10, 44, 61].

2.5.2 Gap 2: Shallow and Static Context Representation

Context-aware recommender systems incorporate factors such as time, location, and environmental conditions into personalization [1]. However, many approaches still rely on static, discretized, or weakly updated context representations, such as predefined time slots, categorical states, or limited environmental descriptors [72, 104]. This limits their responsiveness to changing user intent, dynamic urban conditions, and real-time IoT streams. Recent surveys therefore call for stronger integration of behavioral signals, contextual data, and adaptive reasoning, yet such approaches remain uncommon in broader smart-city intelligence infrastructures [81].

2.5.3 Gap 3: Recommendation Without Scenario-Based Reasoning

Smart-city recommender systems commonly generate ranked lists of services, activities, or PoIs, but they rarely support scenario-based reasoning over the likely consequences of alternative actions. This limits their value for urban decision support because recommendations are often evaluated as immediate prediction outputs rather than as actions that interact with users, places, environmental conditions, and urban systems over time. DTs are increasingly viewed as core smart-city infrastructures [92], but recommender systems and learning-based personalization are seldom embedded as

architectural components within them [41]. Consequently, DTs without adaptive intelligence and human-in-the-loop feedback have limited practical and policy value [44, 71].

2.5.4 Gap 4: Weak Productivity-Oriented Evaluation

Smart-city evaluation remains dominated by technical metrics such as prediction accuracy, latency, throughput, precision, recall, and RMSE. These metrics are useful for benchmarking but provide limited evidence of effects on human behavior, productivity, well-being, and lived urban experience [55, 81]. Human productivity provides a relevant human-centered outcome because it links urban intelligence to task completion, value creation, and time use [33]. While human-computer interaction commonly evaluates productivity through task completion time, error rate, and efficiency [19]. Environmental studies further show that temperature, ventilation, and noise affect cognitive performance and work output [36, 91, 107]. However, productivity gains in smart-city systems are often assumed rather than measured and remain weakly connected to observable behavioral or environmental indicators. Although some data-driven urban interventions have shown improvements in mobility efficiency and well-being [85], such evaluations remain uncommon and are not yet consistently connected to productivity-oriented assessment. This supports calls for human-centered and impact-aware evaluation beyond purely algorithmic measures [10]. Similar critiques warn that DTs without human-centered validation risk losing institutional and societal relevance [71].

Taken together, these gaps define the contribution structure of this thesis. Gap 1 motivates the unified IoT–AI–DT architecture developed in Chapter 3. Gap 2 motivates the structured IoT-derived context modeling and context-aware wide-and-deep model developed in Chapters 4 and 5. Gap 3 motivates the DT-based scenario simulation layer developed in Chapter 6. Gap 4 motivates the productivity-oriented evaluation presented in Chapter 7. The thesis therefore positions personalization in smart cities as an integrated, context-aware, and productivity-oriented decision-support problem rather than as an isolated recommendation task.

Chapter 3

System Architecture and Data Flow

3.1 End-to-End System Overview

The proposed system is organized as an end-to-end IoT–AI–DT pipeline that connects physical sensing, structured context construction, predictive and recommender intelligence, DT reasoning, and feedback-enabled interaction. Figure 3.1 provides the general architectural grounding by showing how IoT streams, recommendation components, privacy and security mechanisms, delivery interfaces, and smart-city services are arranged within a unified smart-city recommender system. Figure 3.2 then presents the proposed framework, where physical entities, sensor acquisition, data management, intelligence, DT components, interaction interfaces, and feedback form a continuous operational cycle.

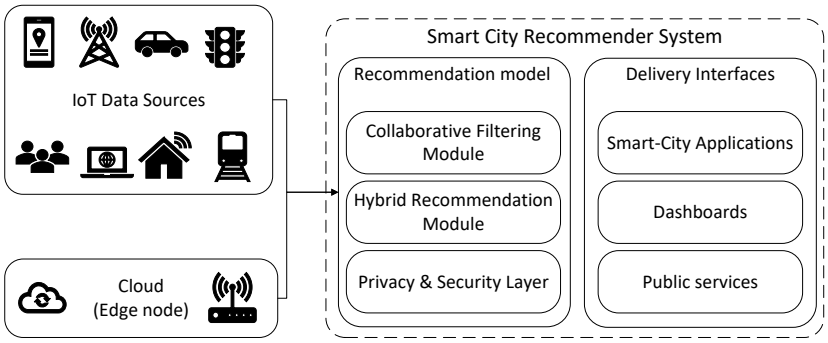

Figure 3.1: General Architecture of Smart Cities Recommender System

The architecture follows a layered structure in which each layer has a distinct operational role while remaining connected to the rest of the pipeline. The physical layer represents users, city or area entities, and PoI. The sensor data acquisition layer collects heterogeneous observations from urban devices and data sources. The data management layer organizes these observations into

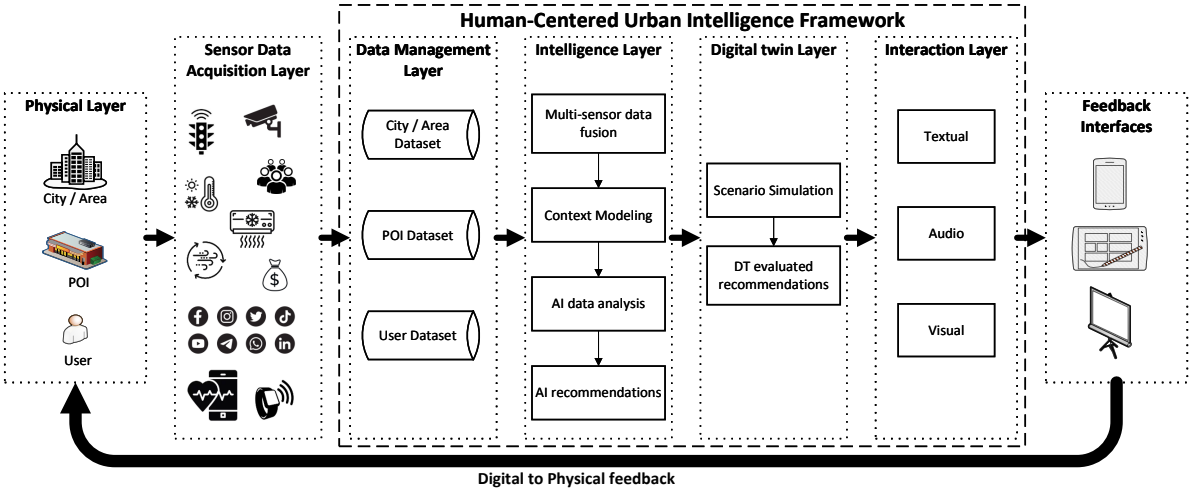

Figure 3.2: High-level architecture of the proposed Human-Centered Urban Intelligence framework

user, PoI, and city or area datasets. The intelligence layer performs context modeling, AI-based analysis and AI-recommndation. The DT layer performs scenario simulation and evaluates the AI-generated activity recommendations. The interaction layer delivers results through textual, audio, and visual interfaces, while feedback interfaces return user and system responses to the physical and digital components. This layered organization supports modularity and coordinated operation across sensing, analytics, and application layers [23, 60].

3.1.1 Formal System Definition

The framework is formalized as a discrete-time context-aware decision-support system built on IoT-based context acquisition, context-aware recommendation, and DT synchronization [2, 30, 78, 92]. At each time step t, the system receives heterogeneous observations from a set of sensing sources $S = \{s_1, s_2, \ldots, s_n\}$. Each source s_i generates an observation $x_{i,t}$, where $x_{i,t}$ may include environmental, spatial, mobility, infrastructure, or user-related measurements. The complete raw input at time t is defined as:

$$X_t = \{x_{1,t}, x_{2,t}, \ldots, x_{n,t}\} \tag{3.1}$$

The preprocessing and context-construction function $\psi(\cdot)$ transforms the raw input stream into a structured context vector:

$$c_t = \psi(X_t, q_t, m_t), \tag{3.2}$$

where q_t represents data-quality indicators such as missingness, noise, uncertainty, and confidence, and m_t represents metadata such as timestamp, location, source identifier, and sampling frequency. The context vector c_t contains environmental, spatial, temporal, activity-related, and user-related features.

The intelligence layer maps the current context vector and user state h_t into a latent embedding:

$$z_t = \phi(h_t, c_t), \tag{3.3}$$

where $\phi(\cdot)$ denotes the embedding and feature-representation function. The predictive model estimates the user or system state:

$$\hat{y}_t = f_\theta(z_t), \tag{3.4}$$

where θ denotes model parameters. The recommender model scores each candidate item, service, activity, or PoI $i \in I$:

$$s_{u,i,t} = g_\theta(z_t, i), \tag{3.5}$$

and returns the top-k recommendations:

$$R_t = \mathrm{TopK}_{i \in \mathcal{I}} \left(s_{u,i,t} \right). \tag{3.6}$$

The DT state is represented as D_t, consisting of the Human DT D_t^H, City or Area DT D_t^C, and PoI DT D_t^P:

$$D_t = \{D_t^H, D_t^C, D_t^P\}. \tag{3.7}$$

The DT update function is defined as

$$D_t = \Gamma \left(D_{t-1}, c_t, \hat{y}_t, R_t, \varrho_{t-1} \right), \tag{3.8}$$

where ϱ_{t-1} represents feedback observed from the preceding recommendation cycle, when such feedback is available. The operator $\Gamma(\cdot)$ denotes the conceptual state-update mechanism through which the previous DT state is refreshed using the current context, AI outputs, recommendation set, and prior feedback.

For each AI-generated recommendation $i \in R_t$, the DT layer then performs scenario evaluation according to

$$\mathcal{E}_{u,i,t} = \Omega\left(D_t^H, D_{i,t}^P, D_t^C, c_t, i\right), \qquad i \in R_t, \tag{3.9}$$

where $\mathcal{E}_{u,i,t}$ denotes the DT scenario-evaluation outcome for candidate i, and $\Omega(\cdot)$ represents the comparative evaluation function applied to the Human, PoI, and City or Area DT states under the current context. The detailed scenario-quality formulation is developed in Chapter 6.

The system therefore operates as a closed loop: IoT streams update context, context supports prediction and recommendation, AI outputs update DT states, DTs evaluate recommended alternatives, and observed responses inform subsequent system updates. Table 3.1 summarizes the main inputs, intermediate states, and outputs used in the formal definition.

Table 3.1: Formal input, intermediate, and output definitions in the proposed architecture

Category	Symbol	Description
Input	X_t	Raw heterogeneous IoT data streams collected at time t, including environmental, mobility, infrastructure, and user-associated observations.
Input	m_t	Metadata associated with each observation, including timestamp, location, source identifier, and sampling frequency.
Input	q_t	Data-quality indicators, including missingness, noise level, uncertainty, and confidence.
Input	ϱ_{t-1}	Observed feedback or response after recommendation delivery.
Intermediate	c_t	Context vector constructed from preprocessed IoT data, representing environmental, spatial, temporal, activity, and user-related conditions.
Intermediate	z_t	Learned embedding or latent representation used by the intelligence layer.
Intermediate	h_t	Dynamic user state, including preferences, behavior history, inferred activity, and recent interactions.
Intermediate	D_t	DT state, including Human, City/Area, and PoI DT representations.
Output	$\hat{y}_t$	Predicted user, context, or system state.
Output	R_t	Ranked list of recommended activities, services, or PoI.
Output	$\mathcal{E}_{u,i,t}$	DT scenario-evaluation outcome for candidate $i \in R_t$.

3.1.2 Layer Roles in the Proposed Architecture

The physical layer and context-processing functions provide the operational bridge between the urban environment and the intelligence components. Heterogeneous IoT devices and data sources, including environmental sensors, infrastructure monitors, mobility traces, and user-associated devices, acquire observations from the physical environment. These sources generate high-volume and

high-velocity streams with different spatial and temporal resolutions [78]. Environmental monitoring systems capture variables such as temperature, air quality, and noise, while mobility platforms produce spatio-temporal traces of individual and collective movement [102]. These observations are not used directly by the AI layer. They first pass through data cleaning, temporal alignment, spatial aggregation, normalization, and context inference so that low-level readings are transformed into higher-level constructs such as user activity, environmental conditions, and situational states [11, 78].

The intelligence layer consumes the structured context vector and applies machine-learning models to generate predicted states, ranked recommendations, and inferred preferences. Prediction and recommendation are complementary: predictive models estimate user, context, or system states, while recommender models score candidate activities, services, or PoI under the current context. Context-aware recommender systems treat variables such as time, location, environmental state, and social conditions as direct inputs to the decision process rather than as external filters [1, 2]. Deep-learning-based recommenders are suitable for large and heterogeneous urban datasets because they can capture nonlinear relations and support personalization across domains such as mobility, energy use, and public services [80, 112]. In this framework, however, the outputs of the intelligence layer are not treated as final decisions. They become inputs to the DT layer, where recommendations and predictions can be evaluated against synchronized representations of users, urban areas, and PoI [80].

The DT layer represents the virtual state of the entities involved in the decision process. It includes Human DTs that model user preferences, activity patterns, and inferred behavioral states; City or Area DTs that represent aggregated infrastructure, environmental conditions, and population-level dynamics; and PoI DTs that represent specific buildings, services, or activity locations with attributes such as availability, capacity, and contextual relevance [30, 68, 92]. These entities consume both contextual data from IoT sensing and analytical outputs from the intelligence layer. Predictions and recommendations update the corresponding twin states and support virtual evaluation of alternatives before real-world action, while evolving twin states provide enriched features and constraints for later prediction and recommendation cycles [30, 68].

3.1.3 Feedback, Update Logic, and System Constraints

The proposed framework is closed-loop rather than one-way. Physical-to-digital feedback updates internal representations through new observations from IoT sensors and user devices. These observations are ingested, contextualized, and propagated to the intelligence and DT layers so that virtual representations remain synchronized with current urban conditions [30, 92]. Digital-to-physical feedback occurs when recommendations, predicted outcomes, or simulation results are delivered

to users or urban services through interaction interfaces. As users respond and conditions evolve, their actions generate new data that re-enter the sensing layer, enabling the system to learn from observed effectiveness and refine its models over time [92].

Scalability and responsiveness constrain this feedback logic because urban data streams are large, heterogeneous, and continuous. The framework therefore assumes streaming-based data pipelines and distributed processing rather than purely batch-oriented operation [38, 115]. Edge resources can support low-latency filtering, aggregation, and preliminary processing near data sources, while cloud resources provide scalable storage, training, and large-scale analytics. The system also assumes that data are incomplete, noisy, and unevenly available across urban areas, so context construction must preserve quality indicators such as missingness, uncertainty, and confidence.

The architecture is designed as a decision-support framework rather than as a fully autonomous control system. Its outputs are advisory and remain subject to user, institutional, or service-provider oversight. DT synchronization is assumed to occur at bounded intervals or after meaningful state changes rather than continuously at raw sensor frequency. Privacy and governance constraints also limit access to personal and behavioral data, requiring aggregation, abstraction, anonymization, or privacy-preserving learning mechanisms where appropriate. These assumptions define the operational boundary of the system while preserving the full IoT–AI–DT pipeline needed for personalized urban intelligence.

Overall, the proposed workflow captures heterogeneous urban data, transforms them into structured context-aware representations, processes them through prediction and recommendation models, incorporates the resulting outputs into Human, City or Area, and PoI DTs, and closes the loop through observed feedback. By structuring sensing, intelligence, DT reasoning, and feedback as one pipeline, this system overview establishes the architectural foundation for the detailed components developed in the subsequent chapters [60, 92].

3.2 Physical Layer and Data Inputs

The physical layer provides the data-entry point of the proposed IoT–AI–DT architecture. Its function is to acquire raw observations from the urban environment, users, PoI, and infrastructure before these observations are transformed into structured context. In this framework, the physical layer includes environmental sensors, mobility and infrastructure data sources, and user-associated devices such as smartphones and wearables [60, 78]. These inputs supply the empirical basis for downstream preprocessing, context modeling, prediction, recommendation, and DT synchronization.

3.2.1 Urban Data Sources

Environmental sensors capture physical conditions that may affect activity suitability, comfort, and productivity. Relevant streams include temperature, humidity, air quality, noise, light intensity, ultraviolet exposure, and energy-related measurements [78, 102]. Wireless and low-cost sensing platforms can improve spatial coverage for environmental monitoring, but they may also introduce variation in measurement quality and reliability [63]. This is consequential for the proposed framework because environmental conditions are later incorporated into context-aware recommendation and productivity-oriented analysis, and prior work links environmental factors to human performance and well-being [111].

Mobility and infrastructure systems form a second major input class. Traffic loop detectors, roadside cameras, GPS traces, public transportation feeds, parking systems, and infrastructure monitors provide observations about movement, congestion, service availability, and urban resource use [32]. These streams are relevant to routing, activity planning, congestion-aware recommendation, and DT updates, but they often originate from different operators, formats, and sensing technologies [32, 80]. Their high update frequency also creates ingestion and transmission demands that require scalable and streaming-oriented support [115].

The physical layer also incorporates user-associated devices when personalization depends on individual context. Smartphones may provide location traces, activity indicators, and interaction logs, while wearables may provide physiological and activity-related signals such as heart rate and physical movement [49]. Such human-centric sensing extends IoT-based context acquisition beyond infrastructure monitoring and supports adaptive user modeling [78]. At the same time, crowdsensing and user-device data introduce trust, participation, and privacy concerns at the point of collection because location, behavioral, and physiological signals may already be sensitive before any higher-level processing occurs [42].

3.2.2 Physical-Layer Constraints

The usefulness of physical-layer data is constrained by heterogeneity, noise, missingness, coverage, latency, bandwidth, energy limits, and privacy. Urban sensing sources differ in sampling rates, spatial resolutions, communication protocols, device reliability, and update patterns. Environmental sensors may report periodically, mobility systems may generate event-driven streams, and user-associated devices may produce irregular observations depending on connectivity, participation, and device state. These differences complicate temporal alignment and spatial integration before context can be inferred [23, 78].

Noise and uncertainty are inherent at this layer. Environmental sensors may be affected by calibration drift, weather exposure, device degradation, and hardware limitations. At the same time, mobility data from GPS, cellular networks, or transport systems may suffer from signal loss, positional error, and incomplete reporting in dense urban environments [32, 102]. User-generated data adds further uncertainty because participation, behavior, and device configurations vary across individuals. Raw physical-layer observations, therefore, cannot be assumed to be complete, accurate, synchronized, or semantically usable.

Coverage is also uneven across urban space. Central districts and infrastructure-rich areas are often more densely instrumented than peripheral or underserved areas, which creates spatial data gaps at the point of acquisition [63]. These gaps can later propagate into biased context representations, predictive models, and recommendation outcomes if they are not explicitly handled. Latency, bandwidth, and energy constraints impose additional limits. High-frequency mobility and environmental monitoring require near-real-time ingestion, buffering, and routing [38], while battery-powered and wireless sensing devices face trade-offs among sampling frequency, communication range, onboard validation, and device lifetime [60].

Although advanced analytics are performed in higher layers, lightweight streaming and edge-oriented mechanisms may be required near the physical layer to preserve responsiveness and reliability. Edge-capable gateways can aggregate local streams, manage connectivity disruptions, and perform basic validation before data moves into the digital processing layer [60]. Streaming infrastructures similarly support the ingestion of heterogeneous sensor streams under low-latency requirements [38]. These mechanisms improve acquisition and transmission, but they do not replace semantic interpretation, context modeling, prediction, or DT reasoning.

Accordingly, the physical layer defines the operational envelope of the entire pipeline. It determines what data enters the system, how complete and reliable that data is, and how quickly it can be made available for downstream processing. Noise, sparsity, uneven coverage, latency, bandwidth constraints, and privacy limitations necessitate preprocessing, quality indicators, temporal alignment, context construction, and privacy-aware handling before AI models or DT components can use the data reliably [23, 42, 49]. The physical layer therefore constrains all subsequent processing and motivates the context-processing layer that transforms heterogeneous urban observations into structured environmental, spatial, temporal, activity-related, and user-related representations.

3.3 Digital Layer: Processing and Intelligence

The digital layer forms the computational bridge between physical-layer sensing and DT-enabled decision support. Its role is to transform heterogeneous IoT streams, environmental observations,

interaction logs, and user-associated signals into structured contextual representations that predictive and recommender models can consume. Intelligence is therefore not derived directly from raw sensing. It is produced through preprocessing, context construction, model-ready representation, AI inference, and DT update inputs [23, 78]. Because the framework targets adaptive smart-city services, these operations must also support continuous data flows and bounded latency through scalable streaming infrastructures [115].

3.3.1 Preprocessing and Context Construction

Physical-layer data are heterogeneous, noisy, sparse, and temporally misaligned. Environmental sensors may be affected by interference, faults, calibration limits, or communication loss, while user-associated data may arrive irregularly and incompletely. The first function of the digital layer is therefore to clean, filter, align, normalize, and organize incoming streams before they are used for learning or DT synchronization [23, 78]. These preprocessing operations convert raw observations into context-ready records while preserving quality indicators such as missingness, uncertainty, and confidence.

Urban intelligence also requires temporal responsiveness. Mobility support, activity recommendation, and context-aware decision support depend on near-real-time ingestion and transformation rather than delayed batch analysis. Streaming infrastructures support this requirement by enabling incremental preprocessing and low-latency routing as data arrive [38]. After cleaning and synchronization, the digital layer fuses temporal, spatial, environmental, behavioral, and device-related signals into unified contextual representations that capture the situational dependencies of urban activity.

Context construction provides the operational link between IoT sensing and model-ready intelligence. Context-aware recommender systems can incorporate context through pre-filtering, post-filtering, or contextual modeling [1, 2]. In the proposed architecture, contextual modeling is the most relevant strategy because environmental, spatial, temporal, activity-related, and user-related features are embedded directly into the downstream learning process rather than treated as external filters. Symbolic and engineered features remain useful for interpretability and control [11], while learned representations support the capture of latent patterns across heterogeneous inputs [103]. The resulting context vector becomes the structured input to the intelligence layer and the update basis for DT entities.

3.3.2 Model-Ready Intelligence and AI Inference

Once raw streams have been transformed into structured context, the digital layer applies AI models to estimate user, service, and system states and to score candidate activities, services, or PoI. At this architectural level, the purpose is not to detail the model families developed later in Chapter 5, but to specify their role in the end-to-end pipeline. The AI component consumes the current context vector, dynamic user state, and candidate service attributes to generate predictions and recommendation scores that support adaptive decision-making under changing urban conditions.

The intelligence layer must operate under sparsity, heterogeneity, nonstationarity, and scale. Collaborative and hybrid recommender approaches provide a basis for personalization, but their effectiveness is limited when interaction data are sparse or when urban context changes rapidly. Hybrid and deep learning approaches address part of this limitation by combining interaction data with content-based and contextual features, enabling richer prediction in cold-start and low-data settings [112]. Deep recommender models are also relevant because they can integrate heterogeneous modalities and learn nonlinear relations among users, activities, spatial conditions, and environmental variables [51, 112]. Large-scale recommender deployments further show that neural models can support strict latency requirements when engineered appropriately [28].

In this framework, AI inference produces model-ready intelligence rather than final autonomous decisions. Its outputs include predicted user or system states, ranked activity or PoI recommendations, inferred contextual relevance, and candidate actions for scenario evaluation. Sequential, predictive, or intent-aware processing may be used where interaction order, short-term goals, or changing activity conditions are relevant [81, 95, 99]. However, these methods are treated here as possible intelligence functions within the pipeline, while the detailed model formulation remains in Chapter 5. The key architectural point is that AI converts structured context into prediction and recommendation outputs that can be passed to the DT layer for synchronization and decision support.

3.3.3 Digital Twin Update Inputs and Deployment Constraints

The outputs of the digital layer provide updated inputs for the DT layer. Predictions, recommendations, contextual states, and observed feedback are used to refresh Human, PoI, and City or Area DTs. In urban systems, DTs must integrate infrastructure, environmental conditions, and human behavior into synchronized representations that support monitoring, simulation, and decision support [30, 92]. Their value therefore depends not only on visualization, but also on the intelligence supplied by the digital layer, including forecasting, scenario evaluation, anomaly detection, and optimization [56, 68]. Within the proposed architecture, the digital layer supplies the structured

and predictive inputs through which DTs participate in the closed-loop interaction between physical and digital domains.

Trust, privacy, and deployment feasibility constrain this process. The digital layer operates on sensitive spatial, behavioral, and contextual data, so privacy-preserving and trustworthy processing must be considered as deployment requirements rather than optional additions. Federated learning can reduce the need to centralize raw data by supporting distributed model training [17], and recommender-specific work identifies its relevance for privacy-aware personalization [100]. Fairness, transparency, accountability, and robustness are also important because urban recommendations may affect access to services and opportunities [34, 37, 59].

The digital layer must also respect latency, scalability, and resource constraints. Cloud-only execution is insufficient for all smart-city services because time-sensitive responses may require computation closer to the data source. Edge computing can support local preprocessing and inference while reducing bandwidth use and delay [62], whereas cloud platforms remain necessary for large-scale training, aggregation, and long-term analytics. Effective edge-cloud orchestration is therefore required to avoid bottlenecks and maintain service quality at city scale [54, 115].

Overall, the digital layer converts raw urban streams into structured context, transforms that context into predictive and recommender intelligence, and provides the updated inputs required by DT representations. It is the operational bridge from sensing to structured context, from context to model-ready inference, and from inference to synchronized DT reasoning within the proposed IoT–AI–DT pipeline.

3.4 Feedback and Update Cycles

Feedback and update cycles define how the proposed IoT–AI–DT architecture remains adaptive after deployment. The system does not operate as a one-way pipeline from sensing to recommendation. Instead, observations from the physical environment update contextual representations and DT states, recommendations influence user or service behavior, and the resulting behavior generates feedback for subsequent updates. This closed-loop logic connects physical sensing, digital reasoning, recommendation delivery, and model adaptation within one operational cycle.

Algorithm 1 summarizes this data-flow cycle. It formalizes the sequence through which IoT observations are validated, transformed into context, processed by AI models, propagated into DT states, delivered as recommendations, and updated using observed feedback. The algorithm draws on established principles of IoT context processing, context-aware recommendation, and DT feedback loops [2, 30, 78, 92].

Algorithm 1: Data Flow in the IoT–AI–DT Recommendation Framework

Input: IoT observations X_t, query data q_t, metadata m_t
Output: Recommendation set R_t, DT scenario evaluations $\mathcal{E}_{u,i,t}$, and updated DT state D_t
for *each time step* t **do**
  Collect raw IoT observations $X_t = \{x_{1,t}, x_{2,t}, \ldots, x_{n,t}\}$
  Validate, clean, and align observations
  Construct context vector $c_t = \psi(X_t, q_t, m_t)$
  Update user state h_t using context and interaction history
  Generate embedding $z_t = \phi(h_t, c_t)$
  Predict system state $\hat{y}_t = f_\theta(z_t)$
  for *each candidate* $i \in I$ **do**
    Compute recommendation score $s_{u,i,t} = g_\theta(z_t, i)$
  end
  Generate top-k recommendations $R_t = \mathrm{TopK}_{i \in \mathcal{I}}(s_{u,i,t})$
  Update DT state $D_t = \Gamma(D_{t-1}, c_t, \hat{y}_t, R_t, \varrho_{t-1})$
  for *each candidate* $i \in R_t$ **do**
    Evaluate DT scenario $\mathcal{E}_{u,i,t} = \Omega(D_t^H, D_{i,t}^P, D_t^C, c_t, i)$
  end
  Deliver recommendations to the user or service interface
  Observe feedback ϱ_t for the next update cycle
end

3.4.1 Closed-Loop Feedback Flow

The feedback loop begins with physical-to-digital updates. Sensors, mobile devices, and urban platforms collect observations about environmental conditions, mobility, infrastructure use, and user activity. These streams are transmitted through communication and middleware layers, where they are prepared for preprocessing, alignment, and contextual integration [60, 78]. In the proposed architecture, these updates refresh raw repositories, structured context vectors, user states, and DT representations rather than remaining as isolated sensor readings.

Streaming-oriented processing supports this stage because urban data often arrives continuously and at different velocities. Near-real-time ingestion allows the system to update contextual and DT states using recent observations, while also preserving latency bounds needed for adaptive recommendation and decision support [38, 115]. This enables the system to respond to changing urban conditions without requiring complete reprocessing of the full data history at every update.

Digital-to-physical feedback occurs when predictions, recommendation lists, alerts, or service signals are returned to users or operational interfaces. In this framework, recommender outputs are the primary mechanism through which digital intelligence influences physical behavior. Recommendations may shape mobility choices, activity selection, energy use, or access to services by

presenting context-sensitive options under current conditions. Prior work in smart environments shows that timely feedback can affect behavioral patterns and resource consumption [52], while urban decision-support systems may also provide administrators with analytics for planning and operational adjustment [27].

The resulting behavior then becomes part of the next physical-to-digital update. A user may accept, reject, ignore, or modify a recommendation, and the surrounding environment may change after the action is taken. These responses produce feedback signals such as explicit ratings, implicit interaction traces, compliance patterns, activity outcomes, and differences between predicted and observed states. The system therefore treats action and observation as linked stages in the same cycle rather than as separate processes.

3.4.2 Learning, Update Timescales, and Constraints

Feedback also supports model adaptation. Observed responses provide evidence for recalibrating user states, contextual weights, latent representations, and ranking strategies. This is important in context-aware recommender systems because preferences, activity goals, environmental exposure, and urban conditions vary across time and location [2, 81]. In the proposed architecture, feedback is not only stored as historical interaction data. It is used to update the next context construction and recommendation cycle, allowing the system to refine its predictions as the relationship between users, places, and conditions changes.

Update cycles operate across multiple timescales. Short-term updates support responsive services, such as mobility guidance, congestion alerts, or activity recommendations under changing conditions [30]. Medium-term updates support periodic forecasting, allocation, service optimization, and recalibration of contextual patterns. Longer-term updates support retraining, feature redesign, policy adjustment, and DT refinement as urban infrastructure, institutional priorities, and user behavior evolve [30, 78, 92]. This multi-timescale structure prevents the system from treating all feedback as equivalent and allows fast operational adjustments to coexist with slower model and governance updates.

These cycles remain constrained by human agency and institutional context. Users may reject or ignore recommendations, so feedback reflects trust, acceptance, perceived usefulness, and contextual appropriateness as well as technical performance [44]. Human-in-the-loop operation therefore requires interpretability, accountability, and support for human judgment, especially when recommendations affect public services or urban resource access [55]. Institutional responsibilities, legal frameworks, infrastructure limits, and governance arrangements may also restrict how far automated adaptation can proceed in practice [71].

Overall, feedback and update cycles operationalize the proposed IoT–AI–DT pipeline. Observations update context and DT states, recommendations influence user or system behavior, behavior generates feedback, and feedback informs the next sensing, learning, and synchronization cycle. The effectiveness of this loop depends on timely physical-to-digital updates, meaningful digital-to-physical feedback, adaptive learning, and clear constraints on how technical outputs are interpreted and acted upon.

3.5 Design Assumptions and Constraints

This section defines the assumptions that bound the proposed IoT–AI–DT architecture and clarifies how its outputs should be interpreted in practical smart-city deployments. The system is assumed to operate as an advisory decision-support framework, not as a fully autonomous urban control platform. Its purpose is to transform heterogeneous IoT observations into structured context, use AI models to generate predictions and recommendations, update DT representations, and return feedback to users or service interfaces while preserving human and institutional oversight.

3.5.1 System Scope and Deployment Assumptions

The architecture assumes partial and incremental deployment across urban environments. Sensing coverage, data integration, service availability, and institutional adoption are not assumed to be uniform across all locations or user groups. Some areas may provide dense and reliable observations, while others may contain missing, delayed, or low-quality data. The system therefore does not claim complete observability of the city or uninterrupted synchronization between all physical and digital entities.

Within this boundary, recommendations and predictions are treated as time-sensitive but not safety-critical outputs. The DT layer is assumed to support scenario-based reasoning and decision interpretation, but its outputs remain advisory. Users, service providers, and policymakers may accept, reject, modify, or override the recommendations produced by the system. This assumption is consistent with the role of the proposed framework as a human-centered decision-support pipeline rather than an automated control mechanism.

3.5.2 Scalability, Data, and Latency Constraints

The architecture assumes that urban data are heterogeneous, incomplete, noisy, and only partially observable. IoT streams may differ in sampling frequency, spatial resolution, reliability, and semantic structure, while user-associated data may be irregular or unavailable because of device limitations,

participation, privacy settings, or connectivity loss. The system therefore tolerates missingness and uncertainty through preprocessing, quality indicators, context construction, and bounded update cycles rather than requiring complete data availability.

Scalability and latency are treated as operational constraints across the full IoT–AI–DT pipeline. The system must ingest observations, preprocess and align them, construct context vectors, perform AI inference, rank candidate activities or services, update DT states, and deliver outputs through interaction interfaces. The architecture does not guarantee strict real-time performance for every component. Instead, it assumes distributed ingestion, selective edge preprocessing, parallelizable inference and ranking, and cloud-based analytics to balance responsiveness with coordination requirements [23, 46, 88, 114].

Two growth dimensions are particularly important: the number of active users and the volume and velocity of sensed data. User growth increases concurrent recommendation requests, maintained user states, inference load, and Human DT updates. Data growth increases ingestion, buffering, synchronization, preprocessing, and storage demand. These pressures are addressed through horizontally distributed ingestion, parallel or batched inference, candidate filtering, caching, and selective DT synchronization.

Formally, latency is defined as the elapsed time between the arrival of relevant IoT observations and the availability of the corresponding recommendation or DT update. For a time step t,

$$L_t = T_t^{out} - T_t^{in}, \tag{3.10}$$

where T_t^{in} denotes the time at which the system receives the relevant observations and T_t^{out} denotes the time at which the recommendation or updated DT state becomes available.

Scalability is defined as the ability to maintain bounded latency and acceptable throughput as the number of concurrent users U, incoming IoT streams n, candidate services m, and active DT entities $|D|$ increases. For an online recommendation cycle with context-vector dimensionality d, embedding dimensionality d_emb, m candidate services, k returned recommendations, and $|D|$ updated DT entities, the approximate per-cycle computational cost is

$$O(n + d + dd_emb + md_emb + m \log k + |D|). \tag{3.11}$$

This expression treats context construction, prediction, candidate scoring, top-k selection, and DT updating as the dominant operations. When m is large, candidate scoring and ranking become the main bottleneck, motivating candidate filtering, caching, or approximate retrieval before full ranking.

Throughput is defined as the number of complete recommendation cycles processed per unit time,

$$\lambda = \frac{N_{cycles}}{\Delta T}. \tag{3.12}$$

Stable operation assumes that the effective processing rate ν satisfies $\nu \geq \lambda$; otherwise, queues accumulate and end-to-end latency increases.

DT synchronization is also bounded by scale. The system does not assume that every Human, PoI, or City/Area DT is updated continuously at raw sensor frequency. Instead, synchronization occurs at defined intervals, after meaningful state changes, or when a recommendation request requires a sufficiently fresh state. This selective update strategy reduces unnecessary communication and computation while preserving useful state for decision support. DT outputs therefore represent the best available structured state under deployment constraints rather than a perfectly current replica of the physical city.

3.5.3 Privacy, Governance, and Human Oversight

The architecture assumes that privacy regulation, consent requirements, ethical considerations, and institutional governance constrain access to personal, behavioral, spatial, and contextual data. The system therefore favors aggregation, abstraction, anonymization, and privacy-preserving learning mechanisms where appropriate, even when these reduce the granularity of personalization. This constraint is especially important for Human DTs, which may depend on sensitive mobility, activity, preference, or device-derived signals.

The system also assumes that AI models and DT representations are context-dependent. Models trained in one city, population, time period, or sensing configuration may not transfer directly to another without recalibration. Distribution shift, changing user behavior, uneven infrastructure, and evolving service conditions require iterative feedback, monitoring, and periodic retraining rather than one-time deployment.

Finally, the proposed architecture assumes continuing human and institutional oversight. Predictive models, recommender outputs, and DT simulations provide structured evidence for decision support, but they do not determine binding outcomes. Their interpretation remains subject to user judgment, service-provider constraints, policy requirements, and governance review. These assumptions define the operational boundary of the framework and prevent its results from being interpreted as autonomous decisions or universally valid predictions.

3.6 Chapter Summary

This chapter defined the end-to-end IoT—AI—DT architecture, formalized the flow from heterogeneous observations to structured context and AI-generated recommendations, and established the feedback, scalability, privacy, and deployment assumptions that bound the framework. The architecture separates physical sensing, context processing, AI recommendation, and DT-based reasoning while connecting them through explicit update and feedback pathways. Chapter 4 next specifies how the heterogeneous observations introduced here are transformed into structured model-ready context.

Chapter 4

IoT and Context Modeling

Building on the system architecture defined in Chapter 3, this chapter defines how raw urban data are transformed into a structured context for the proposed IoT–AI–DT pipeline. Chapter 3 specified the overall system, its data flow, and its feedback logic. The purpose of this chapter is narrower: it specifies the IoT acquisition, context modeling, preprocessing, and data-management layer that supplies downstream prediction, recommendation, and DT synchronization with usable contextual inputs.

The chapter focuses on three connected functions. First, it defines how heterogeneous environmental, mobility, infrastructure, and user-related signals are acquired through the IoT layer. Second, it explains how these observations are organized into structured representations of environmental, user, spatial, and temporal context. Third, it specifies the preprocessing operations required to manage heterogeneity, synchronization, noise, and incomplete data before these inputs are consumed by later AI and DT modules.

This layer is necessary because downstream intelligence depends not only on data availability, but also on whether context is timely, aligned, and semantically usable. Poor sensing quality, delayed updates, or weakly structured context can degrade model performance and reduce the operational value of synchronized DT representations [23, 30, 68]. The framework developed in this chapter therefore serves as the empirical interface between physical urban conditions and the higher-level intelligence layers of the thesis.

The chapter also considers the practical constraints that shape this interface, including incomplete sensing coverage, noisy or unreliable observations, scalability and synchronization pressures, and communication limits that affect update frequency and contextual granularity [38, 96]. These issues are addressed only insofar as they affect the design and operation of the IoT and context pipeline developed in this chapter.

4.1 IoT Data Acquisition Layer

The IoT data acquisition layer provides the operational interface between the physical urban environment and the downstream context modeling, prediction, recommendation, and DT synchronization modules. In this chapter, its role is not to restate a generic IoT architecture, but to define how heterogeneous urban signals are sensed, ingested, described, and constrained before they become usable contextual inputs. The layer therefore focuses on three connected requirements: capturing relevant environmental, infrastructural, and user-associated signals; preserving the metadata needed for contextual interpretation; and exposing data through interoperable interfaces that do not bind higher-level modules to specific devices, vendors, or communication technologies [11, 60].

4.1.1 Sensing Sources and Captured Signals

The sensing function of the acquisition layer collects the raw observations from which later context representations are constructed. In the proposed framework, these observations come from three main source groups: environmental sensors, infrastructure-oriented sources, and user-centric devices. Environmental sensors provide physical variables such as temperature, humidity, air quality, noise, and ultraviolet exposure. These measurements are retained because they directly support the environmental context dimension used later for recommendation and DT synchronization [60, 102].

Infrastructure-oriented sources capture urban system states associated with traffic systems, parking facilities, energy meters, public transportation services, buildings, and other operational assets. These signals provide information about congestion, service availability, resource use, and localized operating conditions. They are necessary because context-aware recommendations in an urban environment depend not only on user preferences but also on the current state of services, infrastructure, and PoI.

User-centric devices, including smartphones, wearables, and personal IoT devices, contribute mobility, activity, interaction, and device-derived signals. These sources support personalization because they connect sensed urban conditions to individual context. However, the framework does not assume that all sensing sources behave uniformly. Environmental sensors, infrastructure platforms, and user devices differ in sampling rate, precision, reliability, connectivity, spatial coverage, and energy constraints. These differences are treated as baseline acquisition conditions that must be preserved and managed before downstream preprocessing and context construction.

For this reason, raw measurements are not forwarded as isolated values. Each observation is expected to remain associated, where available, with temporal, spatial, source, sampling, and quality information. At the sensing stage, this includes at minimum the timestamp of acquisition,

location or spatial reference, source identifier, sampling rate or update frequency, and basic quality indicators. Preserving these descriptors allows later modules to distinguish current from stale data, high-confidence readings from uncertain readings, and directly observed values from missing or reconstructed segments. The acquisition layer therefore prepares raw signals for later contextualization without performing the full semantic interpretation assigned to the context-modeling layer.

4.1.2 Ingestion, Metadata, and Interoperability

The ingestion function forms the transition between distributed sensing and structured context preparation. Its purpose is to receive heterogeneous and asynchronous urban streams, associate them with the metadata required for later interpretation, and regulate their flow into downstream processing modules. In this framework, ingestion performs four core operations: metadata association, format and unit normalization, buffering of variable-rate streams, and initial quality flagging.

Metadata association links each incoming observation to its timestamp, location, source identifier, sampling rate, and acquisition channel. This step is essential because later context inference depends on alignment across streams rather than on isolated measurements. For example, an environmental reading, a mobility update, and a user-device signal can only be interpreted together if their temporal and spatial relationships are preserved. Sampling metadata also matters because urban streams may arrive periodically, event-driven, or irregularly, depending on device type and connectivity conditions.

Normalization addresses heterogeneity in formats, units, naming conventions, schemas, and data scales. Environmental sensors, transport systems, user devices, and service platforms may expose data using different structures and update schedules. Without normalization, higher-level modules would inherit device-specific and vendor-specific dependencies. The acquisition layer therefore converts incoming observations into common representations while preserving the source references needed for traceability and quality assessment.

Buffering and stream regulation are required because urban data do not arrive at uniform rates. Congestion, public events, environmental incidents, or intermittent connectivity may produce bursts, delays, or missing segments. The ingestion layer buffers and regulates these streams so that downstream preprocessing and context construction are not destabilized by temporary input surges. This does not make scalability a separate broad topic in this section; rather, scalability is treated as a practical acquisition requirement supported by buffering, asynchronous producer-consumer decoupling, and modular stream handling.

Interoperability is handled as a practical requirement of ingestion and data exposure. The framework uses standardized representations and access mechanisms so that context modeling, AI

inference, and DT synchronization can consume acquired data without direct dependence on low-level device formats, communication protocols, or vendor-specific interfaces [11, 38, 78]. This is particularly important because smart urban deployments commonly combine sensing and service components from different vendors, institutions, and administrative domains. New sensing components are therefore expected to integrate through shared data representations and coordinated access interfaces rather than through direct modification of the higher-level context and intelligence modules.

The result of ingestion is not a final context model, but a context-ready acquisition record. Each record contains the sensed value, its normalized representation, its time and location descriptors, its source and sampling information, and its initial quality status. This structure supports the thesis-specific context dimensions developed in Section 4.2, while preserving the separation between acquisition, preprocessing, semantic context modeling, and predictive reasoning.

4.1.3 Edge-Cloud Distribution and Governance Constraints

Edge and cloud resources are used in this layer as a task-distribution mechanism rather than as a separate conceptual architecture. The assignment of work is driven by latency, bandwidth, reliability, and computational requirements. Edge resources handle lightweight and latency-sensitive acquisition tasks such as local filtering, preliminary aggregation, compression, buffering, and basic validation, especially where connectivity is unstable or rapid local updates are required. Cloud resources support larger-scale storage, historical aggregation, broader fusion, model training, and computationally heavier prediction or recommendation tasks [23].

This distribution preserves the same acquisition pathway while allowing implementation choices to vary across deployment settings. A sensing stream may be filtered at an edge gateway, buffered during connectivity disruption, and later synchronized with cloud storage for broader analysis. Conversely, data requiring historical comparison or city-wide aggregation may be routed to cloud resources after initial validation. The important design point is that edge and cloud placement should not change the meaning or structure of the acquired data. Standardized interfaces and metadata preservation allow workloads to shift across resources without requiring the downstream context and DT modules to be redesigned.

Communication is treated within this same acquisition logic. The framework does not commit to a single protocol family. Instead, it assumes that implementations may combine short-range wireless links, IP-based connectivity, gateways, and cellular backhaul depending on deployment conditions. This abstraction prevents higher layers from depending on protocol-specific decisions about coverage, bandwidth, latency, or energy use. It also reflects the practical reality that mobile

and battery-powered devices may experience packet loss, interference, intermittent connectivity, or constrained update frequency.

Privacy and governance are retained here only as constraints on acquisition and exposure. At the IoT layer, governance determines what data are collected, how identifiable they remain, who can access them, and what later privacy-preserving operations remain feasible [42, 100]. These constraints are especially important because measurements become more sensitive when linked with time, location, source identity, and user-associated signals. Environmental readings may reveal behavioral patterns when tied to place and time, while smartphones and wearables can introduce direct mobility, activity, and physiological information [42].

Accordingly, the acquisition layer assumes controlled exposure, traceable data flows, and support for data minimization rather than unrestricted central collection. Where required, aggregation, anonymization, access control, or later privacy-preserving learning mechanisms can be applied before data are broadly stored or shared [42, 100]. Multi-stakeholder urban deployments also require the acquisition layer to support data from municipal, private, and citizen-controlled sources. Interoperable interfaces therefore serve not only modularity, but also auditable handling, distributed ownership, and controlled data exchange.

Overall, the IoT data acquisition layer defines what is sensed, how it is ingested and described, and which acquisition constraints affect its use. Environmental sensors, infrastructure-oriented sources, and user-centric devices provide the raw evidence for context construction. Ingestion preserves the metadata, quality indicators, and interoperable representations needed for downstream alignment. Edge-cloud distribution, communication variability, privacy, and governance then define the practical limits under which acquisition can operate. This layer therefore supplies the structured acquisition foundation required by the context modeling, preprocessing, AI, and DT components developed in the rest of the chapter and thesis.

4.2 Context Modeling

4.2.1 Role and Adopted Context Dimensions

In the proposed framework, context modeling is the mechanism that converts heterogeneous IoT observations into structured inputs for recommendation and DT synchronization. Context is not treated as raw sensor data. It is modeled as a structured representation built from environmental, spatial, user, activity-related, behavioral, and temporal dimensions derived from heterogeneous urban data streams [11, 78]. This distinction is necessary because raw observations from sensors, mobile devices, wearables, infrastructure systems, and digital services are not directly suitable for downstream prediction or recommendation. They must first be organized into semantically

meaningful, temporally aligned, and model-ready representations linked to entities, locations, and time intervals [11, 22, 78].

The role of this layer is therefore narrower than the general context-aware computing background discussed in Chapter 2. It specifies how context is represented in this thesis and how it connects IoT acquisition to AI-driven recommendation and DT updating. The practical flow is acquisition, structuring, encoding, and delivery to downstream modules, with emphasis on the representation choices required by the proposed framework rather than on reintroducing the full context-aware computing life cycle.

The framework models context through five connected dimensions. Environmental context represents sensed physical conditions that may influence user comfort, activity suitability, and urban service conditions. In this thesis, the core environmental variables are temperature, humidity, noise level, air quality, and ultraviolet exposure. These variables are acquired through IoT sensing and associated with time, location, source, and quality metadata. Table 4.1 summarizes the baseline environmental variables used in the framework.

Table 4.1: Environmental variables acquired through IoT sensors to support context-aware recommendation and DT synchronization

Environmental Variable	*Description*
Temperature	Ambient temperature measured within the monitored environment.
Humidity	Relative humidity level in the surrounding indoor or outdoor environment.
Noise Level	Ambient sound intensity measured in decibels (dB), reflecting environmental noise conditions.
Air Quality	Concentration of environmental pollutants such as carbon monoxide (CO), carbon dioxide (CO_2), and particulate matter.
Ultraviolet Index	Measurement of ultraviolet radiation exposure in the environment.

Spatial context represents location-dependent information such as geographic position, region identifier, distance to candidate PoI, accessibility, and semantic place relationships. User context captures the attributes required for personalization, including relatively stable attributes such as long-term preferences and profile characteristics, and dynamic attributes such as recent interactions, current location, device-derived state, and optional physiological indicators. Activity and behavioral context represent inferred user actions, movement states, engagement patterns, routines, and deviations derived from temporal sensor traces, location transitions, and interaction sequences. Temporal context represents time-dependent conditions such as time of day, day of week, seasonal indicators, event ordering, and sequence position. Together, these dimensions define the context

feature space used by the recommender models and by the Human, PoI, and City or Area DTs.

4.2.2 Formal Context Model

The context state at time t is formalized as a structured vector derived from five context dimensions. Let N_t denote the environmental state, Λ_t the spatial state, H_t the user state, G_t the activity and behavioral state, and V_t the temporal state. The context model is defined as

$$c_t = f_{\text{ctx}}(N_t, \Lambda_t, H_t, G_t, V_t), \tag{4.1}$$

where c_t is the structured context vector supplied to downstream prediction, recommendation, and DT synchronization modules.

This notation refines the context variable introduced in Chapter 3. The context vector was defined at the system level as the output of preprocessing and context construction, where raw IoT inputs, metadata, and quality indicators are transformed into c_t. In this chapter, the same context object is decomposed into its internal representational dimensions.

The environmental component N_t contains sensed physical conditions such as temperature, humidity, noise level, air quality, and ultraviolet exposure. The spatial component Λ_t represents location-dependent information such as geographic position, region identifier, distance to candidate PoI, and semantic place relationships. The user component H_t captures profile characteristics, preferences, recent interactions, current device-derived state, and optional physiological indicators. The activity and behavioral component G_t represents current or inferred activity conditions, including movement state, engagement pattern, routine behavior, or activity category. The temporal component V_t represents time-of-day, day-of-week, seasonal, and sequence-based indicators.

The complete context vector is constructed by concatenating encoded representations of these dimensions:

$$c_t = [n_t \| l_t \| u_t \| a_t \| \tau_t], \tag{4.2}$$

where n_t, l_t, u_t, a_t, and τ_t are the encoded environmental, spatial, user, activity and behavioral, and temporal vectors, respectively, and $\|$ denotes vector concatenation. This formulation makes explicit that context in the proposed framework is not a single observation, but a multidimensional representation derived from heterogeneous IoT and user-related data streams.

4.2.3 Context Representation, Encoding, and Quality

The framework adopts a hybrid representation strategy. Operational context is stored as structured multidimensional records associated with entities, locations, timestamps, source descriptors, and

quality indicators. This supports efficient updating while preserving the relationships among users, places, services, and environmental states required by downstream analytics and DT synchronization [11, 78].

At the feature level, each context dimension is encoded before inclusion in the complete context vector. Continuous environmental variables are represented as normalized numerical values. The environmental vector is written as

$$n_t = [\widetilde{\text{temp}}_t, \widetilde{\text{hum}}_t, \widetilde{\text{noise}}_t, \widetilde{\text{aqi}}_t, \widetilde{\text{uv}}_t], \tag{4.3}$$

where each term denotes a normalized environmental measurement at time t. For a continuous variable $x_{j,t}$, z-score normalization is defined as

$$\widetilde{x}_{j,t} = \frac{x_{j,t} - \mu_j}{\sigma_j}, \tag{4.4}$$

where μ_j and σ_j are the mean and standard deviation of feature j estimated from the training data. Alternatively, min–max normalization may be used when bounded sensor ranges are known:

$$\widetilde{x}_{j,t} = \frac{x_{j,t} - x_j^{min}}{x_j^{max} - x_j^{min}}. \tag{4.5}$$

Categorical and high-cardinality variables are encoded using learned embeddings. For a categorical variable $x_{k,t}$, such as activity type, region identifier, PoI category, or temporal bucket, the embedding representation is defined as

$$v_{k,t} = Emb_k(x_{k,t}), \tag{4.6}$$

where $Emb_k(\cdot)$ is a learnable lookup function that maps category values into dense vectors. This allows sparse symbolic variables to be represented in a continuous latent space suitable for downstream AI models.

Context quality is treated as part of the representation rather than as an external preprocessing concern. Accuracy, timeliness, completeness, and reliability affect whether derived context can support dependable recommendations and DT updates [11]. Because urban IoT data are noisy, incomplete, and temporally inconsistent, context records preserve quality indicators such as missingness, uncertainty, source reliability, and recency where available [11, 78]. These indicators allow later modules to distinguish directly observed values from reconstructed or uncertain values.

Inference over context is therefore limited to the operations required by the implemented framework. Deterministic transformations are used for normalization, encoding, temporal assignment,

and feature construction. Uncertainty-aware handling is supported through preserved quality indicators and missingness metadata. Higher-level predictive inference is assigned to the AI models developed in Chapter 5 rather than treated as a separate general reasoning layer in this section.

Within the broader architecture, context modeling functions as the thesis-specific interface between IoT acquisition and higher-level intelligence. It provides the structured environmental, spatial, user, activity, behavioral, and temporal features required by the recommender models and DT entities, while reducing direct dependence on raw sensor formats and device-level heterogeneity. This separation supports interoperability, modular updating, and scalable integration with the predictive and simulation components developed in later chapters.

4.3 Data Preprocessing and Management

Data preprocessing is the operational stage that converts acquired IoT observations into model-ready context for the framework defined in this thesis. Its role is to clean heterogeneous streams, standardize their representations, align them across time, handle missing observations, encode contextual variables, fuse multiple sources, and produce structured records that can be consumed by the AI and DT layers. This section therefore focuses on the adopted pipeline rather than on a general explanation of preprocessing in context-aware systems.

The need for this stage follows from the characteristics of the sensed data used in the framework. Environmental sensors, infrastructure systems, mobile devices, wearables, vehicular sources, and platform data differ in source type, update rate, spatial granularity, reliability, and semantic meaning. These observations cannot be passed directly to context models or learning modules. They must first be transformed into temporally coherent, quality-controlled, and semantically usable representations that support downstream reasoning [11, 78]. In the proposed pipeline, the intended flow is therefore: standardize, validate, fill missing values, encode, align, fuse, enrich, and hand off to prediction, recommendation, and DT synchronization.

4.3.1 Standardization, Quality Control, and Missing-Data Handling

The first preprocessing stage standardizes heterogeneous inputs. Incoming measurements are associated with source metadata, normalized across units and schemas, and reorganized into a common representation that supports later fusion across environmental, spatial, temporal, user-related, activity-related, and service-related dimensions. This harmonization is required because urban sensing streams differ not only in format, but also in semantic scope, sampling frequency, and spatial resolution.

Quality control is then applied before context construction. Raw streams are filtered for obvious errors, checked against expected ranges, smoothed where appropriate, and flagged when anomalies indicate device malfunction, environmental interference, communication failure, or stale observations. In the proposed framework, these operations determine whether a segment can be used directly, reconstructed, aggregated at a coarser resolution, or excluded from downstream inference. Quality control is therefore part of the operational reliability of the pipeline rather than an optional cleaning step.

Missing observations are handled explicitly because outages, maintenance, communication loss, irregular user participation, and device constraints are expected in urban IoT streams. For each feature j at time t, a missingness indicator is defined as

$$\delta_{j,t} = \begin{cases} 1, & \text{if feature } j \text{ is observed at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

The framework applies a hierarchical imputation strategy. The completed value $\hat{x}_{j,t}$ is defined as

$$\hat{x}_{j,t} = \begin{cases} x_{j,t}, & \delta_{j,t} = 1, \\ x_{j,t-1}, & \delta_{j,t} = 0 \text{ and a recent previous value exists,} \\ \mu_j, & \delta_{j,t} = 0 \text{ and no recent previous value exists,} \end{cases} \tag{4.8}$$

where μ_j is the historical mean or median of feature j estimated from the available training data. Short gaps are therefore handled through forward filling, while longer or unresolved gaps are handled through statistical imputation.

The missingness indicators are retained after imputation as part of the quality metadata. This allows downstream models to distinguish directly observed values from reconstructed values. When the missingness rate for a feature or stream exceeds an acceptable threshold, the affected segment may be aggregated at a coarser temporal resolution or excluded from context construction to avoid unreliable inference.

4.3.2 Context Encoding and Model-Ready Representation

After standardization, quality control, and missing-data handling, the pipeline encodes each context dimension into a representation suitable for the AI and DT layers. This subsection does not redefine the formal context model from Section 4.2. Instead, it specifies how the cleaned observations are converted into the input forms required by the later learning architecture.

Continuous environmental variables such as temperature, humidity, noise level, air quality, and ultraviolet exposure are encoded as normalized numerical features. Spatial variables are represented using coordinates, distance features, region or grid identifiers, and location embeddings. User-related variables are represented through profile attributes, preference vectors, interaction summaries, and optional user embeddings. Activity-related variables are represented through activity labels, movement-state indicators, routine patterns, or learned activity embeddings. Temporal variables are represented through time buckets or cyclic encodings when appropriate, so that downstream models can use recurrent daily or weekly structure.

For cyclic time variables such as hour of day, sinusoidal encoding may be used:

$$\tau_t^{sin} = \sin\left(\frac{2\pi h_t^{\text{day}}}{24}\right), \qquad \tau_t^{cos} = \cos\left(\frac{2\pi h_t^{\text{day}}}{24}\right), \tag{4.9}$$

where h_t^{day} is the hour of the day, this representation avoids treating adjacent hours around midnight as distant categories.

The encoded context record therefore contains normalized continuous values, embedded categorical values, temporal encodings, service or PoI attributes, and quality metadata. Table 4.2 summarizes the encoding strategy retained for the proposed framework and aligns the preprocessing output with the feature groups used by the AI model in Chapter 5.

Table 4.2: Encoding strategy for model-ready context in the proposed framework

Feature Group	**Example Variables**	**Encoding Method**	**Output Form**
Environmental	Temperature, humidity, noise, air quality, ultraviolet exposure	Z-score or min–max normalization	d_E
Spatial	Coordinates, region, grid cell, PoI proximity, semantic place type	Numerical coordinates, distance features, region or PoI embeddings	d_S
Temporal	Hour of day, day of week, temporal bucket, sequence position	Cyclic encoding, temporal bucket embedding, sequence-derived features	d_T
User	Profile attributes, preferences, recent interactions, device-derived state	Dense profile vector, interaction summary, user embedding	d_U
Activity	Inferred activity, movement state, routine pattern, engagement state	Categorical encoding, activity embedding, sequence-derived features	d_A
PoI or Service	PoI identifier, category, availability, capacity, accessibility, service attributes	Identifier embedding, categorical encoding, normalized service-level features	d_P
Quality Metadata	Missingness, recency, source reliability, anomaly flags	Binary indicators, confidence scores, quality flags	d_Q

The resulting representation supports both branches of the later wide-and-deep model. The wide component can use sparse categorical identifiers and selected cross-features, while dense numerical features and learned embeddings can be used by the deep component. This preserves the link between preprocessing in Chapter 4 and the input representation adopted for prediction and recommendation in Chapter 5.

4.3.3 Temporal Alignment, Fusion, and Semantic Enrichment

After standardization and encoding, the pipeline aligns streams across time. Because the framework combines continuously sampled environmental data with irregular user interactions, service events, and platform updates, synchronization is performed through buffering, windowing, and resampling. These operations ensure that observations are compared within shared temporal intervals rather than mixed across stale and current states.

Once aligned, the streams are fused into composite context records. Fusion links environmental measurements, spatial descriptors, user state, activity indicators, temporal features, PoI attributes, and quality metadata into a unified representation. This stage is the point at which separate observations become structured contextual evidence for downstream prediction, recommendation, and DT updating.

Semantic enrichment assigns meaning to the fused records by associating measurements with entities, locations, time intervals, activities, and candidate services. A noise reading, for example, becomes more useful when linked to a place, time window, source, confidence level, and candidate user activity. This enrichment supports the context dimensions adopted in Section 4.2 and creates the feature-ready records required by the AI models developed later in the thesis.

4.3.4 Implementation Constraints and Final Handoff

The preprocessing pipeline is intended to operate incrementally as new observations arrive, but the section does not treat streaming as a separate architectural topic. Ingestion, validation, normalization, lightweight aggregation, and local feature preparation may occur continuously so that context updates remain responsive to changing urban conditions. This prevents preprocessing from becoming a bottleneck between sensing and higher-level intelligence.

Edge and cloud resources are used only as implementation locations for preprocessing tasks. Edge devices and gateways may handle local filtering, compression, preliminary aggregation, initial validation, and feature preparation close to the data source. Cloud resources may handle broader normalization, historical aggregation, fusion, semantic enrichment, storage, and training support.

The division depends on task complexity, latency requirements, bandwidth constraints, and deployment conditions.

Privacy-aware management is embedded in the same implementation logic. Since location traces, behavioral signals, user-device data, and activity-related observations may be sensitive, aggregation, anonymization, access control, and data-minimization steps are applied where required before broader storage or analysis. These operations constrain how much contextual detail is exposed while preserving the structured representations needed for context-aware intelligence.

The output of the preprocessing and management pipeline is a synchronized, quality-controlled, encoded, fused, and semantically enriched context record. This record forms the practical handoff from IoT acquisition to the AI and DT layers. For the AI layer, it supplies the normalized numerical features, embeddings, temporal encodings, PoI attributes, and quality indicators used for prediction and recommendation. For the DT layer, it supplies the updated environmental, spatial, user, activity, service, and quality states needed for synchronization. Preprocessing therefore establishes the concrete pathway through which raw heterogeneous urban observations become usable intelligence within the proposed framework.

4.4 Limitations of IoT-Derived Context

The IoT and context-modeling layer defines how heterogeneous urban observations become structured inputs for prediction, recommendation, and DT synchronization. Its reliability, however, is bounded by the conditions under which sensing, ingestion, preprocessing, and context construction operate. These limitations do not concern the conceptual value of IoT-derived context, but the practical boundary within which such context can be treated as complete, timely, interoperable, and transferable.

4.4.1 Coverage, Quality, and Uncertainty

The proposed framework depends on sufficiently broad and stable sensing coverage, yet real-world urban deployments are rarely uniform. Smart-city infrastructures combine heterogeneous sensing devices, communication protocols, and vendor-specific platforms, which makes integration into a unified context acquisition pipeline a complex and resource-intensive task [60, 96]. In practice, sensing installations often reflect local funding priorities, population density, administrative boundaries, or pilot-specific objectives. As a result, some areas may be densely monitored while others remain only partially observed. This uneven coverage affects the completeness and representativeness of inferred context states and can reduce the consistency of downstream recommendations and DT updates.

Coverage limitations are also operational. Large-scale deployments require sustained calibration, fault detection, firmware maintenance, device replacement, and coordination across public, private, and institutional infrastructures. At the city scale, managing heterogeneous sensing assets can create orchestration pressures that slow deployment and increase the cost of maintaining reliable context streams [115]. The effectiveness of the framework therefore depends not only on the proposed architecture, but also on the maturity, continuity, and spatial distribution of the sensing infrastructure that supplies it.

Data quality further constrains the reliability of IoT-derived context. Urban sensing data are noisy because of packet loss, calibration drift, variable sampling rates, intermittent connectivity, environmental interference, and device degradation. These issues become more significant when measurements from multiple sources must be temporally aligned and fused into a shared contextual representation. Low-cost environmental sensors can increase deployment reach, but their accuracy may vary under harsh urban conditions [63, 102]. Crowdsensing may improve spatial coverage, yet it can also introduce reliability variation, participation bias, and uneven representativeness [42].

This uncertainty can be represented by modeling an observed sensor value as

$$x_{j,t}^{obs} = x_{j,t}^{true} + \epsilon_{j,t}, \tag{4.10}$$

where $x_{j,t}^{obs}$ is the observed value of feature j at time t, $x_{j,t}^{true}$ is the unobserved true value, and $\epsilon_{j,t}$ represents measurement noise introduced by sensing, transmission, or calibration error. Since $x_{j,t}^{true}$ is not directly available, the framework treats noise, missingness, and confidence as part of the context representation rather than as external issues removed entirely during preprocessing.

For this reason, each feature may be associated with a quality or confidence score,

$$q_{j,t} \in [0, 1], \tag{4.11}$$

where lower values indicate missing, stale, noisy, or unreliable observations. The context representation can therefore be extended as

$$c_t = [n_t \| l_t \| u_t \| a_t \| q_t \| \delta_t], \tag{4.12}$$

where q_t contains quality or confidence scores and δ_t contains missingness indicators. This representation preserves the distinction between directly observed, reconstructed, stale, and uncertain values before they are passed to downstream AI and DT layers.

Although preprocessing can reduce obvious inconsistencies, uncertainty remains in higher-level context inference. Synchronization errors, unreliable readings, missing segments, and semantic

mismatches can propagate through fusion and affect the modeled context state. Prior work therefore emphasizes that robust context-aware systems require explicit handling of uncertainty rather than relying only on deterministic preprocessing pipelines [11, 22, 78]. In the proposed framework, missingness indicators, confidence scores, and quality-aware filtering define the reliability boundary of IoT-derived context while preserving compatibility with later prediction, recommendation, and DT synchronization.

4.4.2 Latency, Interoperability, and Transferability

A second limitation concerns the balance between latency, scalability, and contextual granularity. Urban environments generate high-volume and high-velocity data from sensors, mobile devices, infrastructure systems, and service platforms. Many context-aware services require timely updates to remain useful, but centralized processing may struggle to satisfy latency and bandwidth requirements at city scale [38, 115]. When update cycles are delayed, the resulting context vector may describe conditions that are no longer operationally relevant for recommendation or DT synchronization.

Edge and fog computing can reduce communication delay and network load by moving selected preprocessing and inference tasks closer to data sources. However, this distribution introduces additional coordination, consistency, monitoring, and fault-tolerance requirements. Distributed processing may also produce fragmented or partially inconsistent context representations when connectivity is unstable or when edge nodes operate with different local data windows [62]. The framework must therefore balance fine-grained responsiveness with computational feasibility. In latency-sensitive settings, this may require coarser aggregation, delayed synchronization, candidate filtering, or selective updating rather than continuous high-frequency context reconstruction.

Interoperability is closely related to this limitation because reliable context construction depends on combining observations across heterogeneous devices, platforms, formats, semantics, protocols, and administrative arrangements. Even when data can be collected from multiple sources, inconsistent schemas, incomplete mappings, and unstable update schedules can affect the reliability of context construction. This increases the burden on ingestion, preprocessing, synchronization, and semantic enrichment components and can weaken the operational coherence of the context pipeline [60, 78].

These interoperability constraints are especially important because the framework combines environmental, spatial, temporal, user-related, activity-related, and service-related information across layers. Where interoperability is limited, the architecture may remain conceptually extensible but operationally difficult to coordinate. In such cases, the effectiveness of context-aware reasoning

depends not only on the context model, but also on the ability to maintain semantically consistent data exchange across heterogeneous urban infrastructures.

Transferability across cities is also limited. IoT configurations and context models are shaped by local infrastructure, climate, sensing density, mobility patterns, service ecosystems, governance arrangements, and user behavior. Design assumptions and learned relationships that are appropriate in one urban environment may therefore require adaptation in another [30, 32]. Strong contextual dependence means that portability should be approached through recalibration rather than direct transfer [55].

The proposed system is therefore best understood as an adaptable architectural template rather than a plug-and-play solution. Deployment in a new city would require local calibration of sensing choices, preprocessing rules, context mappings, quality thresholds, latency-management strategies, and DT update policies. Together, sensing coverage, data quality, uncertainty, latency, interoperability, and transferability define the reliability boundary of IoT-derived context within the proposed framework.

4.5 Chapter Summary

This chapter defines how heterogeneous urban data are transformed into a structured context for the proposed IoT–AI–DT framework. Building on Chapter 3, it specified the acquisition, context modeling, preprocessing, and management functions required to convert environmental, infrastructural, spatial, temporal, user-related, and activity-related observations into semantically usable contextual inputs.

The main contribution of the chapter is the definition of an operational IoT-derived context layer that links raw urban sensing to downstream intelligence. In this layer, heterogeneous observations are acquired with their associated time, location, source, sampling, and quality metadata; organized into structured context dimensions; standardized, quality-controlled, aligned, fused, and semantically enriched; and finally represented as a model-ready context vector. This context vector provides the direct handoff to Chapter 5, where it is used for context-aware prediction and recommendation, and to Chapter 6, where it supports Human, PoI, and City or Area DT synchronization.

The chapter also defined the reliability boundary of this layer. Uneven sensing coverage, variable data quality, missingness, uncertainty, latency, and scalability trade-offs, interoperability constraints, and limited transferability across cities can affect how completely and reliably IoT-derived context represents current urban conditions. Overall, the chapter established the data-acquisition and context-structuring foundation required for the predictive models, recommender systems, and DT reasoning developed in the following chapters.

Chapter 5

AI Models for Context-Aware Prediction and Recommendation

Chapter 4 defined how heterogeneous urban observations are acquired, structured, preprocessed, and represented as model-ready context. This chapter uses that context representation as the input to the predictive and recommender layer of the proposed IoT–AI–DT framework. Its purpose is to define how structured environmental, spatial, temporal, activity-related, user-related, and PoI features are mapped into predictions and ranked recommendations.

The AI layer serves as the model anchor of the framework. Because the input space combines heterogeneous IoT-derived variables, dynamic user behavior, environmental conditions, and spatial-temporal constraints, the model must learn nonlinear relationships that are difficult to specify through fixed rules or heuristics [76, 78]. Within the proposed framework, AI supports two linked tasks. The first is contextual prediction, where the model estimates an outcome such as preference, comfort, productivity, or activity suitability under the current context. The second is recommendation, where candidate activities, services, or PoI are scored and ranked according to their predicted relevance for a user under those conditions. This formulation treats context as a core predictive signal rather than as a secondary filter, addressing the limitations of recommendation approaches that rely mainly on static preferences or historical co-occurrence patterns [2, 103].

The chapter centers on a context-aware wide-and-deep architecture. The wide component captures explicit and recurring feature interactions, while the deep component learns nonlinear relationships among user, PoI, environmental, spatial, temporal, and activity-related representations. This design supports both memorization of stable patterns and generalization across sparse or previously unseen contextual conditions, which is necessary for adaptive recommendation in dynamic urban settings [76, 103].

The outputs of the AI layer are not treated as final isolated recommendations. They provide predictive scores, ranked alternatives, and learned representations that are passed to the DT layer

for scenario-based reasoning and later evaluation. In this sense, Chapter 5 connects the structured context developed in Chapter 4 to the DT reasoning developed in Chapter 6, while establishing the learning model used for the quantitative comparison and integrated evaluation in later chapters.

The following sections define the prediction and recommendation problems, specify the loss functions used or available for the modeled tasks, describe the feature representation strategy, present the wide-and-deep architecture, outline the training configuration, and identify the baseline models used for comparative evaluation.

5.1 Problem Formulation

This section formalizes the prediction and recommendation problems addressed by the AI layer of the proposed IoT–AI–DT framework. The framework first transforms heterogeneous urban observations into a structured, model-ready context through acquisition, preprocessing, and context-modeling processes. This structured context is then used as input to predictive and recommender models that generate scores, rankings, and learned outputs for downstream DT reasoning and evaluation. Figure 5.1 situates this formulation within the model-development workflow: the problem definition specifies the prediction and recommendation targets, the structured context pipeline supplies the training data, the model-development stage instantiates the wide-and-deep architecture and baseline models, and the evaluation stage assesses predictive and recommendation performance.

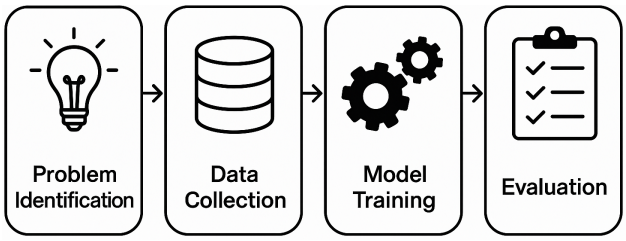

Figure 5.1: Recommender-system development workflow as used to organize the problem definition, data preparation, model development, and evaluation stages

5.1.1 Prediction and Recommendation Tasks

Let U denote the set of users. Each user $u \in U$ is associated with a dynamic user state vector h_t at time t, representing attributes derived from historical interactions, inferred preferences, recent behavior, and other user-related signals available to the model. The structured context vector c_t is the model-ready representation produced by the context-modeling layer. It may include environmental, spatial, temporal, activity-related, user-related, quality, and PoI or service-level features.

Environmental conditions are therefore treated as part of c_t, rather than as a separate vector. A separate environmental state n_t is not introduced here, unless future extensions explicitly define it as a distinct system-level urban state outside the context vector.

The prediction task estimates an outcome for a user under the current structured context. The model learns a function

$$f_\theta : (h_t, c_t) \rightarrow \hat{y}_t, \tag{5.1}$$

where $\hat{y}_t$ denotes the predicted outcome and θ denotes the trainable model parameters. Depending on the application setting, $\hat{y}_t$ may represent a continuous target such as a contextual preference score, expected comfort, productivity-related outcome, or activity suitability. The observed target y_t is obtained from historical logs, recorded outcomes, or observed behavior. The prediction objective is to learn parameters that minimize the difference between observed and predicted outcomes over the training data.

The recommendation task extends the same input representation by introducing a candidate set $\mathcal{I}$ of activities, services, actions, or PoI. For a user u, context c_t, and candidate item $i \in \mathcal{I}$, the recommender learns a scoring function

$$g_\theta : (h_t, c_t, i) \rightarrow s_{u,i,t}, \tag{5.2}$$

where $s_{u,i,t}$ denotes the predicted relevance or suitability of candidate i for user u at time t. Recommendations are produced by scoring the candidate set and returning the top-k ranked items:

$$R_t = \mathrm{TopK}_{i \in \mathcal{I}} \left(s_{u,i,t} \right). \tag{5.3}$$

This formulation treats contextual variables as direct model inputs rather than as post-processing filters, allowing environmental, spatial, temporal, activity-related, user-related, and PoI-level features to affect both prediction and ranking.

The training data for prediction are represented as tuples

$$\mathcal{D}_{\text{pred}} = \{(h_t, c_t, y_t)\}_{t=1}^{N}, \tag{5.4}$$

where y_t is the observed prediction target. The training data for recommendations is represented as tuples

$$\mathcal{D}_{\text{rec}} = \{(h_t, c_t, i_t, r_t)\}_{t=1}^{N}, \tag{5.5}$$

where i_t is an observed candidate item, service, activity, or PoI, and r_t denotes the supervisory signal associated with that candidate. These tuples are extracted from historical interaction logs

after the upstream sensing, preprocessing, synchronization, and context-construction stages have already been completed.

Temporal and spatial dependencies enter the formulation through h_t and c_t. The user state h_t is not treated as a static profile; it may be updated from prior interactions and contextual observations. The context vector c_t includes spatial and temporal components that represent location, accessibility, proximity to candidate PoIs, time-dependent patterns, and other contextual conditions. This allows the model to learn user-context-item relationships directly within the prediction and recommendation functions.

Prediction and recommendation are related but analytically distinct. Prediction estimates an outcome under a given user state and context, while recommendation scores and ranks candidate actions, services, or PoIs. Prediction outputs may support recommendations, and recommendation outcomes may later influence user state through feedback. They are nevertheless defined separately to preserve modularity in task formulation and evaluation. Both tasks share the same structured context pipeline and provide outputs that can be propagated to the DT layer for scenario-based reasoning and integrated evaluation.

5.1.2 Training Objectives and Loss Functions

From an optimization perspective, the learning problem is expressed as empirical risk minimization. For the continuous prediction setting used in the experimental model, the objective is

$$\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \mathcal{L}_{\text{pred}} \left(y_t, f_\theta(h_t, c_t) \right). \tag{5.6}$$

This objective is defined independently of a specific architecture so that the same formulation can support the wide-and-deep model introduced later and the baseline models used for comparative evaluation.

For continuous prediction tasks, including the estimation of contextual preference scores, comfort-related outcomes, productivity-related outcomes, or activity suitability, the model is trained using mean squared error. Given N observed training examples, the prediction loss is defined as

$$\mathcal{L}_{\text{pred}} = \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \hat{y}_n \right)^2, \tag{5.7}$$

where y_n is the observed target value and $\hat{y}_n$ is the predicted output produced by the model. In this study, mean squared error is the actual training loss minimized during model optimization. This choice is aligned with the continuous-score prediction formulation. It directly supports evaluation

through root mean squared error, which reports the prediction error in the same scale as the target variable.

The recommendation component uses the learned prediction scores to order candidate activities, services, actions, or PoIs for each user-context state. In this setting, recommendation is therefore treated as a score-based ranking step derived from the model output rather than as a separately optimized pairwise ranking objective. The ranking R_t is obtained by applying the top-k operation to the predicted candidate scores. At the same time, predictive accuracy is trained through the MSE objective in Equation 5.7 and assessed using the corresponding RMSE-based evaluation.

Together, the task definitions and training objective establish the technical role of the AI layer within the thesis pipeline. Structured context is combined with dynamic user state and candidate PoI or service representations; the AI model predicts continuous outcomes and scores candidate options; the ranked outputs are then available for DT-based reasoning and evaluation. This positions the AI layer as the bridge between IoT-derived context modeling and downstream DT decision support.

5.2 Feature Representation

The feature representation layer defines how structured context enters the prediction and recommendation model developed in this chapter. The model consumes preprocessed and synchronized features rather than raw IoT streams. This preserves the architectural separation between context construction and learning: sensing, alignment, missingness handling, and encoding are completed upstream, while the resulting representation is used here for prediction and recommendation.

The feature space is organized around the requirements of the proposed wide-and-deep architecture. Sparse categorical identifiers and selected feature crosses are routed to the wide branch to support memorization of recurring user-context-PoI patterns. Dense numerical variables and learned embeddings are routed to the deep branch to support generalization across unseen combinations of users, activities, locations, and environmental states [24, 112]. This representation links the context-aware problem formulation in Section 5.1 to the architecture specified in Section 5.3.

5.2.1 User, Activity, and PoI Features

User features provide the personalization basis of the model. They include historical interactions, preference indicators, recent behavior, and user-state variables derived from the preprocessing pipeline. Interaction features include visits, clicks, ratings, service usage logs, and implicit feedback, which capture behavioral regularities used to predict future choices [80, 112]. The representation

distinguishes stable user information, such as long-term preferences or profile attributes, from dynamic user state, such as recent interactions, current location, inferred activity, and device-derived signals.

Activity features describe the behavioral condition under which a recommendation is generated. They include activity category, movement state, routine pattern, engagement state, and sequence-derived indicators where available. These variables allow the model to represent not only who the user is, but also what the user is doing or is likely to do next. Recency-based summaries and temporal aggregation are used to preserve short-term intent while retaining longer-term behavioral structure [72, 104].

PoI and service-level features describe the candidate entities ranked by the recommender. They include PoI or service identifier, category, popularity, accessibility, operating hours, capacity, demand history, and service availability. Dynamic service features, such as occupancy, waiting time, congestion, and temporal popularity, allow candidate services to be represented as changing urban resources rather than static items [64, 106]. Where relational information is available, graph-derived attributes may also represent links among users, PoIs, activities, and regions [69, 74, 109].

5.2.2 Environmental and Spatio-Temporal Features

Environmental features represent external conditions that influence activity suitability, comfort, mobility choices, and service engagement. In the proposed framework, these features are derived from the IoT-based context layer and include temperature, humidity, noise level, air quality, ultraviolet exposure, weather-related signals, and crowd-density indicators where available [78, 102]. Their role in this section is to specify how environmental context becomes model input rather than to repeat the earlier context-modeling discussion.

Continuous environmental variables are encoded as normalized numerical features. When observations are incomplete or uncertain, missingness and quality indicators from the upstream context layer can be retained as auxiliary inputs. This allows the model to distinguish observed values from imputed or lower-confidence values and preserves the link between sensing quality and downstream prediction [11, 78].

Spatial features include geographic coordinates, region identifiers, spatial grid cells, distance to candidate PoIs, semantic place type, and neighborhood-level descriptors. These features help constrain recommendations to feasible and contextually appropriate options, especially where service availability and user mobility are location dependent [9, 32]. Temporal features include hour of day, day of week, weekend indicators, seasonal indicators, recency features, sliding-window summaries, and temporal buckets. These features preserve recurring behavioral and service-demand patterns

while supporting alignment between user interactions, environmental observations, and PoI states [38, 78, 115].

Spatio-temporal variables may also be combined as feature crosses when they represent recurring patterns that should be memorized. Examples include region-activity, time-activity, user-region, and PoI-time combinations. These crosses are routed to the wide branch, while their embedded or numerical forms are routed to the deep branch for more flexible representation learning.

5.2.3 Encoding for Wide-and-Deep Learning

The proposed model separates the input space into sparse and dense representations. Sparse features include user identifiers, PoI or service identifiers, activity categories, spatial regions, temporal buckets, and selected feature crosses. The wide branch uses these features to memorize frequent co-occurrence patterns, including user-activity, user-PoI, time-activity, region-activity, and context-PoI associations.

Dense features include normalized environmental variables, numerical spatial features, distance measures, user-state summaries, PoI attributes, and aggregated behavioral statistics. These features are provided to the deep branch together with learned embeddings for high-cardinality categorical inputs. User identifiers, PoI identifiers, activity categories, spatial regions, and temporal buckets are mapped into trainable embedding vectors, allowing sparse symbolic variables to be represented in a continuous latent space [28, 112].

The two branches therefore receive different views of the same structured context. The wide branch receives explicit sparse indicators and feature crosses for memorization, while the deep branch receives normalized dense variables and embeddings for generalization. The fused output produces the prediction or recommendation score used by the downstream ranking process.

Overall, this section defines the handoff from the context-construction layer to the model architecture in Section 5.3. Environmental, spatial, temporal, activity, user, and PoI-related context are converted into the sparse and dense representations required by the proposed wide-and-deep model, supporting prediction, recommendation, and later DT-based reasoning.

5.3 Model Architecture

5.3.1 Architecture Rationale and Overview

The proposed model adopts a context-aware wide-and-deep architecture for prediction and recommendation in smart urban environments. The architecture is designed to operate on model-ready

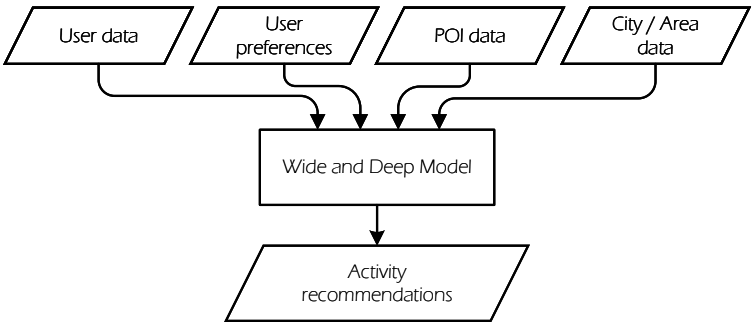

Figure 5.2: Hybrid wide-and-deep learning architecture for personalized activity recommendation using multisource contextual data

representations from the upstream IoT and preprocessing layers, including user, activity, PoI, environmental, spatial, temporal, and quality-related features. These inputs are heterogeneous: some are sparse categorical identifiers, some are engineered feature crosses, and others are dense numerical variables derived from IoT-based context construction. A single linear or neural model would not capture these feature types equally well. Linear components are effective for memorizing recurring feature co-occurrences, while deep neural components are better suited for learning nonlinear relationships across sparse and heterogeneous contextual inputs. For this reason, this thesis adopts the wide-and-deep paradigm [24].

The wide component captures explicit and historically stable interactions, such as user–PoI, user–activity, time–activity, region–activity, and context–PoI patterns. These interactions are important in urban recommendation because many decisions recur under similar contextual conditions, for example, when a user repeatedly selects a particular activity type in a specific region or time window. The deep component learns latent representations from embeddings and dense numerical features, allowing the model to generalize beyond observed feature crosses and to represent nonlinear relations among users, PoIs, environmental conditions, spatial constraints, temporal patterns, and activity states. This combination follows the memorization and generalization logic of wide-and-deep recommender systems [24], while adapting it to the IoT-derived context and DT pipeline developed in this thesis.

Figure 5.2 shows the role of the model within the proposed recommendation workflow. User data, user preferences, PoI data, and city or area context are encoded as structured inputs and passed to the wide-and-deep model. The output is an activity recommendation or prediction score that can be used for ranking and subsequently propagated to the DT layer for scenario-based reasoning. The architecture therefore functions as the learning bridge between structured context construction and DT-based reasoning.

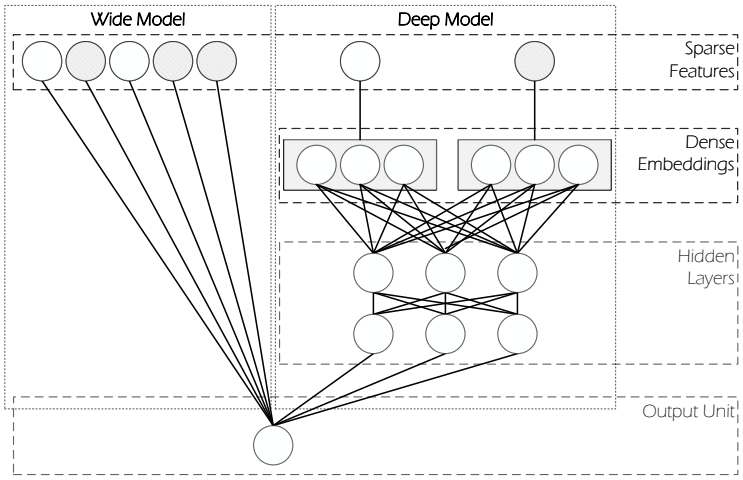

Figure 5.3: Wide-and-Deep Model

5.3.2 Wide-and-Deep Branch Specification

As illustrated in Figure 5.3, the model consists of two parallel branches that operate on different views of the same structured context. The wide branch receives sparse categorical indicators and engineered feature crosses, while the deep branch receives dense numerical variables and learned embeddings. The outputs of the two branches are fused to produce the final prediction or recommendation score.

Let $\mathbf{x}_{wide}$ denote the sparse input vector supplied to the wide branch. This vector contains one-hot or multi-hot categorical indicators and selected cross-features:

$$\mathbf{x}_{wide} = [\mathbf{x}_{sparse} \| \mathbf{x}_{cross}], \tag{5.8}$$

where $\mathbf{x}_{sparse}$ represents categorical indicators and $\mathbf{x}_{cross}$ represents engineered feature crosses. The wide branch computes a scalar output:

$$o_{wide} = \mathbf{w}_{wide}^{T} \mathbf{x}_{wide} + b_{wide}, \tag{5.9}$$

where $\mathbf{w}_{wide}$ and b_{wide} are trainable parameters. This component allows the model to memorize frequent and meaningful associations among users, locations, time periods, activity categories, and PoIs.

The deep branch learns nonlinear latent representations from embeddings and dense numerical features. High-cardinality categorical variables, including user identifiers, PoI identifiers, activity categories, spatial regions, and temporal buckets, are mapped into trainable embedding vectors. For

a user identifier u, candidate PoI or service i, activity category a, spatial region s, and temporal bucket τ, the embedding lookups are defined as

$$\mathbf{v}_u = \mathbf{E}_u[u], \quad \mathbf{v}_i = \mathbf{E}_i[i], \quad \mathbf{v}_a = \mathbf{E}_a[a], \quad \mathbf{v}_s = \mathbf{E}_s[s], \quad \mathbf{v}_\tau = \mathbf{E}_\tau[\tau], \tag{5.10}$$

where $\mathbf{E}_u$, $\mathbf{E}_i$, $\mathbf{E}_a$, $\mathbf{E}_s$, and $\mathbf{E}_\tau$ are trainable embedding matrices. Embedding-based representations are widely used in neural recommender systems because they reduce sparsity and allow latent similarities among users, items, and contextual states to be learned directly from data [51, 112].

The embedding dimensions used in the implemented model are

$$\mathbf{v}_u \in \mathbb{R}^{16}, \quad \mathbf{v}_i \in \mathbb{R}^{16}, \quad \mathbf{v}_a \in \mathbb{R}^{8}, \quad \mathbf{v}_s \in \mathbb{R}^{8}, \quad \mathbf{v}_\tau \in \mathbb{R}^{8}. \tag{5.11}$$

Let $\mathbf{x}_{num}$ denote the normalized continuous feature vector. This vector includes environmental measurements, distance features, numerical user-state features, PoI or service attributes, and other dense variables produced by the preprocessing pipeline. Concatenating embeddings and numerical features forms the input to the deep branch:

$$\mathbf{x}_{deep} = [\mathbf{v}_u \| \mathbf{v}_i \| \mathbf{v}_a \| \mathbf{v}_s \| \mathbf{v}_\tau \| \mathbf{x}_{num}], \tag{5.12}$$

where $\|$ denotes vector concatenation.

The deep branch consists of three fully connected hidden layers. Given $\mathbf{h}_0 = \mathbf{x}_{deep}$, the representation at hidden layer l is computed as

$$\mathbf{h}_l = \mathrm{ReLU}\left(\mathbf{W}_l \mathbf{h}_{l-1} + \mathbf{b}_l\right), \quad l \in \{1, 2, 3\}, \tag{5.13}$$

with hidden layer sizes

$$\mathbf{h}_1 \in \mathbb{R}^{128}, \quad \mathbf{h}_2 \in \mathbb{R}^{64}, \quad \mathbf{h}_3 \in \mathbb{R}^{32}. \tag{5.14}$$

Through these transformations, the deep branch learns nonlinear dependencies among user state, PoI attributes, temporal patterns, spatial structure, environmental context, and activity-related conditions [75].

5.3.3 Context Integration and Model Configuration

Contextual information is incorporated directly into both branches of the architecture. In the wide branch, context appears through explicit feature crosses that represent repeated user-context-PoI

patterns. Examples include user–activity, user–PoI, time–activity, region–activity, and context–PoI combinations. This allows the model to retain stable associations supported by historical observations.

In the deep branch, context appears through normalized continuous variables and learned embeddings. Environmental variables such as temperature, humidity, noise level, air quality, and ultraviolet exposure are entered as normalized numerical features. Spatial and temporal variables enter through distance features, region embeddings, temporal bucket embeddings, and other encoded representations. User-state and PoI-level attributes are also included in the dense input vector. This design follows the principle of contextual modeling in context-aware recommender systems, where contextual variables are incorporated into the learning function itself rather than applied only through pre-filtering or post-filtering mechanisms [1]. It also allows context to influence both memorized interactions and learned latent representations [57].

Table 5.1: Architecture specification of the proposed context-aware wide-and-deep model

Component	**Specification**
Wide branch	Linear component over sparse interaction and cross-features
Identifier embeddings	16 dimensions
Categorical context embeddings	8 dimensions
Deep branch	Fully connected layers with sizes 128, 64, and 32
Activation function	ReLU after each hidden layer
Dropout	0.0

Table 5.1 summarizes the implemented architecture of the proposed context-aware wide-and-deep model. These settings align the architecture with the training configuration described later in this chapter while keeping the model compact enough for repeated evaluation and comparison against baselines.

Table 5.2 summarizes how the model receives the structured feature groups defined in Section 5.2. Sparse identifiers and feature crosses support the wide branch, while embeddings and normalized continuous variables support the deep branch. This keeps the model architecture consistent with the upstream context representation and avoids duplicating the preprocessing logic already defined there.

5.3.4 Fusion, Output, and Architectural Alignment

The fusion layer combines the memorization capacity of the wide branch with the representation-learning capacity of the deep branch. Let o_{wide} be the scalar output of the wide branch and let $\mathbf{h}_3$

Table 5.2: Input encoding strategy for the proposed AI model

Feature Type	**Encoding Method**	**Output Dimension**
User identifier	Embedding lookup	16
PoI or service identifier	Embedding lookup	16
Activity category	Embedding lookup	8
Spatial region or grid cell	Embedding lookup	8
Temporal bucket	Embedding lookup	8
Environmental variables	Normalized continuous values	d_E
User-state variables	Normalized or dense profile vector	d_U
PoI/service attributes	Normalized numerical or categorical encoding	d_P
Feature crosses	Sparse one-hot or multi-hot encoding	d_W

be the final hidden representation of the deep branch. For a continuous prediction task, the fused output is computed as

$$\hat{y}_{u,i,t} = o_{wide} + \mathbf{w}_{deep}^T \mathbf{h}_3 + b, \tag{5.15}$$

where $\mathbf{w}_{deep}$ and b are trainable output-layer parameters. This output is used when the target is represented as a continuous outcome, such as a contextual preference score, activity suitability score, or productivity-related estimate.

For implicit-feedback recommendation, the scalar output is passed through a sigmoid function to estimate relevance:

$$s_{u,i,t} = \text{sigm}\left(o_{wide} + \mathbf{w}_{deep}^T \mathbf{h}_3 + b\right), \tag{5.16}$$

where $s_{u,i,t} \in [0, 1]$ represents the predicted relevance of item, activity, or PoI i for user u under context t. For ranking-based recommendation, the model computes a score for each candidate item $i \in I$ and returns the top-k candidates:

$$R_t = \text{TopK}_{i \in \mathcal{I}}\left(s_{u,i,t}\right). \tag{5.17}$$

The same fused architecture therefore supports both prediction and recommendation. Equation 5.15 is used for continuous prediction, while Equation 5.16 and Equation 5.17 are used for relevance estimation and candidate ranking. The loss function determines how the fused output is optimized, as defined in Section 5.1. Continuous prediction uses the regression objective, while implicit-feedback recommendation can use binary cross-entropy or a ranking-oriented loss when the experiment is framed as pairwise ranking.

The architecture also provides a consistent interface for downstream DT reasoning. The model

receives structured context from the IoT and preprocessing layers, transforms it into prediction and recommendation scores, and outputs ranked candidate activities or PoIs. The DT layer can then use these outputs to evaluate alternatives through scenario simulation. In this sense, the wide-and-deep model is not a detached recommender component. It is the predictive bridge between a structured urban context and the DT-based decision-support layer.

Scalability is addressed through the structure of the model rather than through a separate deployment discussion. Sparse linear computation in the wide branch can be parallelized efficiently, while the embedding and dense layers of the deep branch can use GPU acceleration. Embeddings also reduce the dimensionality of high-cardinality categorical variables, which supports learning over large user, PoI, activity, spatial, and temporal spaces. Similar decompositions are used in large-scale recommender systems to support efficient inference under heterogeneous feature conditions [28]. The resulting architecture remains modular: wide features, dense features, embeddings, fusion, and output scoring can be updated or extended without changing the overall modeling pipeline.

5.4 Training Configuration and Optimization

The proposed context-aware wide-and-deep model is trained after the IoT-derived observations from Chapter 4 have been transformed into synchronized, encoded, and quality-controlled feature representations. The training process therefore operates on structured user, activity, PoI, environmental, spatial, and temporal inputs rather than on raw sensor streams. The objective is to select a model that can map this structured context to prediction and recommendation outputs that are subsequently available for DT reasoning and evaluation.

The trainable parameters include the wide-branch weights, deep-network weights, embedding matrices, and output-layer parameters. The complete parameter set is defined as

$$\Theta = \{\mathbf{w}_{wide}, b_{wide}, \mathbf{E}_u, \mathbf{E}_i, \mathbf{E}_a, \mathbf{E}_s, \mathbf{E}_\tau, \mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{w}_{deep}, b\}. \tag{5.18}$$

The reported models were trained as explicit-feedback rating predictors using mean squared error loss, as defined in Equation 5.7. The objective minimizes the squared difference between the observed user–PoI-event rating and the predicted rating. Model selection is based on validation RMSE, while final performance is evaluated using RMSE, MAE, R^2, and explained variance.

Model parameters are learned using mini-batch gradient-based optimization. For each mini-batch, the mean squared error loss is computed, and the parameters are updated using the Adam optimizer:

$$\Theta^{(k+1)} = \text{AdamUpdate}\left(\Theta^{(k)}, \nabla_\Theta \mathcal{L}, \eta\right), \tag{5.19}$$

where k denotes the optimization step, η is the learning rate, and $\mathcal{L}$ is the selected training loss.

The final training configuration is summarized in Table 5.3. The table reports the optimization, regularization, validation, and execution settings used to train the proposed model. Architectural settings are retained in Table 5.1, while the hyperparameter search space is reported separately in Table 5.4.

Table 5.3: Training configuration for the proposed context-aware wide-and-deep model

Parameter	**Setting**
Optimizer	Adam
Learning rate	5×10^{-4}
Batch size	128
Regularization	L2 weight decay of 10^{-5}
Training loss	Mean squared error
Validation strategy	Train, validation, and test split with early stopping based on validation RMSE
Maximum epochs	300
Early stopping patience	30 epochs
Hardware	NVIDIA GeForce RTX 2070 GPU

Early stopping is applied to select the model checkpoint that generalizes best under the validation criterion. Training stops when validation performance does not improve for 30 consecutive epochs, and the selected model is the checkpoint with the lowest validation RMSE. Under this strategy, the best validation configuration reached an RMSE of 0.3685 at epoch 72. This checkpoint is retained for subsequent prediction, recommendation, and DT-oriented evaluation because it provides the selected mapping from structured context to model outputs.

The validation set is also used for hyperparameter selection. The search focuses on parameters that affect representation capacity, regularization, and optimization behavior, while model selection is based on the lowest validation RMSE under the mean squared error training objective. The evaluated search space is summarized in Table 5.4.

The resulting selection procedure trains the proposed model on the structured context representation produced by the IoT and context-modeling pipeline, evaluates candidate configurations using validation RMSE, and retains the best-performing checkpoint for downstream prediction, recommendation ranking, and DT-based reasoning.

Table 5.4: Hyperparameter search space used for the proposed context-aware wide-and-deep model

Hyperparameter	**Search Values**
User/item embedding dimension	16, 64
Categorical context embedding dimension	8, 16
Hidden layers	[128, 64, 32], [256, 128, 64]
Dropout rate	0.00, 0.10
Learning rate	0.0010, 0.0005, 0.0001
Weight decay	0.0000, 0.000001, 0.00001
Batch size	64, 128, 256, 4096
Selection criterion	Lowest validation RMSE

5.5 Baseline Models for Comparative Evaluation

The comparative evaluation uses collaborative filtering, matrix factorization, and neural collaborative filtering as interaction-based baselines aligned with the quantitative evaluation in Chapter 7. These baselines isolate whether the proposed context-aware wide-and-deep model improves prediction and recommendation performance by using the structured context as a direct input. The comparison therefore tests the value of integrating environmental, spatial, temporal, user-state, activity-related, and PoI-level features, rather than relying primarily on user-item interactions.

The purpose of this comparison is not to restate the general development of recommender systems, but to test whether direct context integration improves prediction and recommendation performance over methods that rely primarily on user-item interactions. The resulting outputs support the later DT reasoning and evaluation pipeline by producing predictive scores and ranked candidate activities, services, or PoIs. In addition to these external baseline models, the evaluation also uses ablation baselines to separate the contribution of contextual representation, personalization, and DT-related features within the proposed framework.

5.5.1 Collaborative Filtering Baseline

The collaborative filtering baseline estimates user preference from observed user-item interaction tendencies. In the bias-aware form used for comparison, the predicted score is defined as

$$\hat{r}_{u,i} = \bar{r} + b_u + b_i, \tag{5.20}$$

where $\bar{r}$ is the global mean interaction score, b_u is the user-specific bias, and b_i is the item-specific bias. This baseline provides an interaction-only reference point and does not include environmental, spatial, temporal, activity-related, or user-state context.

5.5.2 Matrix Factorization Baseline

The matrix factorization baseline extends the interaction-only comparison by representing users and items through latent vectors. The predicted score is defined as

$$\hat{r}_{u,i} = \alpha_u^T \beta_i + b_u + b_i + \bar{r}, \tag{5.21}$$

where $\alpha_u \in \mathbb{R}^{d_{\mathrm{MF}}}$ is the latent vector for user u, $\beta_i \in \mathbb{R}^{d_{\mathrm{MF}}}$ is the latent vector for item i, and d_{MF} is the latent factor dimension. This baseline captures latent user-item structure, but it does not directly model the changing contextual conditions that affect recommendation relevance in the proposed framework.

5.5.3 Neural Collaborative Filtering Baseline

The neural collaborative filtering baseline replaces the matrix factorization inner product with a learned nonlinear interaction function. User and item embeddings are concatenated and passed through a multilayer neural network to produce a continuous rating prediction:

$$\hat{r}_{u,i} = \mathbf{w}_N^T \mathrm{NCF}\left([n_u \| n_i]\right) + b_N, \tag{5.22}$$

where n_u and n_i are user and item embeddings, $\mathrm{NCF}(\cdot)$ denotes a multilayer neural network, $\mathbf{w}_N$ is the output-layer weight vector, and b_N is the output-layer bias. In the implemented baseline, the model is formulated as an explicit-feedback regression model and is trained using mean squared error against the observed 1–5 rating targets. During validation and testing, predicted values are clipped to the valid rating interval:

$$\hat{r}_{u,i}^{\,\mathrm{final}} = \min\left(5, \max\left(1, \hat{r}_{u,i}\right)\right). \tag{5.23}$$

This baseline tests whether nonlinear user–item interaction modeling is sufficient without the full structured context used by the proposed model.

5.5.4 Proposed Context-Aware Wide-and-Deep Model

The proposed model differs from the baselines by using the structured context vector pipeline as a direct model input. Its scoring function is defined as

$$s_{u,i,t} = g_\theta \left(h_t, c_t, i \right), \tag{5.24}$$

where h_t represents the dynamic user state, c_t represents the structured context vector containing environmental, spatial, temporal, user-related, activity-related, quality, and PoI-level features, and i is the candidate item, service, activity, or PoI. The model therefore evaluates candidates using both interaction information and IoT-derived context, producing scores and rankings that can be passed to the DT layer for scenario-based reasoning and integrated evaluation.

The comparison between the interaction-based baselines and the proposed model is summarized in Table 5.5.

Table 5.5: Baseline models used for comparative evaluation

Model	**Primary Inputs**	**Comparison Role**
Collaborative Filtering	User-item interactions	Tests interaction-only bias-based prediction.
Matrix Factorization	User and item latent factors	Test the latent user-item structure without explicit context.
Neural CF	User and item embeddings	Tests nonlinear interaction modeling without full contextual integration.
Proposed Context-Aware W&D	User, PoI, environmental, spatial, temporal, activity, and user-state context	Tests whether structured context improves prediction and recommendation outputs for DT reasoning.

These baselines isolate the central modeling question of this chapter: whether context-aware wide-and-deep learning improves over interaction-only collaborative, latent-factor, and neural collaborative approaches when structured IoT-derived context is used as a core predictive input rather than as an auxiliary post-processing signal.

5.5.5 Ablation Study Configurations

The baseline comparison in Table 5.5 evaluates the proposed model against established interaction-based recommender models. The ablation comparison has a different role. It examines how the proposed framework changes when specific components are removed or restricted. This makes it possible to separate the contribution of contextual representation, user-item personalization, and DT-related features within the integrated evaluation.

Table 5.6 defines the ablation configurations used for this purpose. The comparison identifies which component is removed or limited in each ablation so that the performance differences reported later in Table 7.4 can be interpreted directly.

Table 5.6: Ablation configurations used to isolate predictor and framework components

Model	Description
Non-context ablation	Uses user–item, user–activity, or user–PoI interaction features only. Environmental, spatial, temporal, and activity context variables are excluded.
Context-only ablation	Uses contextual variables for prediction and recommendation, but does not include interaction features or DT-related features.
No-DT-features ablation	Uses the same user, item, and contextual inputs as the full model but excludes the engineered DT-related features.
Proposed full model	Uses IoT-derived context, context-aware wide-and-deep recommendation, and DT-related features.

The ablation baselines complement the interaction-based baselines by shifting the comparison from external recommender models to internal framework components. Collaborative filtering, matrix factorization, and neural collaborative filtering test whether structured IoT-derived context improves over interaction-based modeling. In contrast, the ablation study tests how contextual features, personalization features, and DT related engineered match features contribute to predictive performance

5.6 Chapter Summary

This chapter defined the AI layer of the proposed IoT–AI–DT framework as the model anchor for context-aware prediction and recommendation. Building on the structured context representation developed in Chapter 4, it formalized how user state, environmental conditions, spatial and temporal context, activity-related information, and PoI characteristics are mapped into predicted outcomes and ranked candidate recommendations.

The chapter specified the feature representation used by the model, separating sparse identifiers and feature crosses from dense numerical variables and learned embeddings. This representation supports the proposed wide-and-deep architecture, in which the wide branch captures recurring user–context–PoI interactions and the deep branch learns nonlinear relationships across heterogeneous contextual inputs. The training configuration and baseline models were also defined to support the comparative evaluation developed in Chapter 7.

Overall, the chapter establishes the transition from structured context to predictive and recommender outputs. These outputs are not treated as isolated rankings; they provide the scores, candidate alternatives, and learned representations that feed the DT layer for scenario-based reasoning and subsequent evaluation.

Chapter 6

Digital Twin Modeling

Building on the context-aware prediction and recommendation models developed in Chapter 5, this chapter positions DT modeling as the scenario-evaluation layer of the proposed framework. The AI layer produces ranked activities, services, and PoIs, but these outputs are not treated as final decisions. They become candidate actions for DT-based scenario evaluation and productivity-oriented decision support [14, 41].

The chapter focuses on three interacting DT entities: the Human DT, the PoI DT, and the City or Area DT. The Human DT represents user preferences, behavior, activity context, and relevant contextual or physiological states. The PoI DT represents localized service, availability, capacity, accessibility, and environmental conditions. The City or Area DT represents broader spatial, infrastructural, and environmental constraints. Together, these entities connect individual needs with urban-scale context while preserving distinct user-level, place-level, and area-level representations [68, 73, 84].

Interaction among these twins allows recommendations to be evaluated as scenarios rather than as isolated rankings. For each candidate action, the framework assesses how the user's state, PoI conditions, and city or area constraints jointly affect expected relevance, feasibility, comfort, access burden, service availability, and productivity alignment. The resulting output supports advisory decision-making by comparing alternatives under current contextual conditions.

The chapter also defines the limits of DT-based simulation. Scenario outputs depend on IoT-derived context quality, twin fidelity, synchronization assumptions, sensing coverage, and the behavioral or environmental assumptions embedded in the simulation. The DT layer is therefore treated as a comparative decision-support mechanism rather than an autonomous decision-maker. Its role is to extend Chapter 5 recommendations into scenario-based evaluation and productivity-oriented advisory output within the integrated IoT–AI–DT framework [110].

6.1 Digital Twin Entities

6.1.1 Entity Decomposition

In the proposed framework, the DT layer extends the AI recommendations generated in Chapter 5 into scenario-based evaluation and decision-support output. The recommender model identifies candidate activities, services, or PoIs, but these candidates must be evaluated against user state, localized service conditions, and broader urban constraints before they can support action. For this reason, the DT layer is decomposed into three interacting entities: the Human DT, the PoI DT, and the City or Area DT.

This decomposition reflects the different spatial scales, temporal dynamics, and data types involved in urban decision support. Human state may change rapidly with activity, location, preference, and recent interaction patterns. PoI state changes through operational conditions such as availability, occupancy, congestion, and service status. City or area state changes at a broader scale through traffic, environmental exposure, infrastructure status, and aggregate demand. Treating these components as separate but connected entities supports scenario evaluation without reducing the urban decision process to a single monolithic representation [30, 92].

Within this structure, the Human DT represents the user-side decision context, the PoI DT represents the candidate option state, and the City or Area DT represents the surrounding urban constraints. Their interaction enables AI-generated recommendations to be compared in terms of feasibility, contextual suitability, access burden, environmental exposure, and productivity alignment.

6.1.2 Human Digital Twin

The Human DT represents the user state required for personalized scenario evaluation. It is a constrained abstraction rather than a complete personal replica, maintaining only the features needed for prediction, recommendation, and decision support. These features include preferences, recent behavior, mobility context, inferred activity, interaction history, and relevant contextual or physiological signals where available [68].

Its state is updated from the sensing and context-modeling layers. Mobile and wearable devices may provide activity-related or physiological signals, IoT infrastructure may provide environmental and spatial context, and interactions with services or digital platforms may provide evidence of preference, intent, or behavioral response [15, 60]. These inputs allow the Human DT to represent the user's current decision situation rather than relying only on static profile information.

In the simulation workflow, the Human DT evaluates each candidate action from the user side. A recommended PoI is therefore assessed not only by predicted preference score, but also by its

alignment with the user's current activity, comfort requirements, time constraints, and productivity-oriented goals. This makes the Human DT the individualized evaluation component of the DT layer.

6.1.3 Point-of-Interest Digital Twin

The PoI DT represents the localized state of each candidate option. It models discrete urban entities such as workplaces, libraries, transport hubs, commercial venues, healthcare facilities, recreational sites, and other service locations that may appear in the recommendation set. Its role is to determine whether a recommended candidate is operationally feasible and contextually suitable under current conditions.

A PoI DT combines static and dynamic attributes. Static attributes include location, category, function, capacity, accessibility, and service type. Dynamic attributes include occupancy, congestion, availability, waiting time, operational status, localized environmental conditions, and inferred service quality. These dynamic attributes are updated through sensors, service logs, platform interactions, or inferred usage patterns, depending on data availability and sensing coverage.

Within the proposed framework, the PoI DT prevents candidate recommendations from being treated as static items. A candidate with a high AI score may be assessed less favorably in the scenario analysis if its PoI twin indicates crowding, limited availability, high noise, or poor accessibility. Conversely, a lower-ranked candidate may be identified as more suitable under the scenario criteria.

6.1.4 City or Area Digital Twin

The City or Area DT represents the broader urban context in which user and PoI decisions occur. It operates at a meso- or macro-scale by integrating spatial, infrastructural, environmental, and aggregate behavioral conditions across neighborhoods, districts, or functional zones [30, 92]. Its purpose in this framework is not to reproduce the entire city, but to provide the urban constraints that shape the interpretation of candidate actions.

This entity may include transport conditions, environmental exposure, infrastructure status, accessibility, congestion, service demand, and aggregate mobility patterns. These layers define the background conditions under which Human and PoI DTs interact. A candidate PoI may appear suitable at the local level. However, the City or Area DT may indicate high travel burden, poor air quality along the route, congestion, or broader service constraints that reduce its suitability.

In the simulation workflow, the City or Area DT contextualizes each recommendation within the surrounding urban system. It allows the framework to evaluate not only whether a PoI matches user preference and local availability, but also whether selecting it is feasible under current area-wide

conditions. Since city-scale models necessarily simplify complex urban processes, their outputs are interpreted as decision-support evidence rather than deterministic predictions [44, 71].

6.1.5 Entity Boundaries and Roles

The boundaries among the three DT entities preserve the logic of scenario-based decision support. The Human DT represents user state, the PoI DT represents candidate option state, and the City or Area DT represents broader urban constraints. Keeping these roles distinct prevents user preference, service availability, and city-level conditions from being collapsed into a single score before their separate effects can be interpreted.

The entities also differ in update frequency and data granularity. Human DT updates are driven by recent behavior, location, activity, interaction patterns, and user-associated sensing. PoI DT updates are driven by operational changes such as occupancy, availability, congestion, service status, and localized environmental conditions. City or Area DT updates are driven by aggregate environmental, infrastructural, spatial, and mobility indicators that usually stabilize over broader intervals.

Together, these entities define the representational basis for DT scenario evaluation. AI recommendations from Chapter 5 provide candidate actions. The Human, PoI, and City or Area DTs then evaluate those candidates under current contextual conditions. The resulting decision-support output is therefore not only a ranked recommendation list, but a simulation-informed interpretation of which alternatives are feasible, contextually suitable, and aligned with user productivity goals within the constraints of the surrounding urban environment.

6.2 Interaction Between Digital Twins

6.2.1 Operational Interaction for Scenario Evaluation

In the proposed framework, interaction between DTs provides the state exchange required to evaluate AI-generated recommendations before they are enacted in the physical environment. Chapter 5 produces candidate activities, services, or PoI through context-aware prediction and recommendation. The role of the DT layer is to examine those candidates as simulated scenarios by coordinating the Human DT, the PoI DT, and the City or Area DT.

This interaction depends on two linked channels. Physical-to-digital updates capture changes in user activity, PoI conditions, and urban context through sensing, service logs, user interaction data, and environmental observations. These updates refresh only the corresponding DT state. Digital-to-physical feedback then returns simulated or recommended outputs to the corresponding Physical

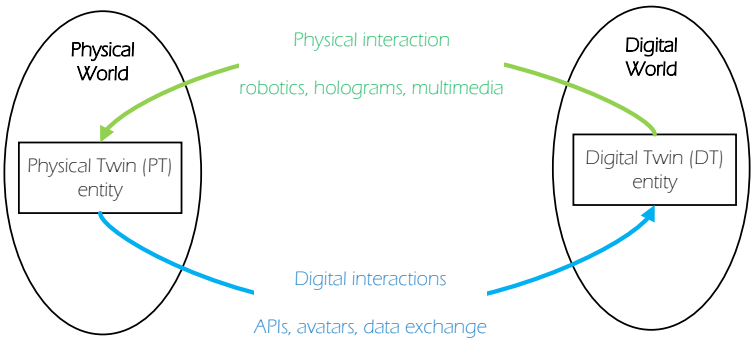

Figure 6.1: Physical and digital interaction channels between a Physical Twin and its corresponding Digital Twin

Twin (PT). Cross-entity interaction occurs within the same domain: PTs interact physically in the physical world, while DTs exchange predictions, constraints, and simulated outcomes digitally in the digital world. In this way, interaction is treated as an operational mechanism for comparing alternatives, rather than as a general DT communication theory.

Figure 6.1 shows the bidirectional communication between a PT entity and its DT counterpart. The physical and digital worlds are connected through update and feedback flows, but their interaction mechanisms differ. Physical interaction occurs through embodied or physical interfaces such as robotics, holograms, and multimedia systems, while digital interaction occurs through computational interfaces such as APIs, avatars, and data exchange. This distinction is important for the proposed framework because physical observations update the DT state, while digital outputs provide feedback that can inform later physical action.

Figure 6.2 applies this communication structure to the Human, PoI, and City or Area entities used in the framework. The Human PT, PoI PT, and City PT interact physically in the physical world, while their corresponding DTs interact digitally in the digital world. The figure also defines the communication boundary between worlds: each PT exchanges updates and feedback only with its own DT counterpart. Thus, cross-entity interaction occurs within a shared domain, either physically among PTs or digitally among DTs, while cross-world communication is limited to counterpart synchronization.

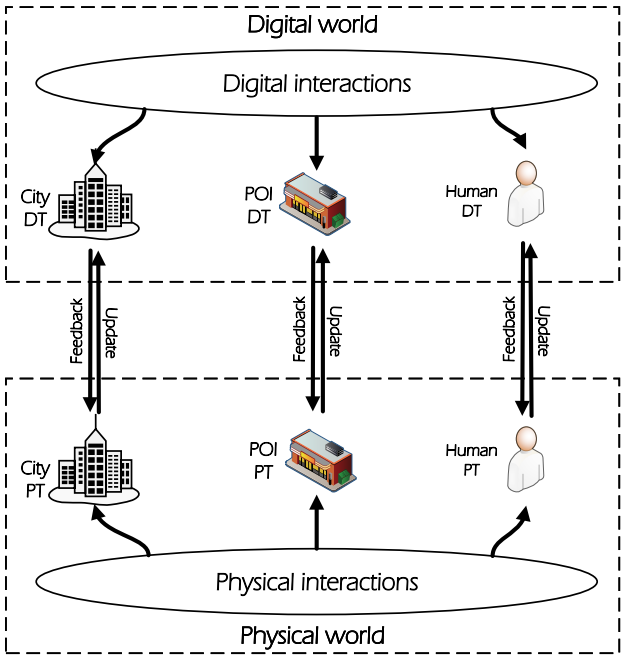

Figure 6.2: Physical-domain and digital-domain interactions among Human, PoI, and City twins with counterpart-only update and feedback links

6.2.2 Interaction Across Human, PoI, and City or Area Twins

Interaction in the proposed framework is organized around three coupled pathways. The first pathway connects the Human DT and the PoI DT. The Human DT contributes user-side information such as preferences, inferred activity, recent behavior, mobility constraints, and productivity-oriented goals. The PoI DT contributes localized information such as availability, occupancy, service status, accessibility, environmental conditions, and capacity. This exchange allows a candidate PoI to be evaluated as a possible action rather than treated only as an item in a ranked list [64, 68, 81].

The second pathway connects the Human DT and the City or Area DT. The City or Area DT supplies broader contextual constraints, including congestion, environmental quality, accessibility, infrastructure status, and area-level service conditions. These conditions shape whether a recommendation that appears suitable for the user is still feasible under the surrounding urban state. At the same time, aggregated and privacy-aware user traces can update city- or area-level representations of demand, exposure, and mobility patterns [10, 23, 44, 110].

The third pathway connects the PoI DT and the City or Area DT. PoI-level states contribute to area-level indicators such as service load, localized congestion, and environmental stress. Conversely, city- or area-level constraints bound PoI-level evaluation by indicating conditions such as access

limitations, infrastructure pressure, environmental alerts, or policy-relevant restrictions [30, 92, 110]. This interaction prevents the evaluation of PoIs from being detached from the broader urban context in which they operate.

Together, these pathways allow the system to evaluate a recommendation through the combined state of the user, the candidate PoI, and the surrounding city or area. A candidate with a high AI recommendation score may receive a less favorable scenario assessment if the PoI is crowded, inaccessible, noisy, or located in an area with poor current conditions. Conversely, an initially lower-ranked option may become preferable if the simulation indicates better feasibility, comfort, and productivity alignment under the current context.

6.2.3 Simulation-Driven Interaction and Shared Context

The interaction among twins is simulation-driven. Rather than waiting for a recommendation to be enacted and then observing its effects, the DT layer evaluates alternative recommendations before physical action occurs. A candidate action can therefore be tested against the current Human DT state, the relevant PoI DT state, and the City or Area DT state. This supports what-if comparison of alternative locations, timings, services, or activity sequences without imposing the cost or disruption of physical experimentation [41, 101].

Shared context is necessary for this process. The Human, PoI, and City or Area DTs rely on the structured contextual representation produced from heterogeneous IoT streams, including environmental, spatial, temporal, activity-related, user-related, and service-related variables. These variables are standardized through preprocessing and temporal alignment so that the twins evaluate candidate actions using a coherent state representation [78, 101].

Through this shared context, the DT layer can compare candidate recommendations according to expected relevance, feasibility, comfort, access burden, service availability, congestion exposure, and productivity alignment. The interaction among twins therefore functions as the state-exchange mechanism that converts AI recommendations from Chapter 5 into scenario-evaluation inputs for the decision-support process developed in the following section.

6.2.4 Feedback and Interaction Constraints

Feedback is retained in this chapter only as a mechanism for updating twin states after simulated or recommended actions are observed. When a recommendation is accepted, rejected, modified, or ignored, the resulting user interaction and environmental observations can update the Human, PoI, and City or Area DTs. These updates allow later simulations to reflect observed behavior

and changing urban conditions without repeating the broader feedback-loop discussion developed in Chapter 3 [31, 45, 101].

The interaction process remains bounded by simulation constraints. Twin states depend on the quality, coverage, latency, and synchronization of IoT-derived context. Incomplete sensing, stale observations, missing PoI updates, uncertain user intent, and simplified city-scale representations can affect the reliability of simulated outcomes. Privacy and governance constraints also limit what user-side or behavioral information can be exchanged among twins. These constraints do not eliminate the value of DT interaction. However, they define how its outputs should be interpreted: as comparative decision-support evidence rather than autonomous or deterministic decisions.

Taken together, interaction among Human, PoI, and City or Area DTs provides the operational bridge from AI recommendation to DT scenario evaluation. The recommender model identifies candidate actions, the twins exchange state information through shared context, and the simulation layer compares alternatives before action.

6.3 Scenario Simulation and Decision Support

6.3.1 Role of Scenario Simulation

In this thesis, scenario simulation is the operational role of the DT layer. The context-aware recommender model developed produces candidate activities, services, or PoIs, but these outputs are not treated as final decisions. They become candidate actions for DT-based scenario evaluation and productivity-oriented decision support. Scenario simulation, therefore, extends the recommendation beyond ranking by examining the expected consequences of possible actions under the current user, PoI, and city or area context.

The purpose of simulation is comparative rather than deterministic. A scenario does not claim to predict a single future state. Instead, it represents a feasible action under specified assumptions and evaluates how that action interacts with the Human DT, the relevant PoI DT, and the City or Area DT. This use of DTs for scenario exploration is consistent with prior work that positions urban DTs as synchronized representations for analysis, simulation, and decision support rather than as static visualizations alone [30, 92]. Within the proposed framework, this capability is narrowed to recommendation refinement: alternatives are compared according to relevance, feasibility, comfort, access burden, service availability, congestion exposure, and productivity alignment.

Scenario simulation is therefore positioned between AI-based recommendation and final advisory output. The resulting decision-support output supports comparison and interpretation of the alternatives using predicted relevance together with scenario-quality criteria.

6.3.2 Scenario Definition and Parameterization

A scenario is defined as a structured representation of a possible user action under a given contextual state. It specifies the user state, the current context vector, the candidate activity or PoI, the DT entities involved, and the criteria used to evaluate the simulated outcome. In this formulation, the scenario is not a generic what-if construct. It is the mechanism through which an AI-generated recommendation is tested against the current Human, PoI, and City or Area DT states.

Scenario parameterization depends on the heterogeneous contextual and IoT-derived data that maintain the DT state, including environmental, spatial, temporal, service-related, and user-related variables [30, 60]. The user state includes preferences, recent behavior, inferred activity, mobility constraints, interaction history, and any relevant contextual or physiological indicators available to the system. The context vector includes the environmental, spatial, temporal, user-related, activity-related, and service-related variables constructed through the IoT and context-modeling layer. The candidate action represents the activity, service, route, timing, or PoI being evaluated. The DT entities provide the state representations required to assess the action from the user, local service, and urban-context perspectives.

Formally, a candidate scenario for user u and candidate action i at time t is represented as

$$\mathcal{Z}_{u,i,t} = \{h_t, c_t, i, D_t^H, D_{i,t}^P, D_t^C, \mathcal{J}\}, \tag{6.1}$$

where h_t denotes the current user state, c_t denotes the structured context vector, i denotes the candidate activity or PoI, D_t^H denotes the Human DT state, $D_{i,t}^P$ denotes the relevant PoI DT state, D_t^C denotes the City or Area DT state, and $\mathcal{J}$ denotes the evaluation criteria used to assess the scenario.

This parameterization preserves the role of each DT entity. The Human DT evaluates whether the candidate fits the user's current activity, preferences, constraints, and productivity-oriented goals. The PoI DT evaluates localized feasibility, including availability, capacity, waiting time, accessibility, environmental conditions, and service status. The City or Area DT evaluates broader constraints such as congestion, travel burden, environmental exposure, infrastructure state, and area-level service pressure.

6.3.3 Multi-Entity Simulation Workflow

The simulation workflow begins with the candidate recommendations generated by the context-aware AI model. At time t, the system receives the current context vector c_t and the dynamic user state h_t. The Human DT is updated using the user's recent context, inferred intent, preferences, and interaction history. In parallel, the relevant PoI DTs are updated using service availability,

capacity, occupancy, accessibility, and localized environmental indicators. The City or Area DT is updated using area-level information such as congestion, environmental state, mobility conditions, and infrastructure constraints.

Multi-entity simulation is the mechanism through which the framework evaluates consequences across user, PoI, and city or area states rather than treating each recommendation as an isolated item score. This coordinated evaluation is necessary because changes in one urban entity can affect accessibility, service demand, user activity, and localized conditions in other entities [30, 68]. For each candidate $i \in R_t$, the DT layer constructs the scenario $\mathcal{Z}_{u,i,t}$ and evaluates the expected outcome of selecting that candidate.

The scenario-evaluation function is expressed as

$$\mathcal{E}_{u,i,t} = \Omega\left(D_t^H, D_{i,t}^P, D_t^C, c_t, i\right). \tag{6.2}$$

where $\mathcal{E}_{u,i,t}$ denotes the DT scenario-evaluation outcome for candidate i at time t, and $\Omega(\cdot)$ denotes the scenario-evaluation function. The outcome summarizes how the candidate is expected to perform under the current user, PoI, and city or area states.

The multidimensional evaluation outcome $\mathcal{E}_{u,i,t}$ is summarized through the scenario criteria into a scalar scenario-quality score $Q_{u,i,t}$. The overall scenario quality for candidate i at time t is defined as

$$Q_{u,i,t} = \omega_1 Rel_{u,i,t} + \omega_2 \mathrm{Feas}_{u,i,t} + \omega_3 \mathrm{Comf}_{u,i,t} + \omega_4 \mathrm{Prod}_{u,i,t} - \omega_5 \mathrm{Bur}_{u,i,t}, \tag{6.3}$$

where $Rel_{u,i,t}$ denotes contextual relevance, $\mathrm{Feas}_{u,i,t}$ feasibility, $\mathrm{Comf}_{u,i,t}$ comfort, $\mathrm{Prod}_{u,i,t}$ productivity alignment, $\mathrm{Bur}_{u,i,t}$ access burden or travel cost and $\omega_1, \ldots, \omega_5$ denote the relative weights assigned to the scenario criteria. A higher value of $Q_{u,i,t}$ indicates greater scenario suitability under the current context.

This workflow makes the DT layer an evaluation mechanism rather than a separate visualization component. The AI model ranks candidates according to predicted relevance. At the same time, the DT layer tests whether those candidates remain suitable when user state, PoI conditions, and city or area constraints are considered together.

Algorithm 2: Digital Twin Scenario Simulation for Recommendation Refinement

Input: User state h_t, context vector c_t, AI recommendation set R_t

Output: Scenario-evaluation outcomes $\mathcal{E}_{u,i,t}$ and scenario-quality scores $Q_{u,i,t}$

Update DT components D_t^H, $D_{i,t}^P$, and D_t^C

for *each candidate* $i \in R_t$ **do**

  Construct scenario $\mathcal{Z}_{u,i,t} = \{h_t, c_t, i, D_t^H, D_{i,t}^P, D_t^C, \mathcal{J}\}$

  Evaluate scenario $\mathcal{E}_{u,i,t} = \Omega(D_t^H, D_{i,t}^P, D_t^C, c_t, i)$

  Compute scenario-quality score $Q_{u,i,t}$

end

Return scenario evaluations for comparative decision support

6.3.4 Illustrative Scenario

A focused work recommendation illustrates the workflow. Suppose a user is seeking a suitable location for an afternoon work session. The AI model may initially rank nearby workspaces, libraries, cafes, or public facilities using the user state and context vector. The DT layer then evaluates these candidates as scenarios rather than accepting the initial ranking directly.

The Human DT represents the user's preference for quiet environments, recent activity patterns, mobility constraints, and inferred intent to complete a productivity-oriented task. The PoI DTs represent the current state of each candidate location, including availability, capacity, distance, noise level, opening status, and service conditions. The City or Area DT represents the surrounding urban context, including congestion, accessibility, environmental quality, and mobility constraints.

Under this simulation, a nearby cafe may receive a lower scenario-quality assessment if its PoI DT indicates high noise or limited seating. A farther library may receive a higher adjusted assessment if it has lower noise, sufficient availability, and better support for focused work, even with a greater travel burden. A delayed activity option may be less suitable if it reduces congestion exposure but conflicts with the user's preferred schedule. The final advisory output therefore reflects not only predicted relevance, but also the simulated consequences of each candidate under current user, PoI, and city or area conditions.

6.3.5 Decision-Support Output and Simulation Constraints

The output of the simulation process is a recommendation list supported by scenario-level evidence. This evidence may include the adjusted assessment, the main factors affecting the score, and a brief explanation of the trade-offs among relevance, feasibility, comfort, access burden, service availability, and productivity alignment. The recommendation is therefore framed as advisory decision support rather than as an automatic instruction.

The main value of this process is that it exposes why a recommendation is suitable under current conditions. A candidate may be highly relevant but infeasible because of limited availability, excessive travel burden, congestion, or poor environmental suitability. Conversely, a candidate with a lower AI prediction score may become preferable after DT evaluation if it better supports the user's intended activity and productivity goals. In this way, simulation extends recommendations beyond ranking by evaluating possible consequences before action.

The interpretation of these outputs remains bounded by simulation constraints. Scenario quality depends on the fidelity of the Human, PoI, and City or Area DTs, the quality and timeliness of IoT-derived context, the completeness of PoI and city-state information, and the assumptions used in the scenario-evaluation function. Because DT outputs depend on synchronization, data quality, model fidelity, and explicit assumptions, they should be interpreted as relative comparisons and sensitivity-oriented decision-support evidence rather than guaranteed forecasts [25, 30, 92].

6.4 Simulation Constraints and Practical Considerations

The Digital Twin layer converts the AI recommendations into scenario-based evaluations. Its outputs support decision analysis by comparing candidate actions through the interacting Human, PoI, and City or Area DTs. However, these outputs should be interpreted as constrained scenario evidence rather than definitive forecasts, because their reliability depends on data quality, synchronization, model fidelity, scenario assumptions, and transparent interpretation [44, 71, 92].

The first constraint is model fidelity. The Human, PoI, and City or Area DTs are task-oriented abstractions that represent only the features required for recommendation refinement and scenario evaluation. The Human DT may capture preferences, recent behavior, activity state, mobility context, and relevant contextual signals, but it cannot fully represent user intent or decision-making. The PoI DT may represent availability, capacity, accessibility, service status, and localized environmental conditions. In contrast, the City or Area DT may represent congestion, infrastructure state, environmental exposure, and broader spatial constraints. These representations are useful only to the extent that they capture the variables relevant to the simulated decision, not because they reproduce the full complexity of the physical city [30, 44, 71, 92].

The second constraint is scenario validity. A simulated recommendation may appear plausible even when the underlying assumptions are weak, the parameters are poorly calibrated, or the relations among user state, PoI conditions, and city or area constraints are unstable. In the proposed workflow, each candidate action is evaluated through the combined state of multiple twins, so errors in one entity can affect the interpretation of the whole scenario. This is especially important when scenario quality is used to adjust the AI recommendation score, since the final decision-support

output depends on both predicted relevance and simulated feasibility, comfort, access burden, service availability, and productivity alignment [44, 55, 71].

The third constraint is synchronization. The framework assumes that twin states are sufficiently current for scenario evaluation, but city-scale deployment cannot guarantee continuous real-time alignment across users, PoIs, and urban subsystems. A user state may be current while PoI occupancy is delayed, or area-level congestion may be updated less frequently than localized service conditions. Such a temporal mismatch can reduce the reliability of the simulated comparison, even when each individual data source is reasonable. DT outputs should therefore be read as evaluations under bounded synchronization rather than as fully current replicas of urban conditions [38, 62, 92, 115].

Trust in the simulation output depends on whether the system can explain how input conditions, twin states, and scenario criteria shaped a recommendation. In this chapter, explainability is not a general governance requirement; it is a practical requirement for interpreting the decision-support output. The user or decision-maker should be able to see why one candidate was preferred over another, including the role of relevance, feasibility, comfort, travel or access burden, service availability, and productivity alignment. This is necessary because the framework combines recommender logic, predictive modeling, and interacting DTs, which can make the basis of the final output difficult to inspect directly [34, 37, 55, 81].

Privacy matters here only insofar as it affects user-state representation. If location, activity, preference, or device-derived signals are unavailable, abstracted, or aggregated, the Human DT may become less precise. This can reduce the reliability of productivity-oriented scenario evaluation, but it does not change the advisory role of the system.

Taken together, these constraints define the practical interpretation of Chapter 6. DT simulation extends AI recommendation into scenario evaluation, but its outputs remain bounded by fidelity, validity, synchronization, and interpretability. They should support human decision analysis under explicit assumptions rather than replace human judgment [30, 44, 71].

6.5 Chapter Summary

Digital Twin modeling serves as the simulation anchor of the proposed IoT–AI–DT framework. Building on the context-aware recommendations produced in Chapter 5, ranked candidate activities, services, and PoI are passed to the DT layer for scenario-based evaluation before being presented as decision-support outputs.

Three operational entities structure this layer: the Human DT, the PoI DT, and the City or Area DT. These entities preserve separate representations of user state, localized PoI, and service

conditions, and broader urban constraints. This entity model allows recommendations to be interpreted through personal context, PoI availability, accessibility, environmental conditions, and area-level constraints rather than through predicted relevance alone.

Interaction among these entities enables candidate recommendations to be evaluated as multi-entity scenarios. User preferences, activity context, service availability, congestion, access burden, comfort, and productivity alignment are considered together. This interaction model supports comparative scenario evaluation while keeping the DT layer within an advisory decision-support role.

Scenario simulation connects AI recommendation outputs to DT-based evaluation. The workflow updates the relevant twin states, constructs candidate scenarios, evaluates simulated outcomes, and combines AI-based recommendation scores with scenario quality. The resulting output is an advisory recommendation that reflects both predicted relevance and simulated suitability under current contextual conditions.

This process remains bounded by simulation constraints. Scenario outputs depend on IoT-derived context quality, sensing coverage, synchronization, model fidelity, PoI update reliability, and the assumptions embedded in the evaluation function. Accordingly, the DT layer supports comparative decision analysis rather than autonomous or guaranteed prediction.

Overall, the chapter establishes the transition from context-aware AI recommendation to DT scenario evaluation and decision-support output. This provides the simulation basis for Chapter 7, where the integrated framework is evaluated through predictive performance, scenario-based reasoning, and productivity-oriented decision support.

Chapter 7

Integrated Evaluation and Productivity

7.1 System Walkthrough

This section provides a brief evaluation-oriented walkthrough of the proposed pipeline. It does not restate the architecture, data flow, model design, or DT structure developed earlier. Instead, it clarifies how the integrated framework is followed in this evaluation as the basis for quantitative model evaluation, ablation analysis, scenario-based reasoning, productivity-oriented interpretation, and evaluation limitations.

Figure 7.1 summarizes the evaluation flow used in this chapter. Physical-world observations are first acquired from heterogeneous sensing and digital sources, including fixed sensors, mobile devices, wearables, and service logs. These inputs provide environmental, spatial, temporal, and user-related signals, but they are not evaluated as raw streams. They are transformed into a structured context through ingestion, temporal alignment, metadata preservation, normalization, filtering, aggregation, and interpolation or imputation where required. This treatment is consistent with smart-city information flows in which IoT supports data acquisition, context-aware computing transforms observations into interpretable states, and higher-level analytics support decision-making [2, 11, 60, 78, 102]. In this chapter, the context representation is therefore treated as the input condition for evaluation rather than as a new architectural contribution.

The structured context is then consumed by the predictive and recommender model. Within the evaluation, the model output is assessed first through quantitative measures that compare predictive performance against the selected baselines and ablation variants. The wide-and-deep model is relevant here because it combines memorization of recurring feature relationships with generalization across less frequent contextual combinations, allowing the evaluation to test whether the inclusion of structured context improves prediction and recommendation performance [28, 112]. The model output is therefore evaluated both as a predictive score and as a recommendation signal that can be passed to the DT layer for scenario-based assessment.

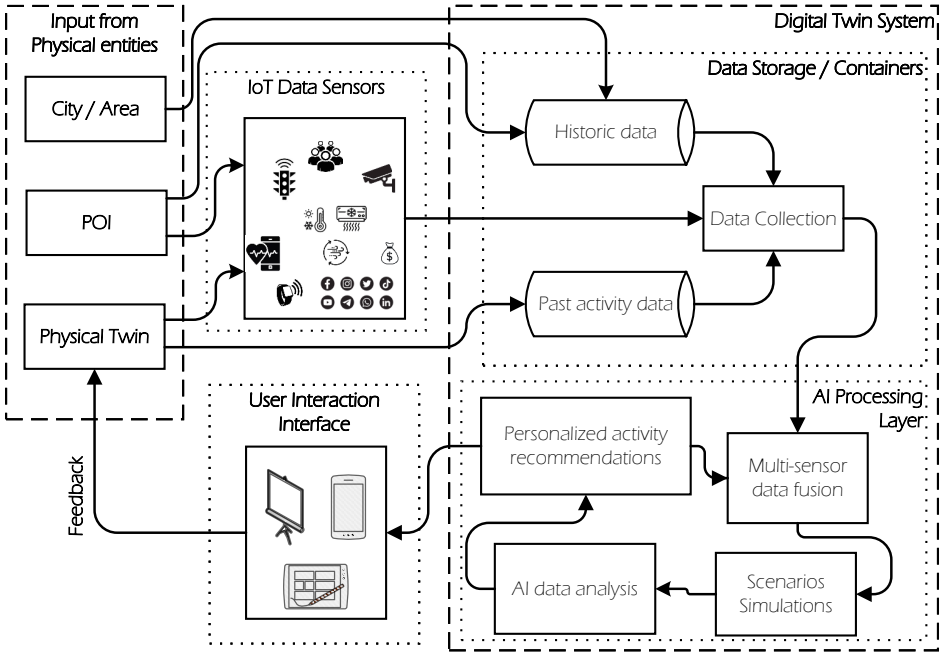

Figure 7.1: Workflow of the proposed Digital Twin system highlighting the end-to-end data cycle

The Digital Twin stage evaluates the recommended alternatives as contextual scenarios rather than as isolated ranked outputs. The Human DT, City or Area DT, and PoI DT provide the representational basis for comparing candidate actions under current user, PoI, environmental, and area-level conditions. This use of DTs is consistent with smart-city DT perspectives that emphasize synchronized digital representations, scenario evaluation, and decision support [30, 68, 92]. In the present evaluation, however, the DT layer is not reintroduced as a technical model. It is used as an evaluation mechanism for examining whether AI-generated recommendations remain feasible, interpretable, and contextually suitable when user state, PoI conditions, and surrounding urban constraints are considered together [44, 71].

The productivity interpretation is also evaluative rather than architectural. Productivity is not treated as a directly measured ground-truth outcome, because complete measurements of useful time, task quality, environmental context, and user constraints are not available. Conceptually, productivity can be expressed as:

$$\Pi = f_{\text{prod}}\left(T_{\text{useful}}, \kappa_{\text{task}}, \chi_{\text{env}}, \chi_{\text{usr}}\right) \tag{7.1}$$

where T_{useful} is time devoted to goal-directed activity, κ_{task} is task quality or success, χ_{env} is

environmental and situational context, and χ_{usr} denotes user preferences and limitations. This framing is consistent with prior work linking productivity to effective time use and task completion, human-computer interaction measures such as task time and success, and environmental effects on performance [19, 33, 36, 91, 107].

For the integrated evaluation, productivity is therefore operationalized indirectly as:

$$\Pi \approx f_{\text{prod}}\left(\chi_{\text{env}}, \rho_{\text{rec}}\right). \tag{7.2}$$

where χ_{env} denotes sensed environmental and situational context, and ρ_{rec} denotes the relevance and contextual suitability of generated recommendations. Equation (7.2) is not an absolute productivity score. It defines the scope of the evaluation: the system is assessed in terms of whether context-aware recommendation and DT-based scenario reasoning can support productivity-oriented decision making by improving alignment among environmental conditions, user situation, and recommended action.

Feedback remains relevant to the interpretation of the evaluation because recommendations, user responses, and environmental changes define how the pipeline would be updated after action. These feedback signals may include explicit user input, implicit behavioral traces, or measurable changes in surrounding conditions. In this evaluation, feedback is not assessed as a full deployment loop. It is used to frame the system as an evolving decision-support process whose reliability depends on context quality, model performance, DT synchronization, and update infrastructure [38, 78]. Accordingly, the remainder of the evaluation examines the proposed framework within its stated scope: predictive performance, ablation evidence, scenario reasoning, productivity-oriented decision support, and the limitations that bound these results.

7.2 Training and Evaluation Methodology

7.2.1 Dataset Preparation and Experimental Splits

The evaluation is based on offline datasets derived from the integrated IoT-driven smart environment developed earlier in the thesis. Raw sensor streams, user interaction logs, and PoI attributes are transformed into structured supervised-learning records using the preprocessing and feature-engineering procedures defined in Section 5.2. Although noise is included as an environmental context variable in the proposed framework, noise measurements were not available in the experimental dataset and were therefore not included in the implemented model evaluation. Each record combines user-related attributes, environmental context, temporal indicators, spatial or PoI descriptors, and explicit preference–context match features. The purpose of this dataset construction is to

evaluate whether the predictive component can estimate context-dependent preference scores that support downstream recommendation and DT reasoning, rather than to repeat the preprocessing or architectural details already presented.

The reported dataset is generated through a structured synthesis process that combines real-world Ottawa climate observations with simulated user preference profiles. The environmental observations are derived from hourly climate data obtained from ClimateData.ca, which aggregates Environment and Climate Change Canada records into standardized downloadable tables suitable for research use. These observations are converted into contextual PoI-event records before being combined with synthetically generated user profiles. Ratings are then produced by matching users with processed PoI-event records and computing learned context-conditioned rating scores over contextual dimensions, including temperature, dew point, humidity, wind speed, visibility, station pressure, wind chill, and time-of-day alignment. The resulting similarity score is mapped to a discrete 1–5 preference rating, with user and item bias terms and small stochastic perturbations included to reduce artificial regularity.

Temporal consistency is retained as a central preparation requirement. Where temporal ordering is available, interactions are partitioned in chronological order rather than through purely random splitting, reducing information leakage from future observations into model training and aligning the evaluation with established practice for recommender systems in dynamic settings [2]. Missing values produced during joining and feature construction are retained through explicit missing-value handling, with categorical missing entries represented using a dedicated `__MISSING__` category. The quantitative evaluation is conducted on predicted preference scores rather than through a separate top-N ranking protocol, preserving alignment with the regression-oriented formulation introduced in Chapter 5 [112].

The training partition is used to estimate model parameters, the validation partition is used for hyperparameter selection and early stopping, and the test partition is held out for final reporting. This separation ensures that the reported test results are not obtained from the same records used for model fitting or validation-based configuration selection.

7.2.2 Training, Validation, Hyperparameter Selection, and Reproducibility

All models are trained offline in a supervised setting using historical user–PoI-event rating records. This design evaluates the predictive component of the proposed framework under controlled experimental conditions rather than claiming real-time deployment performance. Training minimizes the discrepancy between predicted and observed rating values, which is consistent with the role of the model as a regression-oriented predictive core within the broader recommendation framework [66].

Table 7.1: Dataset characteristics used in the integrated evaluation

Element	Description
Data source	Real-world hourly Ottawa climate observations, combined with user preference profiles and processed PoI-event contextual records.
Number of users	500 unique user preference profiles.
Number of base PoIs	131 unique physical PoI or climate-station locations.
PoI-event records	740,582 processed contextual PoI-event records, where each record represents a base PoI under a specific date and time-of-day context.
Number of rating records	1,000,000 user–PoI-event rating records used for training, validation, and testing.
Feature groups	User preference features, PoI identifiers, environmental context, temporal context, spatial-related identifiers, and explicit preference–context match features.
Prediction target	User preference score represented on a 1–5 rating scale.
Recommendation target	Predicted relevance or suitability score used for ranking candidate PoIs or activities.
Training split	700,000 records, corresponding to 70% of the rating dataset.
Validation split	100,000 records, corresponding to 10% of the rating dataset.
Test split	200,000 records, corresponding to 20% of the rating dataset.
Preprocessing	Normalization of continuous variables, embedding of categorical variables, missing-value handling, temporal alignment, and feature-vector construction.

The proposed context-aware wide-and-deep model was trained using GPU acceleration on an NVIDIA GeForce RTX 2070. Hyperparameter tuning was performed over 24 configurations and four random seeds: 42, 256, 1026, and 2026. The validation set was used only for tuning, convergence monitoring, and checkpoint selection. The final selected configuration used 16-dimensional user and item embeddings, 8-dimensional categorical context embeddings, hidden layers of 128, 64, and 32 units, a learning rate of 0.0005, weight decay of 0.00001, batch size of 128, and early stopping with patience of 30 epochs. Although the maximum number of epochs was set to 300, the best validation checkpoint was selected at epoch 72 based on validation RMSE.

Regularization is included to reduce overfitting in the presence of high-dimensional user, PoI, and contextual features. The reported configuration uses L_2 weight decay, while the evaluated search space also considered dropout. Such regularization is standard in neural recommender systems and supports more robust generalization beyond the training records [51]. Temporal validation is important in this setting because user preferences and environmental conditions may change across time; separating training, validation, and test periods reduces optimistic bias and better reflects the need to generalize to future observations [47].

Table 7.2 reports the selected hyperparameters configuration for the proposed context-aware wide-and-deep model. The final model contains 51,266 trainable parameters and reaches its best

Table 7.2: Selected hyperparameters configuration for the proposed context-aware wide-and-deep model after validation tuning

Hyperparameter	**Selected Value**
User/item embedding dimension	16
Categorical context embedding dimension	8
Hidden layers	[128, 64, 32]
Dropout rate	0.00
Learning rate	0.0005
Weight decay	0.00001
Batch size	128
Trainable parameters	51,266
Best validation RMSE	0.3685
Best epoch	72
Selected seed	42

validation performance at epoch 72 using random seed 42. The best validation RMSE is 0.3685, indicating stable predictive performance under the selected experimental configuration.

Reproducibility is supported through fixed random seeds, consistent preprocessing, fixed train-validation-test partitions, validation-based checkpoint selection, and explicit reporting of the selected hyperparameters. All evaluated models use the same data split and metric definitions, so differences in reported performance reflect model behavior under a shared evaluation protocol rather than differences in data preparation or test conditions.

7.2.3 Evaluation Metrics and Quantitative Scope

The quantitative evaluation uses four regression metrics: explained variance (EV), the coefficient of determination (R^2), mean absolute error (MAE), and root mean squared error (RMSE). These metrics are used because the evaluated model estimates continuous preference-related scores that support later recommendation and DT reasoning. EV and R^2 assess goodness of fit and variance capture, while MAE and RMSE assess prediction error, with RMSE assigning a stronger penalty to larger deviations [52, 58].

All reported metric values are computed on the held-out test partition using the final validation-selected model. No post-hoc calibration, test-time adaptation, or retuning is applied after model selection. The resulting quantitative results should therefore be interpreted as offline regression-based evidence about predictive fit and error behavior under the reported dataset and split.

The scope of the quantitative evaluation is deliberately bounded. It evaluates how accurately the predictive component estimates context-dependent preference scores, and it supports comparison between the proposed model and the baseline models reported later. It does not constitute a separate top-N ranking experiment, a live deployment study, or a controlled user study. Accordingly, the quantitative results can support claims about predictive performance under the reported experimental conditions. However, they cannot by themselves prove real-world productivity gains, long-term behavioral change, or urban-scale operational impact.

7.2.4 Productivity-Oriented Evaluation Metrics

Because the thesis evaluates decision support as well as prediction accuracy, productivity is operationalized through decision-support proxies that can be measured in future controlled evaluations. In this section, these metrics provide an interpretive framework for scenario-based analysis rather than direct empirical measurements from a user study. Productivity is therefore treated as the extent to which the system may reduce decision time, improve decision accuracy, or lower user effort when selecting context-appropriate urban activities, services, or PoIs under the stated evaluation scope.

Decision time measures the time required by a user or evaluation procedure to select an activity, service, or PoI. The absolute time saved by the proposed framework is defined as

$$TS = T_{\text{baseline}} - T_{\text{proposed}}, \tag{7.3}$$

where T_{baseline} is the decision time under a baseline condition and T_{proposed} is the decision time under the proposed framework. The relative time-saving rate is defined as

$$TSR = \frac{T_{\text{baseline}} - T_{\text{proposed}}}{T_{\text{baseline}}} \times 100. \tag{7.4}$$

A positive value of TSR indicates that the proposed framework reduces the time needed to reach a decision.

Decision accuracy measures whether the selected recommendation matches the best available option under the evaluated context. It is defined as

$$DA = \frac{N_{\text{correct}}}{N_{\text{total}}}, \tag{7.5}$$

where N_{correct} is the number of decisions that match the ground-truth or best-ranked option, and N_{total} is the total number of evaluated decisions. The decision-accuracy improvement rate is defined

as

$$DAR = \frac{DA_{\text{proposed}} - DA_{\text{baseline}}}{DA_{\text{baseline}}} \times 100. \tag{7.6}$$

User effort is measured through observable interaction costs, such as the number of inspected alternatives, rejected recommendations, search steps, or user-system interactions required before a decision is made. User-effort reduction is defined as

$$UER = \frac{E_{\text{baseline}} - E_{\text{proposed}}}{E_{\text{baseline}}} \times 100, \tag{7.7}$$

where E_{baseline} is the effort required under the baseline condition and E_{proposed} is the effort required under the proposed framework.

These productivity-oriented metrics define measurable decision-support outcomes, but the reported experiments do not directly measure time saved, user effort reduction, task completion improvement, or long-term productivity change through controlled human-subject evaluation. Their role here is to guide the interpretation of scenario-based decision support and to clarify what the evaluation can and cannot prove. Improved predictive accuracy and scenario reasoning may support more efficient decision support, but they should not be interpreted as direct evidence of broad productivity gains beyond the stated experimental scope.

7.3 Quantitative Results

This section evaluates the predictive contribution of the proposed context-aware wide-and-deep model within the Chapter 7 evaluation scope. The comparison uses the same train, validation, and test split across all evaluated models and reports repeated-run results using RMSE, MAE, R^2, and explained variance. The baseline and ablation configurations are defined in Chapter 5; therefore, this section focuses on measured performance, component-level effects, and statistical reliability rather than on restating model definitions. Recommendation is operationalized in this evaluation as regression-based rating prediction, so the quantitative results assess how effectively each model predicts held-out preference scores under the available contextual and interaction data.

7.3.1 Predictive Performance Across Baselines and Ablations

Table 7.3 shows that the proposed context-aware wide-and-deep model achieved the strongest predictive performance among the evaluated models, with an RMSE of 0.3695, MAE of 0.2647, R^2 of 0.8185, and explained variance of 0.8185. The low standard deviations across repeated runs indicate stable offline performance under the reported evaluation setting. The closest model was the

Table 7.3: Predictive performance of the proposed model and baseline recommender models across repeated runs

Model	**RMSE**μ	**RMSE**σ	**MAE**μ	**MAE**σ	$R^2\mu$	$R^2\sigma$	**EV**μ	**EV**σ
Proposed Context-Aware W&D	0.3695	0.0017	0.2647	0.0021	0.8185	0.0012	0.8185	0.0012
No-DT-features ablation	0.3795	0.0008	0.2777	0.0017	0.8085	0.0009	0.8086	0.0009
Context-Only ablation	0.4079	0.0006	0.3153	0.0042	0.7788	0.0010	0.7792	0.0010
Non-Context ablation	0.5839	0.0006	0.4669	0.0007	0.5467	0.0008	0.5468	0.0009
Matrix Factorization	0.5863	0.0007	0.4674	0.0005	0.5430	0.0014	0.5430	0.0014
Collaborative Filtering	0.5932	0.0006	0.4730	0.0005	0.5322	0.0007	0.5322	0.0007
Neural CF	0.5944	0.0015	0.4751	0.0023	0.5303	0.0022	0.5315	0.0012

No-DT-features ablation variant, which reached an RMSE of 0.3795 and R^2 of 0.8085. This result indicates that the full model provides a measurable predictive gain over the context-aware variant without DT-related features or reasoning.

The larger differences appear when contextual representation or user-item personalization is reduced. The context-only ablation increased RMSE to 0.4079 and reduced R^2 to 0.7788, showing that contextual variables alone do not provide the same predictive strength as the combined context-aware and personalized representation. The non-context ablation produced an RMSE of 0.5839 and R^2 of 0.5467, which is close to the interaction-based collaborative filtering, matrix factorization, and neural collaborative filtering baselines. This pattern indicates that the main predictive contribution comes from integrating contextual, user-state, activity, and PoI-level information within the wide-and-deep learning structure rather than from neural interaction modeling alone.

7.3.2 Ablation Effects

The ablation configurations are defined in Section 5.5. In this section, they are used only to interpret the measured effect of restricting major components of the proposed model: DT-related features or reasoning, user-item personalization, and contextual feature representation.

Table 7.4 shows that removing contextual feature representation caused the largest performance loss, increasing RMSE by 0.2145, increasing MAE by 0.2022, and reducing R^2 by 0.2718. This result provides the clearest evidence that context-aware representation is central to the predictive performance of the proposed model. Limiting user-item personalization in the context-only model also reduced performance, with RMSE increasing by 0.0384 and R^2 decreasing by 0.0397. This indicates that the model benefits from the combination of contextual inputs and personalized interaction structure rather than from context alone.

Table 7.4: Measured ablation effects relative to the proposed context-aware wide-and-deep model

Comparison Model	**Removed or Limited Component**	**ΔRMSE**	**ΔMAE**	ΔR^2
No-DT-features ablation	DT-related features	0.0101	0.0130	-0.0100
Context-Only ablation	User-item personalization features	0.0384	0.0506	-0.0397
Non-Context ablation	Contextual feature representation	0.2145	0.2022	-0.2718

The No-DT-features ablation produced the smallest degradation, with RMSE increasing by 0.0101, MAE increasing by 0.0130, and R^2 decreasing by 0.0100. This result indicates that DT-related features or reasoning provide an incremental predictive benefit in the regression task. Their broader contribution is therefore not limited to error reduction; it is evaluated more directly through scenario-based reasoning and productivity-oriented decision support in the following sections of Chapter 7.

7.3.3 Statistical Validation

Table 7.5: Paired statistical comparison between the proposed model and baseline models

Comparison Model	**RMSE Gain**	**RMSE p-value**	**MAE Gain**	**MAE p-value**	R^2 **Gain**	R^2 **p-value**
Collaborative Filtering	37.71%	< 0.0001	44.03%	< 0.0001	0.2863	< 0.0001
Context-Only ablation	9.42%	< 0.0001	16.03%	0.0004	0.0397	< 0.0001
Matrix Factorization	36.99%	< 0.0001	43.36%	< 0.0001	0.2755	< 0.0001
Neural CF	37.84%	< 0.0001	44.28%	< 0.0001	0.2882	< 0.0001
No-DT-features ablation	2.65%	0.0006	4.67%	0.0040	0.0100	0.0007
Non-Context ablation	36.73%	< 0.0001	43.31%	< 0.0001	0.2718	< 0.0001

Table 7.5 reports paired comparisons between the proposed model and each baseline across repeated runs. The proposed model achieved statistically significant improvements over all comparison models for RMSE, MAE, and R^2. The gains over collaborative filtering, matrix factorization, neural collaborative filtering, and the non-context ablation are large, with RMSE gains ranging from 36.73% to 37.84% and MAE gains ranging from 43.31% to 44.28%. These results show that interaction-only and non-context representations do not capture the contextual variation required for the rating-prediction task under the reported evaluation setting.

The gain over the context-only ablation is smaller but still statistically significant, with a 9.42% RMSE gain, 16.03% MAE gain, and 0.0397 increase in R^2. This supports the interpretation that

context-aware prediction is strongest when contextual features are combined with user-item personalization. The gain over the No-DT-features ablation is the smallest, with a 2.65% RMSE gain, 4.67% MAE gain, and 0.0100 increase in R^2, but the improvement remains statistically significant. This indicates that the DT related component adds a reliable but limited predictive contribution, while the larger improvement comes from context-aware wide-and-deep learning.

Overall, the quantitative results show that the proposed model improves prediction primarily by integrating contextual and personalized representations within a wide-and-deep architecture. The ablation results identify contextual feature representation as the largest contributor to error reduction, while user-item personalization provides an additional improvement over context-only prediction. DT-related features or reasoning provide a smaller offline predictive gain, but their main evaluation role is tied to the scenario-based and productivity-oriented analyses that follow.

7.4 Scenario-Based Evaluation

7.4.1 Evaluative Framing

Scenario-based evaluation is used here to assess whether the framework can distinguish among plausible alternatives under changing urban conditions. The purpose is evaluative rather than descriptive. Rather than restating the simulation process or the DT architecture, this section examines whether the framework produces context-sensitive, coherent, and interpretable distinctions when applied to representative decision situations [44, 71].

The scenarios are illustrative rather than exhaustive. They do not establish statistical generalization, behavioral causality, or population-scale impact. Instead, they provide a qualitative basis for evaluating whether sensed context, recommendation outputs, and DT-based comparison add interpretive decision-support value beyond quantitative model accuracy [44].

7.4.2 Scenario Categories

The scenarios were selected using the contextual representations applied in the framework so that users, candidate PoIs, environmental conditions, and time windows could be compared consistently [78, 104]. This consistency is necessary because the evaluation concerns whether observed differences in suitability arise from contextual variation and user intent rather than from changes in representation.

Three scenario categories were examined. The first concerns work and study activities, including the selection of locations for focused work, reading, or studying. These scenarios make contextual

sensitivity visible because suitability depends on conditions such as noise, air quality, temperature, and crowd density.

The second concerns physical and leisure activities, including walking, exercising, and outdoor recreation. These scenarios test whether the framework responds plausibly when environmental exposure affects the suitability of alternatives, particularly through temperature, ultraviolet conditions, and air quality [43].

The third concerns routine urban activities, including commuting, short errands, and informal social interaction. These scenarios evaluate whether contextual information helps differentiate among options that may appear similar in conventional recommendation ranking but differ in situational suitability [68].

7.4.3 Interpretation Criteria

The scenarios are interpreted through three qualitative criteria. The first is contextual suitability, which refers to the extent to which a candidate option aligns with the environmental and temporal conditions relevant to the stated user intent. A study-related option, for example, is more suitable when the contextual state supports concentration and task continuity.

The second criterion is consistency. Recommendation outputs and comparative assessments should change coherently as contextual variables shift. This criterion is important because abrupt or opaque changes across similar cases would weaken the evidence that the framework provides dependable decision support.

The third criterion is explanatory value. Scenario-based evaluation allows the analysis to inspect why one option becomes more suitable than another under specific conditions, even where the system does not generate explicit natural-language explanations. This criterion responds to wider concerns about transparency and interpretability in smart-city decision-support systems [71]. Together, these criteria position the scenarios as evidence of plausibility, comparative sensitivity, and interpretability rather than as proof of downstream causal effects [44].

7.4.4 Findings and Limits

Across the examined categories, the scenarios indicate that the framework can differentiate between candidate activities and PoIs in ways that reflect both user intent and changing urban context. The main evaluative finding is not that the system identifies a universally optimal option, but that it produces interpretable comparisons whose suitability changes meaningfully with environmental and temporal conditions [44, 71].

The scenarios also suggest that DT-enabled comparison adds value beyond recommendation ranking alone by making trade-offs among alternatives more visible before action is taken. This value is interpretive: the comparison helps examine why one option becomes preferable under a specific contextual configuration, rather than merely presenting a ranked output [14, 35].

These findings remain bounded by the exploratory scope of the evaluation. The scenarios do not demonstrate causal effects on behavior, productivity, or well-being, and they do not prove real-world impact beyond the examined cases. Their credibility depends on the quality of contextual data, the fidelity of DT state updates, and the appropriateness of the scenario assumptions [44, 71]. Additional observations, repeated scenario interpretation, and updated contextual states could strengthen later evaluation cycles [78, 101]. Ethical and practical constraints also limit the granularity and breadth of scenario analysis, particularly where behavioral modeling raises issues of privacy, consent, and governance [16, 39].

The scenario-based evaluation therefore provides a transition from model-centered evaluation to productivity-oriented interpretation. It shows how context-aware recommendation and DT-based comparison can surface trade-offs relevant to activity suitability, time use, and decision quality, while preserving the limits of the present evidence. Section 7.5 builds on this framing by discussing productivity implications without treating the scenarios as measured productivity gains.

7.5 Productivity-Oriented Analysis

This section interprets the Chapter 7 evaluation results in productivity-oriented terms. Productivity is not treated as a directly measured global outcome. Instead, it is evaluated as the decision-support relevance of the proposed framework: whether context-aware prediction, scenario-based reasoning, and DT-evaluated recommendations help align suggested actions with user goals, activity conditions, comfort requirements, and time-use constraints. The analysis therefore remains anchored in the quantitative model results, the ablation findings, and the scenario-based evaluation reported in the preceding sections.

Within this scope, productivity refers to the extent to which the system can reduce decision friction by presenting more suitable alternatives under current environmental, spatial, temporal, activity-related, and user-related conditions. This interpretation is consistent with the productivity-oriented metrics introduced in Section 7.1 and the scenario evaluation in Section 7.4. However, it does not claim that the system directly measures actual time savings, labor output, or city-wide productivity improvement.

Table 7.6: Residual-error analysis for evaluating decision-support reliability

Model	Residual Mean	Residual Mean Std.	Residual Std.	Median AE	P90 AE	Max AE
Proposed Context-Aware W&D	-0.0007	0.0045	0.3695	0.1625	0.6757	1.8567
No-DT-features ablation	-0.0008	0.0089	0.3794	0.1801	0.7006	1.8649
Context-Only ablation	0.0119	0.0123	0.4076	0.2427	0.7134	1.8963
Non-Context ablation	-0.0053	0.0074	0.5839	0.3966	0.9567	2.8934
Matrix Factorization	0.0014	0.0055	0.5863	0.3945	0.9609	2.8902
Collaborative Filtering	-0.0005	0.0007	0.5932	0.3994	0.9741	2.8294
Neural CF	0.0194	0.0259	0.5937	0.4026	0.9755	2.8171

Table 7.6 extends the predictive evaluation by examining residual-error behavior. The proposed context-aware wide-and-deep model produces the lowest residual standard deviation, median absolute error, 90th-percentile absolute error, and maximum absolute error among the evaluated models. This indicates that the proposed model improves both typical-error and tail-error behavior relative to the interaction-based baselines and ablation variants. For productivity-oriented decision support, this matters because lower and more stable residual errors make recommendations more dependable when they are used to compare candidate activities, services, or PoI. However, the residual analysis should be interpreted as evidence of improved predictive reliability, not as direct evidence that users saved time or became more productive.

7.5.1 Individual and Everyday Decision Support

At the individual level, the productivity relevance of the framework lies in its ability to support everyday decision-making under changing urban conditions. The scenario results show that recommendations are not evaluated only as ranked outputs, but as candidate actions that must remain suitable for the user's activity, context, and constraints. This shifts the analysis from generic productivity improvement to decision-support utility: whether the system can help the user identify options that better fit the current situation.

Context-aware recommendation supports this role by incorporating temporal, spatial, environmental, and user-related signals into the prediction process. Prior work on context-aware recommendation shows that treating context as part of the recommendation function can improve relevance compared with static or context-agnostic recommendation approaches [1, 2]. In the proposed framework, this relevance is evaluated through the model results and scenario interpretations rather than assumed as a general productivity gain.

For everyday activity planning, the relevant productivity mechanism is the reduction of the mismatch between user intent and suggested action. A recommendation that aligns with activity type, travel burden, environmental comfort, service availability, and timing can support more efficient planning than a recommendation based only on historical preference or user-item similarity. IoT-derived context contributes to this alignment by supplying environmental and situational information. At the same time, prior work on IoT and context-aware systems supports the importance of heterogeneous sensor data for inferring user needs and situational conditions [60, 78].

The scenario-based evaluation further clarifies how comfort and task alignment enter the productivity interpretation. For a focused-work scenario, a candidate location is not simply better because it has a higher predicted preference score. It becomes more suitable when the Human DT, PoI DT, and City or Area DT indicate that the option is feasible, available, comfortable, and aligned with the user's intended activity. This interpretation is consistent with DT research that frames simulation as a way to compare alternatives before action, while still recognizing that such comparisons remain bounded by model assumptions and data quality [30, 92].

The analysis therefore supports a limited claim: the proposed framework can improve the relevance and interpretability of recommendations for individual activity planning within the evaluated setting. It does not establish that productivity improved in behavioral, economic, or organizational terms. Such claims would require user studies, longitudinal deployment, or direct measurement of task completion time, cognitive effort, or behavioral outcomes.

7.5.2 Urban-Level Interpretation and Constraints

At the urban level, productivity should be interpreted cautiously. The evaluation does not provide direct evidence of city-wide productivity improvement, congestion reduction, or service-efficiency gains. What it supports is a narrower interpretation: context-aware and DT-evaluated recommendations may provide decision-support signals that are relevant to urban resource use when aggregated or applied across services. For example, recommendations that account for congestion, accessibility, environmental exposure, and service availability can help compare alternatives under current urban constraints. Related smart mobility and PoI recommendation studies show that contextual and behavioral information can support travel planning and activity selection. However, those findings should not be generalized here as measured city-wide effects [13, 32].

The same caution applies to health, lifestyle, and activity recommendations. Context-sensitive systems may support better activity scheduling when environmental and user-state information is available, and prior work has examined such mechanisms in health and smart-city recommender settings [15, 20]. In this thesis, however, those implications remain interpretive. The evaluation shows that the proposed framework can represent and reason over activity suitability, comfort,

and contextual relevance, but it does not measure downstream health, lifestyle, or productivity outcomes.

Digital Twin simulation is most defensible here as a comparative reasoning mechanism. Urban DT research emphasizes that simulation is valuable for comparing scenarios and examining possible interventions. However, critical studies also warn against treating DTs as complete or fully predictive representations of urban systems [44, 71]. Accordingly, the urban-level contribution of the proposed framework is not an unsupported claim of improved city productivity. It is the ability to connect individual recommendations with broader contextual constraints such as congestion, accessibility, service state, environmental quality, and localized demand.

Several constraints limit this interpretation. First, productivity-oriented recommendations depend on the quality and availability of IoT-derived context. Missing data, noisy sensors, uneven coverage, and synchronization problems can weaken the context vector and reduce recommendation reliability. Prior work on scalable smart-city systems emphasizes the importance of preprocessing, robust data handling, and fault-tolerant architectures under such conditions [38, 115]. In the proposed framework, these issues directly affect whether a recommendation can be trusted as decision-support evidence.

Second, latency constrains productivity relevance. A recommendation that arrives after the decision window has passed may be accurate but operationally unhelpful. Edge-enabled smart-city systems can reduce delay, but stronger models may increase computational cost and inference time [62]. The productivity value of the framework therefore depends on balancing model sophistication with the timing requirements of the activity being supported.

Third, model generalization remains limited. Urban behavior, environmental conditions, service availability, and user preferences change over time. Recommender systems that rely on historical data may lose relevance unless they are updated and recalibrated. Research on context-aware and intent-aware recommendation similarly stresses the need for ongoing adaptation in dynamic environments [81]. For this thesis, that means the reported evaluation should be interpreted within its experimental scope, not as evidence of stable performance across all cities, users, or deployment conditions.

Finally, governance, fairness, and privacy impose practical limits on productivity-oriented personalization. Recommendations can influence access to mobility, services, and information, so efficiency-oriented optimization may create unequal outcomes if data or models are biased [29]. Privacy also creates a trade-off: fine-grained personalization depends on detailed user and context data, while privacy-preserving methods may reduce granularity, increase overhead, or limit model accuracy [81]. These constraints are especially important for Human DTs, where user state, mobility, activity, and preference signals may be sensitive. DT outputs should therefore inform

decision-making rather than replace empirical validation or human oversight [44, 71].

Overall, the productivity-oriented analysis shows that the proposed IoT–AI–DT framework supports decision-making by improving predictive reliability, contextual suitability, scenario comparison, and alignment between user goals and available urban options. Its value lies in decision-support relevance within the evaluated scope: helping assess which activities, services, or PoI are more suitable under current conditions. The results do not justify broad claims of productivity improvement at the individual or urban scale. They support a more bounded conclusion: context-aware recommendation and DT-based scenario reasoning can provide a structured basis for productivity-oriented decision support, subject to data fidelity, latency, adaptability, privacy, fairness, and simulation limits.

7.6 Integrated Discussion and Limitations

The evaluation in this chapter supports the proposed framework as a bounded context-aware decision-support pipeline. The quantitative results show that the context-aware wide-and-deep model achieves strong predictive performance under the reported offline validation setting, with stable error behavior relative to the evaluated baselines and ablations. The ablation analysis indicates that contextual representation contributes to predictive improvement, while the scenario-based evaluation shows how DT reasoning can support interpretation of recommendation outcomes under changing user, PoI, and area-level conditions. These findings support the value of evaluating the framework through model performance, scenario reasoning, and productivity-oriented decision support rather than through predictive accuracy alone.

The productivity-oriented interpretation should remain limited to the scope of the evaluation. The analysis in Section 7.5 suggests that the framework can reduce decision friction by narrowing broad sets of urban options into smaller sets of contextually plausible alternatives. This is relevant for activity planning, service selection, and route-related choices, where users may otherwise need to compare many options under changing environmental and service conditions. However, the evaluation does not provide causal, longitudinal, or city-scale evidence of sustained productivity gains. The findings therefore support productivity-oriented decision assistance within the evaluated setting, but they do not establish general productivity improvement across populations, cities, or long deployment periods.

The scenario-based evaluation clarifies both the contribution and the boundary of the DT layer. As shown in Section 7.4, Human, PoI, and City or Area DT states provide a structured basis for examining how recommendation outcomes may vary under different environmental conditions, user states, service constraints, and access burdens. This makes the DT layer useful for comparing

alternatives and making trade-offs more explicit. At the same time, the scenario outputs depend on entity fidelity, contextual data quality, synchronization assumptions, and the criteria used to define scenario quality. The evaluation therefore supports DT reasoning as comparative decision-support evidence, not as a guarantee of real-world outcomes.

7.6.1 Evaluation Scope and Generalization

The reported findings are shaped by the data, environment, and validation design used in this chapter. Training and testing were conducted within a specific urban setting with particular spatial layouts, sensing assumptions, user-PoI interaction patterns, and contextual feature distributions. The model should therefore not be assumed to transfer directly to other cities without recalibration, retraining, or domain adaptation. Urban environments differ in mobility infrastructure, land use, service distribution, climate, density, and sensing coverage, and these differences may alter the relationship between user behavior, environmental context, and PoI relevance. Under such distribution shifts, prediction accuracy may decline, especially in context-aware recommender systems whose signals are closely tied to local urban dynamics [2, 80, 104].

The evaluation data also define a clear boundary. The reported results rely on synthetically generated ratings derived from real environmental data and designed preference models. This supports controlled experimentation, repeated comparison, and ablation analysis, but it does not provide direct evidence of user satisfaction, behavioral adoption, trust, or real-world effectiveness. The absence of benchmarking against public recommender datasets also limits direct cross-study comparison. However, most available public datasets do not contain the environmental and urban-context variables required for this framework.

The offline validation procedure provides reproducible evidence of predictive fit and error magnitude through Explained Variance, R^2, Mean Absolute Error, and Root Mean Squared Error. These metrics are appropriate for evaluating technical model performance, but they do not directly measure decision quality, user burden, behavioral adaptation, or productivity outcomes. Offline evaluation also assumes that historical or simulated interaction patterns provide a useful proxy for future utility. In dynamic urban settings, this assumption may be affected by congestion, weather, service disruptions, preference drift, and changing activity constraints. Strong offline performance should therefore be interpreted as evidence of technical feasibility within the reported conditions, rather than as a complete account of real-world decision-support performance [47].

Data availability and quality further constrain interpretation. The framework depends on contextual inputs from IoT infrastructure, yet sensor streams may include missing values, heterogeneous sampling rates, measurement noise, delayed updates, and uneven spatial coverage. Preprocessing and quality indicators reduce some of these effects, but residual inconsistencies may still influence

learned representations and prediction variance. Feature-based representations help address parts of the cold-start problem, but interaction history remains important. Predictions for new users, rarely visited PoIs, or underrepresented areas may therefore be less reliable, consistent with established limitations in recommender-system research [70, 98].

7.6.2 Scenario, Productivity, and Deployment Limits

The scenario-based component evaluates recommendations under selected contextual conditions, but its validity is bounded by scenario design. The scenarios represent structured decision contexts rather than exhaustive simulations of urban behavior. They make trade-offs among relevance, feasibility, comfort, access burden, service availability, and productivity alignment visible, but they do not validate the full closed-loop interaction between recommendations, user behavior, DT updates, and later model outputs. The evaluation therefore does not yet show how recommendations may change behavior over time or how those behavioral changes may reshape future system states [30, 52, 92].

The productivity analysis has the same boundary. Productivity is interpreted through decision-support functions such as reducing search space, improving contextual fit, supporting activity planning, and making trade-offs more explicit. These indicators are useful for evaluating whether the framework is aligned with productivity-oriented decision support, but they are not equivalent to direct measurement of long-term productivity. The chapter does not include longitudinal user studies, controlled field experiments, or direct measures of task completion time, cognitive load, sustained behavioral change, or economic output. Consequently, the evaluation supports productivity-oriented interpretation within the modeled and scenario-based setting, but it does not establish general productivity gains at the city scale.

Deployment and governance constraints affect the evaluation only insofar as they shape the validity of the reported results. The experiments assume a centralized training and evaluation pipeline and do not test operational factors such as latency, fault tolerance, distributed IoT processing, streaming adaptation, or edge inference. Practical deployment may require distributed training, incremental updating, or edge-based computation to reduce communication overhead and response time [38, 115]. Similarly, the evaluation reports aggregate predictive performance rather than subgroup-level outcomes across demographic groups, spatial regions, activity levels, or sensing densities. It therefore does not establish whether the framework performs equitably across users, neighborhoods, or PoI categories, a concern that remains important for urban recommender systems [17, 29, 34, 53].

Overall, the evaluation supports the proposed framework within a bounded scope. It shows that context-aware wide-and-deep prediction can outperform the selected baselines under the reported

offline setting, that ablations clarify the contribution of contextual representation, and that scenario-based DT reasoning can support productivity-oriented interpretation of recommendations. It does not prove unrestricted real-world deployment performance, cross-city generalization, long-term productivity gains, or closed-loop operational robustness. Full validation would require field studies, cross-environment testing, user-in-the-loop evaluation, subgroup analysis, and deployment-oriented assessment under streaming and governance constraints.

7.7 Chapter Summary

Chapter 7 evaluated the proposed framework within a bounded offline and scenario-based evaluation scope. The quantitative results showed that the proposed context-aware wide-and-deep model achieved the strongest predictive performance among the evaluated models, with an RMSE of 0.3695, MAE of 0.2647, R^2 of 0.8185, and explained variance of 0.8185. The ablation analysis further indicated that contextual feature representation was the main contributor to performance improvement, while user-item personalization and DT related components provided additional support.

The scenario-based evaluation showed that DT reasoning can extend predictive recommendations into comparative decision support. Within the stated scope, the framework examined candidate activities, services, or PoI against synchronized user, PoI, and city or area conditions before action is taken. This supports the interpretation of contextual trade-offs, but does not establish autonomous decision-making or real-world deployment performance.

The productivity-oriented analysis interpreted these results as decision-support evidence rather than as direct proof of productivity gains. The framework may support productivity-oriented decision-making by reducing the search space, improving contextual fit, supporting activity planning, and clarifying trade-offs among relevance, feasibility, comfort, access burden, and service availability.

Overall, Chapter 7 establishes the framework as a context-aware predictive and scenario-reasoning pipeline for productivity-oriented decision support under constrained experimental conditions. Its findings remain limited by the offline evaluation design, data quality, sensing coverage, generalization, deployment realism, governance constraints, and the absence of field-based or longitudinal user validation.

Chapter 8

Conclusion and Future Directions

8.1 Summary of Contributions

This thesis contributed an integrated framework for human-centered urban intelligence by connecting IoT-based context acquisition, context-aware AI recommendation, DT scenario reasoning, and productivity-oriented evaluation within a single decision-support pipeline. The contribution directly responds to the gaps identified in Chapter 2: fragmented IoT, AI, and DT systems; shallow or static context representation; recommendation without scenario-based reasoning; and limited evaluation beyond predictive accuracy. It also corresponds to the contributions stated in Chapter 1, namely the unified IoT–AI–DT architecture, the context-aware wide-and-deep model, and DT-based decision simulation.

The first contribution is the unified IoT–AI–DT architecture developed in this thesis. Rather than treating sensing, recommendation, and simulation as independent modules, the framework specifies an end-to-end structure in which heterogeneous IoT observations are transformed into structured context, processed by AI models, propagated into DT entities, and returned through feedback-enabled decision support. This contribution addresses the architectural fragmentation identified in Chapter 2 by defining how physical sensing, digital processing, predictive intelligence, DT synchronization, and user-facing feedback operate as connected parts of one pipeline.

The second contribution is the treatment of context as a core operational input to prediction and recommendation. The thesis defined a structured context representation that incorporates environmental, spatial, temporal, activity-related, user-related, and PoI-level information, together with preprocessing and quality-handling mechanisms for heterogeneous urban data. This contribution addresses the gap of shallow and static context representation by specifying how IoT-derived observations are cleaned, aligned, encoded, and prepared for downstream learning and DT updating, rather than being used only as auxiliary or post-processing variables.

The third contribution is the formulation and evaluation of a context-aware wide-and-deep model for urban recommendation. The model integrates user, activity, PoI, environmental, spatial, and temporal features within a unified learning architecture, allowing context to influence prediction directly. The evaluation in Chapter 7 supports this contribution by showing that, under the reported experimental setting, the proposed context-aware wide-and-deep model achieved stronger predictive performance than the collaborative filtering, matrix factorization, and neural collaborative filtering baselines. This finding supports the thesis claim that IoT-derived context can improve recommendation performance when it is embedded in the learning architecture rather than treated as an external filter.

The fourth contribution is the DT-based scenario reasoning layer. The thesis defined interacting Human, PoI, and City or Area DTs and used them to evaluate candidate recommendations as scenarios before action. This contribution addresses the gap between recommendation and decision support by extending the output of the AI model beyond ranking alone. Candidate activities, services, or PoI are assessed through user state, PoI conditions, and broader urban constraints, allowing recommendations to be interpreted in terms of feasibility, comfort, access burden, service availability, and productivity alignment.

A further evaluative contribution is the productivity-oriented assessment of the integrated framework. Chapter 7 combined model-level metrics with scenario-based interpretation and productivity-oriented analysis, thereby assessing the framework not only as a predictive model but also as a decision-support system. This contribution remains bounded by the evaluation scope of the thesis: the productivity analysis is exploratory and system-level, not a definitive measurement of real-world productivity gains. Its value lies in showing how technical outputs can be related to user decision-making, activity planning, service use, and urban-resource alignment rather than being evaluated only through algorithmic accuracy.

Taken together, these architectural, modeling, simulation, and evaluative contributions close the thesis argument by showing that personalization in smart urban environments can be framed as an integrated, context-aware, and DT-supported decision-support problem. The thesis does not claim to provide a fully deployed smart-city platform, a complete governance model, or universally transferable results. Its contribution is the design, implementation, and evaluation of a coherent research framework that connects the gaps identified in Chapter 2, the contributions stated in Chapter 1, and the evidence reported in Chapter 7, while preserving the need for further work on deployment, data quality, privacy, scalability, fidelity, and transferability [40, 81].

8.2 Key Findings

This thesis finds that context-aware personalization in smart urban environments is most defensible when treated as an integrated decision-support problem rather than as an isolated recommendation task. The evidence across the preceding chapters supports four main findings: the architectural integration of IoT, AI, and DT components; improved offline predictive performance from context-aware modeling; the value of DTs for scenario-based decision support; and the usefulness of productivity-oriented evaluation as a complement to accuracy-based assessment.

8.2.1 Integrated Architecture for Urban Personalization

The first finding is that the proposed IoT-AI-DT architecture provides a coherent response to the fragmentation identified in Chapter 2. By linking IoT-based context acquisition, structured context modeling, AI-driven recommendation, and DT representation within a single pipeline, the framework shows how sensing, prediction, simulation, and feedback can be organized as connected system functions rather than separate technical layers.

The level of evidence for this finding is architectural and implementation-oriented. The thesis demonstrates that heterogeneous sensor, environmental, user-state, activity, spatial, temporal, and PoI data can be transformed into structured contextual representations and passed through downstream recommendation and DT components. This supports the Chapter 1 contribution of an end-to-end integration of IoT, AI, and DTs, while remaining bounded by the system assumptions and deployment constraints discussed earlier in the thesis.

8.2.2 Context-Aware Recommendation Performance

The second finding is that contextual modeling improved recommendation performance in the reported offline predictive evaluation. The proposed context-aware wide-and-deep model achieved the lowest RMSE (0.3695), the lowest MAE (0.2647), and the highest R^2 and explained variance (0.8185) among the evaluated models. These results indicate that the model captured variation in the recommendation task more effectively than the collaborative filtering, matrix factorization, and neural collaborative filtering baselines considered in the evaluation.

The evidence for this finding is quantitative but limited to the reported offline setting. The results support the claim that integrating spatial, temporal, environmental, user-state, activity, and PoI-level features can strengthen predictive recommendations under the evaluation conditions used in the thesis. They do not, by themselves, establish real-world behavioral impact or long-term deployment effectiveness. This interpretation is consistent with prior evidence that deep

and context-aware recommender models are better suited to heterogeneous smart-city data than interaction-only approaches [112].

8.2.3 Digital Twin-Based Decision Support

The third finding is that DT modeling extends the framework beyond immediate recommendation ranking. By representing users, urban areas, and PoI as interacting digital entities, the system enables scenario-based examination of alternative actions before they are enacted. In this role, the DT layer connects predictive recommendations with anticipatory reasoning about possible outcomes under current contextual conditions.

The evidence for this finding is scenario-based and analytical. Chapter 7 shows that DT representations can be used to compare alternatives and reason about likely effects on user activity planning and urban conditions. However, this evidence should be interpreted as support for feasibility, architectural coherence, and methodological value rather than as proof of operational performance in a deployed smart-city environment. The finding therefore closes the thesis argument by showing how DTs can become part of the recommendation logic itself, without overstating the validation beyond the scope of the study.

8.2.4 Productivity-Oriented Evaluation

The fourth finding is that personalization can be evaluated in productivity-oriented terms, provided that the evidence is interpreted cautiously. The scenario-based analysis indicates that context-aware recommendations and DT-supported comparison can help align suggested actions with user goals, environmental conditions, and available urban resources. In this sense, the framework supports evaluation beyond accuracy alone by considering productivity-related dimensions such as time efficiency, decision friction, contextual suitability, and the relationship between individual activity choices and wider urban conditions [80].

The level of evidence for this finding is conceptual and scenario-based rather than directly behavioral. The thesis does not claim to measure productivity as a complete ground-truth outcome across real users and deployed urban systems. Instead, it shows that productivity can be framed as an evaluation concern linked to recommendation relevance, contextual fit, and DT-based scenario reasoning. This directly addresses the Chapter 2 gap concerning weak human-centered evaluation while preserving the limitations identified in Chapter 7.

Overall, the findings support the central thesis argument: integrated IoT, AI, and DT technologies can provide a structured basis for context-aware and productivity-oriented urban decision

support. At the same time, the conclusions remain limited by data quality, sensing coverage, privacy, scalability, model fidelity, and transferability across urban contexts. These constraints mean that the framework should be understood as a research prototype and methodological contribution rather than as a fully validated deployment model. Addressing these limitations requires future work on real-world validation, governance, longitudinal evaluation, and scalable deployment under diverse urban conditions [30].

8.3 Limitations

The findings of this thesis should be interpreted within the scope of evidence developed across the preceding chapters. The work addresses the gaps identified in Chapter 2 by connecting IoT-derived context, context-aware recommendation, DT-based scenario reasoning, and productivity-oriented evaluation within a unified framework. It also supports the contributions stated in Chapter 1 through the proposed IoT–AI–DT architecture, the context-aware wide-and-deep model, and the DT-based decision-simulation layer. The evidence presented in Chapter 7 demonstrates the feasibility and comparative value of these components through offline model evaluation, ablation analysis, and scenario-based assessment. The full framework, however, remains evaluated as an integrated research prototype rather than as a long-term deployment in a live urban environment [41].

One important limitation concerns data availability, sensing coverage, and context construction. The framework depends on heterogeneous IoT inputs, including environmental, spatial, temporal, user-related, activity-related, and PoI data. These inputs may vary in completeness, sampling rate, spatial coverage, reliability, and semantic consistency. The preprocessing and context-modeling layers account for these issues through alignment, imputation, quality indicators, and structured representations. However, the resulting recommendations remain conditioned by the quality and availability of the underlying data. For this reason, the reported improvements in contextual relevance and recommendation quality should be understood as results within the studied data setting and would require recalibration and re-evaluation before transfer to other cities, populations, or infrastructure contexts [81].

A second limitation concerns model evaluation and generalization. The proposed context-aware wide-and-deep model is evaluated against selected recommender baselines and ablation variants, and the results in Chapter 7 show that structured contextual features improve predictive performance within the reported experimental setting. This supports the thesis claim that IoT-derived context can strengthen urban recommendations compared with interaction-only approaches. At the same time, the evaluation is based on offline data, selected metrics, and bounded experimental splits. The model results therefore provide evidence of comparative predictive performance within the thesis

setting, while broader operational robustness would need to be assessed through future deployment-oriented and longitudinal studies.

A third limitation concerns the DT and simulation layer. The Human, PoI, and City or Area DTs provide structured abstractions for comparing candidate actions under current contextual conditions. This supports the thesis contribution of extending recommendations beyond ranking toward scenario-based decision support. These abstractions are intentionally task-focused: they represent the user, PoI, and urban conditions needed for recommendation refinement rather than the full behavioral, social, institutional, and infrastructural complexity of a city. Accordingly, the DT outputs should be interpreted as comparative decision-support evidence under modeled assumptions, not as precise forecasts of real-world outcomes [30].

A fourth limitation concerns productivity measurement. The thesis evaluates productivity-oriented support through scenario-based indicators such as contextual fit, decision support, activity planning, access burden, and alignment between user goals and available urban options. This addresses the Chapter 2 gap concerning evaluation beyond accuracy alone and is supported by the integrated analysis in Chapter 7. Direct measurement of productivity through longitudinal user studies, controlled field experiments, or city-scale organizational data remains outside the scope of the present evaluation. The productivity conclusions should therefore be read as evidence that the framework supports productivity-oriented reasoning. In contrast, direct measurement of individual, organizational, or urban productivity remains a direction for future work.

Finally, deployment, privacy, governance, and fairness require further development before real-world implementation. The thesis discusses privacy-aware and decentralized directions, but a deployment-grade framework would also require enforceable privacy guarantees, fairness auditing, accountability mechanisms, institutional oversight, and multi-stakeholder governance. These requirements are especially important because the proposed framework uses contextual, behavioral, mobility, and user-state information through Human DTs and context-aware recommendations. They also align with broader smart-city concerns regarding the translation of ethical principles into operational systems [40, 42]. Taken together, these limitations clarify the level of evidence provided by the thesis and position the work as a foundation for future deployment studies, longitudinal validation, and governance-aware urban intelligence research.

8.4 Future Research Directions

Future research should extend the proposed IoT–AI–DT framework from scenario-based and offline evaluation toward longitudinal, deployment-oriented validation. The results in Chapter 7 support the feasibility of integrating IoT-derived context, context-aware recommendation, and DT-based

scenario reasoning. However, they do not remove limitations related to adaptation over time, real-world deployment, DT fidelity, privacy, human evaluation, governance, and scalability.

8.4.1 Adaptive and Longitudinal Learning

A first direction is adaptive and longitudinal learning. The current framework represents environmental, spatial, temporal, activity-related, and user-related context. However, it does not fully validate how preferences, routines, activity goals, and environmental responses evolve over extended use. Future work should therefore study models that update user states, contextual weights, and recommendation behavior from repeated feedback while preserving uncertainty indicators for missing, imputed, or low-confidence data [81].

8.4.2 Real-World Deployment and Digital Twin Validation

A second direction is deployment in operational urban environments. The framework should be evaluated where IoT streams are incomplete, latency varies, PoI conditions change, and users may accept, reject, or modify recommendations. Practical deployment would require reliable sensing coverage, interoperable data formats and APIs, scalable edge-cloud processing, and clear policies for privacy, consent, data ownership, and access control. A phased deployment, beginning with a bounded district, campus, or service domain, would allow synchronization policies, scalability limits, and governance requirements to be validated before broader city-wide adoption.

This deployment work should be paired with stronger DT validation. In this thesis, DTs support comparative scenario reasoning, but their outputs remain bounded by sensing quality, synchronization assumptions, model fidelity, and scenario-evaluation criteria. Future research should compare simulated relevance, feasibility, comfort, access burden, service availability, and productivity alignment with observed outcomes after recommendations are acted upon. This would clarify when DT reasoning improves decision support and when delayed, missing, or simplified state representations weaken it [30, 92].

8.4.3 Privacy-Preserving Personalization and Human-in-the-Loop Evaluation

A third direction is privacy-preserving personalization. The Human DT and context-aware recommender depend on user-related, spatial, behavioral, and activity signals, which may be sensitive. Future work should evaluate federated learning, local inference, aggregation, anonymization, and privacy-aware context abstraction in terms of their effect on recommendation quality, DT synchronization, and productivity-oriented decision support.

Future studies should also include human-in-the-loop evaluation. Predictive accuracy and scenario quality do not by themselves demonstrate human-centered value. Users should be involved in assessing explanation quality, perceived usefulness, cognitive burden, decision confidence, and conditions under which recommendations are accepted, rejected, or overridden. This feedback should inform later updates to user states, contextual representations, and DT assumptions.

8.4.4 Governance and Scalability

A final direction is governance-aware scalability. The proposed framework identifies edge, fog, and cloud processing as mechanisms for managing latency and computation. However, it does not demonstrate city-scale operation across heterogeneous institutions, sensing infrastructures, and service providers. Future work should therefore examine distributed learning and inference, resource allocation, interoperable DT update policies, and audit mechanisms that maintain bounded latency, accountability, and institutional oversight [62, 115].

Scalability should be evaluated together with fairness, explainability, and contestability because urban recommendations may affect access to services and opportunities. Governance mechanisms should therefore make recommendation outputs auditable and understandable while preserving the advisory role of the system rather than turning it into autonomous urban control [29, 34].

Bibliography

[1] G. Adomavicius and A. Tuzhilin. "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". In: *IEEE Transactions on Knowledge and Data Engineering* 17.6 (June 2005), pp. 734–749.

[2] Gediminas Adomavicius et al. "Context-Aware Recommender Systems". In: *AI Magazine* 32.3 (Sept. 2011), pp. 67–80.

[3] May Altulyan et al. "Recommender Systems for the Internet of Things: A Survey". In: *arXiv preprint* (July 2020).

[4] Gricela Andrade-Ruiz et al. "Emerging Perspectives on the Application of Recommender Systems in Smart Cities". In: *Electronics 2024, Vol. 13, Page 1249* 13.7 (Mar. 2024), p. 1249.

[5] Bokolo Anthony Jnr. "A case-based reasoning recommender system for sustainable smart city development". In: *AI and Society* 36.1 (Mar. 2021), pp. 159–183.

[6] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: A survey". In: *Computer Networks* 54.15 (Oct. 2010), pp. 2787–2805.

[7] Manlio Bacco et al. "Environmental Monitoring for Smart Cities". In: *IEEE Sensors Journal* 17.23 (Dec. 2017), pp. 7767–7774.

[8] Hakan Bagci and Pinar Karagoz. "Random walk based context-aware activity recommendation for location based social networks". In: *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015* (2015).

[9] Linas Baltrunas et al. "Context-Aware Places of Interest Recommendations for Mobile Users". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6769 LNCS.PART 1 (2011), pp. 531–540.

[10] Michael Batty. "Digital twins". In: *Environment and Planning B: Urban Analytics and City Science* 45.5 (Sept. 2018), pp. 817–820.

[11] Claudio Bettini et al. "A survey of context modelling and reasoning techniques". In: *Pervasive and Mobile Computing* 6.2 (Apr. 2010), pp. 161–180.

[12] Simon Elias Bibri and John Krogstie. "Environmentally data-driven smart sustainable cities: applied innovative solutions for energy efficiency, pollution reduction, and urban metabolism". In: *Energy Informatics* 3.1 (Dec. 2020), pp. 1–59.

[13] Joan Borràs, Antonio Moreno, and Aida Valls. "Intelligent tourism recommender systems: A survey". In: *Expert Systems with Applications* 41.16 (Nov. 2014), pp. 7370–7389.

[14] Stefan Boschert and Roland Rosen. "Digital Twin—The Simulation Aspect". In: *Mechatronic Futures*. Cham: Springer International Publishing, Jan. 2016, pp. 59–74.

[15] Houda El Bouhissi et al. "Health Recommender System for Smart Cities". In: *CEUR Workshop Proceedings*. Vol. 3426. 2023, pp. 334–343.

[16] Trevor Braun et al. "Security and privacy challenges in smart cities". In: *Sustainable Cities and Society* 39 (May 2018), pp. 499–507.

[17] H. Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017* (Feb. 2017).

[18] Robin Burke. "Hybrid recommender systems: Survey and experiments". In: *User Modelling and User-Adapted Interaction* 12.4 (2002), pp. 331–370.

[19] Stuart K. Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction*. CRC Press, 1983.

[20] Fran Casino et al. "Healthy Routes in the Smart City: A Context-Aware Mobile Recommender". In: *IEEE Software* 34.6 (Nov. 2017), pp. 42–47.

[21] Kinjal Chaudhari and Ankit Thakkar. "A Comprehensive Survey on Travel Recommender Systems". In: *Archives of Computational Methods in Engineering* 27.5 (Nov. 2020), pp. 1545–1571.

[22] Guanling Chen and David Kotz. *A Survey of Context-Aware Mobile Computing Research*. Tech. rep. USA, 2000.

[23] Qi Chen et al. "A Survey on an Emerging Area: Deep Learning for Smart City Data". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 3.5 (Oct. 2019), pp. 392–410.

[24] Heng Tze Cheng et al. "Wide & deep learning for recommender systems". In: *ACM International Conference Proceeding Series* (Sept. 2016), pp. 7–10.

[25] Ruijie Cheng, Lei Hou, and Sheng Xu. "A Review of Digital Twin Applications in Civil and Infrastructure Emergency Management". In: *Buildings* 13.5 (Apr. 2023), p. 1143.

[26] Diane J. Cook and Sajal K. Das. "Smart Environments: Technology, Protocols and Applications". In: *Smart Environments: Technology, Protocols and Applications* (Feb. 2005), pp. 1–404.

[27] María E. Cortés-Cediel, Iván Cantador, and Olga Gil. "Recommender systems for e-governance in smart cities: State of the art and research opportunities". In: *ACM International Conference Proceeding Series* (Aug. 2017).

[28] Paul Covington, Jay Adams, and Emre Sargin. "Deep Neural Networks for YouTube Recommendations". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. New York, NY, USA: ACM, Sept. 2016, pp. 191–198.

[29] Yashar Deldjoo et al. "Fairness in recommender systems: research landscape and future directions". In: *User Modeling and User-Adapted Interaction* 34.1 (Mar. 2024), pp. 59–108.

[30] Tianhu Deng, Keren Zhang, and Zuo-Jun (Max) Shen. "A systematic review of a digital twin city: A new pattern of urban governance toward smart cities". In: *Journal of Management Science and Engineering* 6.2 (June 2021), pp. 125–134.

[31] Li Deren, Yu Wenbo, and Shao Zhenfeng. "Smart city based on digital twins". In: *Computational Urban Science* 1.1 (Dec. 2021), p. 4.

[32] Sergio Di Martino and Silvia Rossi. "An Architecture for a Mobility Recommender System in Smart Cities". In: *Procedia Computer Science* 98 (Jan. 2016), pp. 425–430.

[33] Peter F. Drucker. "Knowledge-worker productivity: The biggest challenge". In: *California Management Review* 2 (1999), pp. 79–94.

[34] Michael D. Ekstrand et al. "Fairness in Information Access Systems". In: *Foundations and Trends in Information Retrieval* 16.1-2 (July 2022), pp. 1–177.

[35] Abdulmotaleb El Saddik. "Digital Twins: The Convergence of Multimedia Technologies". In: *IEEE Multimedia* 25.2 (Apr. 2018), pp. 87–92.

[36] Gary W. Evans and Lorraine Maxwell. "Chronic noise exposure and reading deficits: The mediating effects of language acquisition". In: *Environment and Behavior* 29.5 (1997), pp. 638–656.

[37] Wenqi Fan et al. "A Comprehensive Survey on Trustworthy Recommender Systems". In: *Comprehensive Survey on Trustworthy Recommender Systems* 1.1 (Sept. 2022), p. 71.

[38] Jesús Arias Fisteus et al. "A Scalable Data Streaming Infrastructure for Smart Cities". In: *CEUR Workshop Proceedings* 1812 (Mar. 2017), pp. 7–13.

[39] Luciano Floridi. "The Ethics of Artificial Intelligence: Principles, Challenges, and Opportunities". In: *The Ethics of Artificial Intelligence: Principles, Challenges, and Opportunities* (Jan. 2023), pp. 1–243.

[40] Luciano Floridi et al. "AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations". In: *Minds and Machines* 28.4 (Dec. 2018), pp. 689–707.

[41] Aidan Fuller et al. "Digital Twin: Enabling Technologies, Challenges and Open Research". In: *IEEE Access* 8 (2020), pp. 108952–108971.

[42] Raj Gaire et al. "Crowdsensing and Privacy in Smart City Applications". In: *Smart Cities Cybersecurity and Privacy*. Elsevier, Jan. 2019, pp. 57–73.

[43] Rogelio Gámez Díaz et al. "Digital Twin Coaching for Physical Activities: A Survey". In: *Sensors* 20.20 (Oct. 2020), p. 5936.

[44] Michael F. Goodchild et al. "Digital twins in urban informatics". In: *Urban Informatics* 3.1 (May 2024), p. 16.

[45] Michael Grieves and John Vickers. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In: *Transdisciplinary Perspectives on Complex Systems*. Cham: Springer International Publishing, Jan. 2017, pp. 85–113.

[46] Jayavardhana Gubbi et al. "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future Generation Computer Systems* 29.7 (Sept. 2013), pp. 1645–1660.

[47] A. Gunawardana and G. Shani. "Evaluating recommender systems". In: *Recommender Systems Handbook, Second Edition* (Jan. 2015), pp. 265–308.

[48] GuoXuetao and LuJie. "Intelligent e-government services with personalized recommendation techniques". In: *International Journal of Intelligent Systems* (May 2007).

[49] Mostafa Haghi et al. "A flexible and pervasive IoT-based healthcare platform for physiological and environmental parameters monitoring". In: *IEEE Internet of Things Journal* 7.6 (2020), pp. 5628–5647.

[50] Negar Hariri, Bamshad Mobasher, and Robin Burke. "Context-aware music recommendation based on latent topic sequential patterns". In: *RecSys'12 - Proceedings of the 6th ACM Conference on Recommender Systems* (2012), pp. 131–138.

[51] Xiangnan He et al. "Neural collaborative filtering". In: *26th International World Wide Web Conference, WWW 2017* (Aug. 2017), pp. 173–182.

[52] Yassine Himeur et al. "A survey of recommender systems for energy efficiency in buildings: Principles, challenges and prospects". In: *Information Fusion* 72 (Aug. 2021), pp. 1–21.

[53] Yassine Himeur et al. "Blockchain-based recommender systems: Applications, challenges and future opportunities". In: *Computer Science Review* 43 (Feb. 2022).

[54] Abir Hoadjli and Khaled Rezeg. "A scalable mobile context-aware recommender system for a smart city administration". In: *International Journal of Parallel, Emergent and Distributed Systems* 36.2 (2021), pp. 97–116.

[55] Xiao Huang. "Computational urban science needs to go beyond computational". In: *Computational Urban Science* 4.1 (July 2024), p. 18.

[56] Ziqi Huang et al. "A Survey on AI-Driven Digital Twins in Industry 4.0: Smart Manufacturing and Advanced Robotics". In: *Sensors* 21.19 (Sept. 2021), p. 6340.

[57] Soo-Yeon Jeong and Young-Kuk Kim. "Deep Learning-Based Context-Aware Recommender System Considering Contextual Features". In: *Applied Sciences* 12.1 (Dec. 2021), p. 45.

[58] Alipujiang Jierula et al. "Study on Accuracy Metrics for Evaluating the Predictions of Damage Locations in Deep Piles Using Artificial Neural Networks with Acoustic Emission Data". In: *Applied Sciences 2021, Vol. 11, Page 2314* 11.5 (Mar. 2021), p. 2314.

[59] Di Jin et al. "A survey on fairness-aware recommender systems". In: *Information Fusion* 100 (Dec. 2023).

[60] Jiong Jin et al. "An Information Framework for Creating a Smart City Through Internet of Things". In: *IEEE Internet of Things Journal* 1.2 (Apr. 2014), pp. 112–121.

[61] Jens Kandt and Michael Batty. "Smart cities, big data and urban policy: Towards urban analytics for the long run". In: *Cities* 109 (Feb. 2021), p. 102992.

[62] Latif U. Khan et al. "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey". In: *IEEE Internet of Things Journal* 7.10 (Oct. 2020), pp. 10200–10232.

[63] Nauman Khan et al. "A low-cost IoT based system for environmental monitoring". In: *Proceedings - 2019 International Conference on Frontiers of Information Technology, FIT 2019* (2019), pp. 173–178.

[64] Samira Khoshahval et al. "A Personalized Location-Based and Serendipity-Oriented Point of Interest Recommender Assistant Based on Behavioral Patterns". In: *Lecture Notes in Geoinformation and Cartography* (2018), pp. 271–289.

[65] Iason Konstantzos et al. "The effect of lighting environment on task performance in buildings – A review". In: *Energy and Buildings* 226 (Nov. 2020), p. 110394.

[66] Yehuda Koren, Robert Bell, and Chris Volinsky. "Matrix Factorization Techniques for Recommender Systems". In: *Computer* 42.8 (Aug. 2009), pp. 30–37.

[67] Deepak Kumar and Nick P. Bassill. "Analysing trends of computational urban science and data science approaches for sustainable development". In: *Computational Urban Science* 4.1 (Nov. 2024), p. 33.

[68] Fedwa Laamarti et al. "An ISO/IEEE 11073 Standardized Digital Twin Framework for Health and Well-Being in Smart Cities". In: *IEEE Access* 8 (2020), pp. 105950–105961.

[69] Yang Liu et al. "Exploring a large-scale multi-modal transportation recommendation system". In: *Transportation Research Part C: Emerging Technologies* 126 (May 2021), p. 103070.

[70] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. "Content-based Recommender Systems: State of the Art and Trends". In: *Recommender Systems Handbook* (2011), pp. 73–105.

[71] Nick Malleson et al. "Digital twins on trial: Can they actually solve wicked societal problems and change the world for better?" In: *Environment and Planning B: Urban Analytics and City Science* 51.6 (July 2024), pp. 1181–1186.

[72] Pablo Mateos and Alejandro Bellogín. "A systematic literature review of recent advances on context-aware recommender systems". In: *Artificial Intelligence Review* 58.1 (Nov. 2024), p. 20.

[73] Neda Mohammadi and John E. Taylor. "Smart city digital twins". In: *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings* 2018-January (Feb. 2018), pp. 1–5.

[74] Conor Morgan, Iulia Paun, and Nikos Ntarmos. "Exploring Contextual Paradigms in Context-Aware Recommendations". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, Dec. 2020, pp. 3079–3084.

[75] Maxim Naumov et al. "Deep Learning Recommendation Model for Personalization and Recommendation Systems". In: *arXiv preprint* (May 2019).

[76] Soroush Ojagh et al. "A location-based orientation-aware recommender system using IoT smart devices and Social Networks". In: *Future Generation Computer Systems* 108 (2020), pp. 97–118.

[77] Oleksii Pasichnyi et al. "Data-driven strategic planning of building energy retrofitting: The case of Stockholm". In: *Journal of Cleaner Production* 233 (Oct. 2019), pp. 546–560.

[78] Charith Perera et al. "Context aware computing for the internet of things: A survey". In: *IEEE Communications Surveys and Tutorials* 16.1 (Mar. 2014), pp. 414–454.

[79] Tieyun Qian et al. "Attribute Graph Neural Networks for Strict Cold Start Recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* 34.8 (Aug. 2022), pp. 3597–3610.

[80] Lara Quijano-Sánchez et al. "Recommender systems for smart cities". In: *Information Systems* 92 (Sept. 2020), p. 101545.

[81] Wajid Rafique, Abdelhakim Senhaji Hafid, and Junaid Qadir. "Developing smart city services using intent-aware recommendation systems: A survey". In: *Transactions on Emerging Telecommunications Technologies* 34.4 (Apr. 2023), e4728.

[82] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized Markov chains for next-basket recommendation". In: *Proceedings of the 19th International Conference on World Wide Web, WWW '10* (2010), pp. 811–820.

[83] Francesco Ricci, Bracha Shapira, and Lior Rokach. "Recommender systems handbook, Second edition". In: *Recommender Systems Handbook, Second Edition* (Jan. 2015), pp. 1–1003.

[84] Timo Ruohomaki et al. "Smart City Platform Enabling Digital Twin". In: *9th International Conference on Intelligent Systems 2018: Theory, Research and Innovation in Applications, IS 2018 - Proceedings* (July 2018), pp. 155–161.

[85] Monica V. Sanchez-Sepulveda et al. "Exploiting urban data to address real-world challenges: Enhancing urban mobility for environmental and social well-being". In: *Cities* 153 (Oct. 2024), p. 105275.

[86] Arun Kumar Sangaiah, S. P. Shantharajah, and Padma Theagarajan. *Intelligent pervasive computing systems for smarter healthcare*. wiley, July 2019, pp. 1–419.

[87] Badrul Sarwar et al. "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the 10th International Conference on World Wide Web, WWW 2001* (Apr. 2001), pp. 285–295.

[88] Mahadev Satyanarayanan. "The emergence of edge computing". In: *Computer* 50.1 (Jan. 2017), pp. 30–39.

[89] Aya Sayed et al. "Intelligent Edge-Based Recommender System for Internet of Energy Applications". In: *IEEE Systems Journal* (Sept. 2021).

[90] Shiva Sepehri et al. "The Effects of Noise on Human Cognitive Performance and Thermal Perception under Different Air Temperatures". In: *Journal of Research in Health Sciences* 19.4 (2019), e00464.

[91] Olli Seppanen, William J. Fisk, and Q.H. Lei. "Effect of temperature on task performance in officeenvironment". In: *5th International Conference on Cold ClimateHeating, Ventilating and Air Conditioning, Moscow, Russia, May 21-24,2006* (July 2006).

[92] Ehab Shahat, Chang T. Hyun, and Chunho Yeom. "City Digital Twin Potentials: A Review and Research Agenda". In: *Sustainability* 13.6 (Mar. 2021), p. 3386.

[93] Qusai Shambour and Jie Lu. "Government-to-Business Personalized e-Services Using Semantic-Enhanced Recommender System". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6866 LNCS (2011), pp. 197–211.

[94] Krishnamoorthi Sivalingam et al. "A Review and Methodology Development for Remaining Useful Life Prediction of Offshore Fixed and Floating Wind turbine Power Converter with Digital Twin Technology Perspective". In: *Proceedings - 2018 2nd International Conference on Green Energy and Applications, ICGEA 2018* (May 2018), pp. 197–204.

[95] Elena Smirnova and Flavian Vasile. "Contextual sequence modeling for recommendation with Recurrent Neural Networks". In: *ACM International Conference Proceeding Series* Part F130153 (Aug. 2017), pp. 2–9.

[96] Petros Spachos and Konstantinos Plataniotis. "BLE Beacons in the Smart City: Applications, Challenges, and Research Opportunities". In: *IEEE Internet of Things Magazine* 3.1 (Mar. 2020), pp. 14–18.

[97] Robert Stackowiak. *Azure Internet of Things Revealed*. Berkeley, CA: Apress, 2019.

[98] Xiaoyuan Su and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques". In: *Advances in Artificial Intelligence* 2009 (Jan. 2009), pp. 1–19.

[99] Fei Sun et al. "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer". In: *International Conference on Information and Knowledge Management, Proceedings* 11 (Apr. 2019), pp. 1441–1450.

[100] Zehua Sun et al. "A Survey on Federated Recommendation Systems". In: *IEEE Transactions on Neural Networks and Learning Systems* 14.8 (Dec. 2022).

[101] Fei Tao et al. "Digital Twin in Industry: State-of-the-Art". In: *IEEE Transactions on Industrial Informatics* 15.4 (Apr. 2019), pp. 2405–2415.

[102] Silvia Liberata Ullo and Ganesh Ram Sinha. "Advances in smart environment monitoring systems using iot and sensors". In: *Sensors (Switzerland)* 20.11 (2020).

[103] Moshe Unger, Alexander Tuzhilin, and Amit Livne. "Context-Aware Recommendations Based on Deep Learning Frameworks". In: *ACM Transactions on Management Information Systems* 11.2 (June 2020), pp. 1–15.

[104] Norha M. Villegas et al. "Characterizing context-aware recommender systems: A systematic literature review". In: *Knowledge-Based Systems* 140 (Jan. 2018), pp. 173–200.

[105] Ana Iolanda Voda and Laura Diana Radu. "Artificial Intelligence and the Future of Smart Cities". In: *BRAIN. Broad Research in Artificial Intelligence and Neuroscience* 9.2 (May 2018), pp. 110–127.

[106] Zehui Wang, Wolfram Höpken, and Dietmar Jannach. "A survey on point-of-interest recommendations leveraging heterogeneous data". In: *Information Technology & Tourism* 27.1 (Mar. 2025), pp. 29–73.

[107] Pawel Wargocki et al. "Perceived Air Quality, Sick Building Syndrome (SBS) Symptoms and Productivity in an Office with Two Different Pollution Loads". In: *Indoor Air* 9.3 (Sept. 1999), pp. 165–179.

[108] Jiancan Wu et al. "Graph convolution machine for context-aware recommender system". In: *Frontiers of Computer Science* 16.6 (Dec. 2022), pp. 1–12.

[109] Carl Yang et al. "Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Vol. Part F1296. New York, NY, USA: ACM, 2017, pp. 1245–1254.

[110] Xinyue Ye et al. "Developing Human-Centered Urban Digital Twins for Community Infrastructure Resilience: A Research Agenda". In: *Journal of Planning Literature* 38.2 (May 2023), pp. 187–199.

[111] Armin Jeddi Yeganeh et al. "Correlation of ambient air temperature and cognitive performance: A systematic review and meta-analysis". In: *Building and Environment* 143 (Oct. 2018), pp. 701–716.

[112] Shuai Zhang et al. "Deep learning based recommender system: A survey and new perspectives". In: *ACM Computing Surveys* 52.1 (Jan. 2019).

[113] Yong Zheng. "Context-Aware Collaborative Filtering Using Context Similarity: An Empirical Comparison". In: *Information* 13.1 (Jan. 2022), p. 42.

[114] Zhi Zhou et al. "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing". In: *Proceedings of the IEEE* 107.8 (Aug. 2019), pp. 1738–1762.

[115] Ivan Zyrianoff et al. "Scalability of Real-Time IoT-based Applications for Smart Cities". In: *Proceedings - IEEE Symposium on Computers and Communications* 2018-June (Nov. 2018), pp. 688–693.