OULU 2023
C 904

ACTA
UNIVERSITATIS OULUENSIS

Hassan Mehmood

CONCEPT DRIFT IN SMART CITY SCENARIOS

UNIVERSITY OF OULU GRADUATE SCHOOL;
UNIVERSITY OF OULU,
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

ACTA UNIVERSITATIS OULUENSIS
C Technica 904

HASSAN MEHMOOD

CONCEPT DRIFT IN SMART CITY SCENARIOS

Academic dissertation to be presented with the assent of the Doctoral Programme Committee of Information Technology and Electrical Engineering of the University of Oulu for public defence in the OP auditorium (L10), Linnanmaa, on 6 November 2023, at 12 noon

UNIVERSITY OF OULU, OULU 2023

Acta Univ. Oul. C 904, 2023

Supervised by
Docent Susanna Pirttikangas
Doctor Ekaterina Gilman

Reviewed by
Professor Kary Främling
Associate Professor Giuseppe Fenza

Opponent
Professor Francisco Camara Pereira

ISBN 978-952-62-3851-7 (Paperback)
ISBN 978-952-62-3852-4 (PDF)

ISSN 0355-3213 (Printed)
ISSN 1796-2226 (Online)

Cover Design
Raimo Ahonen

PUNAMUSTA
TAMPERE 2023

Mehmood, Hassan, Concept drift in smart city scenarios.
University of Oulu Graduate School; University of Oulu, Faculty of Information Technology and Electrical Engineering
Acta Univ. Oul. C 904, 2023
University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

Abstract

Exponential population growth and urbanisation pose potential challenges to mobility, governance, well-being, the environment, and the safety of modern cities. This demands data-driven predictions and decision-making systems to achieve sustainable societal goals. Smart city data are being employed to improve citizens' lifestyles, derive climate initiatives, provide quality health care and education, achieve better governance, and design better urban policies. However, the data from smart cities is vast and heterogeneous, requiring efficient and fault-tolerant data platforms supporting continuous data collection, storage, analysis, visualisation, and results delivery in both batch and real-time fashion. In addition, real-world data brings challenges and may come from malfunctioning, replaced or differently calibrated devices. Concept drift is a crucial barrier to relying on predictions from real-world data streams. It emerges due to unforeseen reasons, changes in statistical properties and the context of data while performing predictive modelling.

This thesis focuses on the challenges mentioned by investigating and proposing efficient concept drift detection approaches, providing distributed data pipeline architectures, and highlighting the potential challenges of concept drift in terms of real-world applicability. As a result, two different algorithms are proposed to perform predictive modelling using machine learning methods integrated with concept drift detection and adaptation methods. The experiment showed that integrating concept drift detection with predictive models increases the effectiveness of drawn predictions. Secondly, a cloud computing-based distributed data pipeline architecture is provided to support data collection, data analysis, concept drift detection, and others. Similarly, an edge computing-based distributed data pipeline is proposed for edge micro data centres to perform computationally demanding processes. The proposed data pipelines are fault-tolerant, can be scaled seamlessly, and support batch and real-time processing, third-party application integration, and more. The overall work has contributed to the existing knowledge base and outperformed current state-of-the-art solutions with real-world use case implementation. Finally, the open issues and challenges of concept drift detection and real-world applicability are discussed.

Keywords: artificial intelligence, big data, concept drift, distributed architectures, internet of things, machine learning, sensors, smart cities, smart city platforms

Mehmood, Hassan, Käsiteliukuma älykaupunkiskenaarioissa.
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähkötekniikan tiedekunta
Acta Univ. Oul. C 904, 2023
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

Tiivistelmä

Eksponentiaalinen väestönkasvu ja kaupungistuminen haastavat modernien kaupunkien hallintoa, logistiikkaa, ihmisten sekä ympäristön hyvinvointia ja turvallisuutta. Kestävien yhteiskunnallisten tavoitteiden saavuttamiseksi voidaan hyödyntää datapohjaisia päätöksentekojärjestelmiä. Älykaupunkien dataa käytetään tukemaan kansalaisten elämäntapoja sekä ilmastoaloitteita, tuottamaan laadukasta terveydenhuoltoa ja koulutusta sekä sujuvoittamaan hallintoa ja kaupunkipolitiikkaa. Kaupungeista saatava data on monimuotoista ja laajaa, saattaa sisältää virheitä ja sen käsittely vaatii tehokkaita ja vikasietoisia data-alustoja. Alustojen on tuettava jatkuvaa datan keruuta, tallennusta, analysointia, visualisointia ja tulosten toimittamista sekä erä- että reaaliaikaisesti. Käsitevuoto (concept drift) on keskeinen este luotettavien ennusteiden tekemiselle reaalimaailman datavirroista. Käsitevuoto ilmenee ennakoimattomista syistä, datan tilastollisten ominaisuuksien ja kontekstin muutoksista ennustavaa mallinnusta tehdessä, johtaen epäluotettaviin datapohjaisiin ennusteisiin.

Työssä tutkitaan käsitevuotoa erityisesti laboratorion ulkopuolella tosimaailmassa. Tuloksena esitetään kaksi erilaista algoritmia käsitevuon havaitsevaan ennustavaan mallinnukseen. Algoritmit ovat koneoppimenetelmiä, joihin on integroitu käsitevuodon havaitsemis- sekä mukautumismekanismit. Työssä osoitetaan, että kun integroidaan käsitevuodon havaitseminen ennustemallinnukseen, voidaan lisätä ennusteiden tehokkuutta. Toiseksi työssä kehitettiin pilvilaskentaan perustuva hajautettu dataputkiarkkitehtuuri, joka tukee datan keruuta, datan analysointia, käsitevuodon havaitsemista ja muita reaaliaikaiseen mallintamiseen tarvittavia datankäsittelymekanismeja. Lisäksi kehitettiin reunalaskentaan perustuva hajautettu dataputkijärjestelmä mikrodatakeskuksineen laskennallisesti raskaita datakäsittelytehtäviä varten. Ehdotetut dataputket ovat vikasietoisia, ne voidaan skaalata saumattomasti ja ne tukevat erä- ja reaaliaikaista käsittelyä sekä kolmansien osapuolten sovellusintegraatioita. Kokonaisuudessaan työssä kehitetyt menetelmät suoriutuvat nykyisiä ratkaisuja paremmin reaalimaailman käyttötapausten toteuttamisessa. Lopuksi käsitellään avoimet kysymykset sekä tulevaisuuden haasteet käsitevuodon havaitsemisen ja reaalimaailman sovellettavuuden suhteen.

Asiasanat: anturit, esineiden internet, hajautetut arkkitehtuurit, koneoppiminen, käsitevuoto, massadata, tekoäly, älykaupungit, älykaupunkialustat

To all those who shine in shadows.

Acknowledgements

I want to thank and acknowledge everyone who guided and supported me throughout this research journey, with special mention to my supervisor, Research Director Susanna Pirttikangas. Without her unwavering support, encouragement, and guidance, achieving this milestone would have been difficult. I am also grateful to my second supervisor, D.Sc. (Tech.) Ekaterina Gilman, for her mentoring, constructive feedback, and introducing distributed computing and concept drift topics.

I'd also like to thank Dr. Panos Kostakos for all the uplifting conversations and ongoing collaboration in the research work. I am also grateful to Professor Timo Ojala and Professor Jukka Reikki for their continuous support and encouragement. I appreciate all discussions and research collaborations with colleagues in the Interactive Edge (iEdge) research group of the Centre for Ubiquitous Computing at the University of Oulu, Finland, and especially Dr. Lauri Lovén, Jacky Cao, Marta Cortés, and Tri Nguyen. I am also thankful for the continuous feedback provided by follow-group members, especially Professor Olli Silvén and Associate Professor Ilkka Nissinen. In addition, I would like to express my sincere thanks to our collaborator, Dr. Ahmed Khalid from DELL EMC, Ireland, for all the brainstorming sessions and research work that has led to the completion of this research journey. Furthermore, I would like to thank Dr. Alexander Subbotin at Nordea for his continuous support and encouragement to complete the research along with industrial career.

My research has been funded by different research projects and funding grants, I would like to thank Professor Rauno Heikkila from the Faculty of Technology and Risto Jurva at the Centre for Wireless Communication, University of Oulu, for funding support provided under the Smart Campus Project. I would also acknowledge the funding organisations such as Nordea Pankin Säätio and Tauno Töning Foundation for the provided personal grants and their interest in my research topic, which was encouraging and I am grateful for.

I express my thanks to Professor Kary Främling at Umeå University, Sweden and Associate Professor Giuseppe Fenza, University of Salerno, Italy, for pre-examining this thesis and their insightful feedback. Furthermore, I respectfully thank Professor Francisco Camara Pereira, Technical University of Denmark, Denmark, for his time and for taking on the role of opponent for the doctoral defence.

Further, I am grateful to my friends Rahman Akber, Bilal Khan, Sofeem Naseem, Dedar Rashid, Asadullah Javed, and Haresh Kumar for being there in need. Lastly, I thank my parents, wife, and family members for their unabated support, belief in my abilities, and everything that contributed to finishing this milestone.

List of abbreviations

IoT	*Internet of Things*
AI	*Artificial Intelligence*
EMDCs	*Edge Micro Data Centres*
KET	*Key Enabling Technologies*
HDFS	*Hadoop Distributed File System*
MQTT	*Message Queuing Telemetry Transport*
ANN	*Artificial Neural Network*
ICT	*Information Communication Technologies*
DBN	*Deep Belief Network*
LSTM	*Long short-term memory*
RNN	*Recurrent Neural Network*
GSTGCN	*Global Spatial-Temporal Graph Convolutional Network*
CNN-LFD	*Convolutional Neural Network-based Levy Flight Distribution*
ANFIS	*Adaptive Neuro-fuzzy Inference System*
kNN	*k-Nearest Neighbours*
MLDPAF	*Machine learning-driven Predictive Analytic framework*
SVM	*Support Vector Machine*
WT-SVM	*Wavelet Transform-Support Vector Machine*
CNN	*Convolutational Neural Network*
PHT	*The Page Hinkley Test*
FCM	*Fuzzy Competence Model*
SPLL	*Semi-Parametric Log-Likelihood*
DDM	*Drift Detection Method*
EDDM	*Early Drift Detection Method*
ADWIN	*Adaptive Windowing*
Meta-ADD	*Meta-Active Drift Detection*
VFDT	*Very Fast Decision Tree*
CVFDT	*Concept Drift Very Fast Decision Tree*
SEA	*Streaming Ensemble Algorithm*
OLIN	*Online Information Network*
DACC	*Dynamic Adaptation to Concept Changes*
OS-ELM	*Online Sequential Extreme Learning Machine*
TML-CD	*Tiny Machine Learning for Concept Drift*
KSWIN	*Kolmogorov-Smirnov Windowing*

MAPE	*Mean Absolute Percentage Error*
MAE	*Mean Absolute Error*
RMSE	*Root Mean Square Error*
API	*Application Programming Interface*
CSI	*Container Storage Interface*
AR	*Augmented Reality*

List of original publications

This thesis is based on a series of peer-reviewed original research papers published in different international scientific forums. The following papers are referenced in the text using their Roman Numerals (I-V).

I Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., ... & Riekki, J. (2019, April). Implementing big data lake for heterogeneous data sources. In 2019 ieee 35th international conference on data engineering workshops (icdew) (pp. 37-44). IEEE.

II Mehmood, H., Kostakos, P., Cortes, M., Anagnostopoulos, T., Pirttikangas, S., & Gilman, E. (2021). Concept drift adaptation techniques in distributed environment for real-world data streams. Smart Cities, 4(1), 349-371.

III Mehmood, H., Khalid, A., Kostakos, P., Gilman, E., & Pirttikangas, S. (2024). A novel edge architecture and solution for detecting concept drift in smart environments. Future Generation Computer Systems, 150, 127-143.

IV Cao, J., Mehmood, H., Liu, X., Tarkoma, S., Gilman, E., & Su, X. (2022). Fighting Pandemics with Augmented Reality and Smart Sensing-based Social Distancing. IEEE Computer Graphics and Applications.

V Mehmood, H., Hiltunen, M., Makkonen, T., Immonen, M., Pirttikangas, S., & Heikkilä, R. (2021). Road map for implementing AI-driven Oulu Smart excavator. In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 38, pp. 819-826). IAARC Publications.

Contents

1 Introduction

1.1 Background and motivation

Digitalisation trends throughout the globe have aided in producing an enormous amount of data. Cities largely contribute to the continuous generation of data through environmental sensors, traffic flow counters, air quality measurements, the internet of things (IoT), and many others (Mehmood et al., 2019). Similarly, efforts towards 6G development, deployment of 5G, and recent advances in technologies such as big data have paved the way towards smart cities. The multitude of technologies, along with artificial intelligence (AI) are contributing towards more efficient and sophisticated smart city platforms, assisting in creating liveable cities for the future. The data from smart cities is being harnessed to develop better governance policies, to support decision-making, and to achieve sustainable societal goals (Androutsopoulou, Karacapilidis, Loukis, & Charalabidis, 2019; Habibzadeh, Kaptan, Soyata, Kantarci, & Boukerche, 2019; Mehmood et al., 2019; B. N. Silva, Khan, & Han, 2018). Such opportunities not only elevate the standard of living for an individual in society but also allow governments to foresee future urban issues, and enact new policies.

Artificial intelligence (AI) plays a pivotal role in the aforementioned developments by enabling new systems to be built on top of robust algorithms and to provide insights into various aspects of an urban spectrum, such as situation-aware traffic controls and water management. The predictions on smart city platforms are drawn from studied patterns in data from heterogeneous sources with variable frequencies, i.e. in batches, near real-time, and real-time. Traditional machine learning algorithms assume the data distribution to be stationary (Pesaranghader, Viktor, & Paquet, 2018). However, these assumptions are not valid when the data continuously changes, which is the case with real-world data. Changes in the statistical properties of a variable for which predictions are made over a certain period due to unforeseen reasons are called concept drift (J. Lu et al., 2018). Concept drift has become a crucial issue for predictions made from large real-world data streams (J. Lu et al., 2018; Wojtkiewicz, Katragadda, & Gottumukkala, 2018). The data may have concept drift in various real-world situations due to differently calibrated sensors, malfunctioning devices, and abrupt switch-off sensors (Gama, Žliobaitė, Bifet, Pechenizkiy, & Bouchachia, 2014). In such situations, if concept drift is not detected and treated for continuously changing data, the model's outcomes will become ineffective (J. Lu et al., 2018). Concept drift is one of the main reasons for the reduced effectiveness of machine learning models (Gama et al., 2014;

J. Lu et al., 2018). Uncertain real-world environments require adaptive machine learning techniques able to detect concept drift and react accordingly (Gilman et al., 2015; J. Lu et al., 2018). Therefore, new topics have been emerged such as adaptive learning, adaptive data-driven prediction, change detection, and active learning, that consider the problem of concept drift (Gama et al., 2014; J. Lu et al., 2018; Park & Kang, 2016).

The data available from smart cities is vast, ever-changing and heterogeneous. For large-scale data analysis, algorithms have incurred higher computational costs (J. Lu et al., 2018). Therefore, analysing such huge amounts of heterogeneous data requires an efficient and suitable infrastructure to support data processing (batch and real-time), large-scale data visualisation, and results delivery to third parties. In addition, better solutions are needed to ensure fault-tolerant diverse data collection from smart cities (Mehmood et al., 2019). In most cases, distributed computing paradigms with cloud computing are used to address these challenges (*CUTLER*, 2018; Mehmood et al., 2019; Mehmood, Kostakos, et al., 2021; Pereira et al., 2022; Sanchez et al., 2014). Cloud computing shows the limitations of high latency, network overhead, and mobility in cloud computing (Khan et al., 2020). On the other hand, the edge computing paradigm not only presents lower computation power than cloud computing, but also ensures lower latency, less network overhead, and exhibits more scalability and fault-tolerant characteristics (Khan et al., 2020; Luo et al., 2019). Several efforts have been made to understand the data challenges at the network's edge in smart cities and IoT deployments (Hossain, Rahman, & Hossain, 2018; Khan et al., 2020; Luo et al., 2019; Xu et al., 2023). Unfortunately, most of the developed urban systems do not consider these data challenges collectively. In addition, due to less or zero support for concept drift detection and adaptation, many smart city platforms and decision-making systems result in reduced system performance with time (J. Lu et al., 2018; Santana, Chaves, Gerosa, Kon, & Milojicic, 2018).

Applications of concept drift detection can be found in different dimensions of smart cities. For example, the predictions drawn from air quality data in a smart city can change with time (Halstead et al., 2022), due to the abrupt or periodic change in its target variables (e.g. level of CO_2). Furthermore, sudden changes in weather or the lifestyle of citizens in a smart city greatly impact the predictive systems utilised for decision-making. Numerous performance constraints can be used to study the predictive accuracy of a model, e.g. variation in weather, seasonality, gentrification, poor urban development policies and others. Similarly, the shopping behaviour of citizens can change due to changing urban mobility patterns in a city, and high sales can be observed in enclosed neighbourhoods and shopping malls in extended wet, snowy weather. Lastly, the sudden disruption caused by situations such as COVID-19 lockdown, stock market crash, fake news, and the gig economy can make predictive models obsolete (R. M. Silva & Almeida, 2021; Suárez-Cetrulo, Cervantes, & Quintana, 2019). Therefore, predictive learning integrated with concept drift detection methods are needed to make effective decision-making systems, Moreover, the applicability of such integration in real-world scenarios should be studied.

This thesis explores the existing AI approaches supporting automated concept drift detection and adaptation using distributed computing paradigms such as cloud and edge computing for smart cities and environments.

1.2 Objectives and scope

This research aims to investigate and develop an automated concept drift detection and adaptation solution for large-scale data streams in smart cities. Additionally, the research examines the current technologies and solutions for data collection, processing, storage, analysis, visualisation, and concept drift detection and adaptation. The research is pursued with the following three research questions:

RQ1. How do AI approaches support concept drift detection in smart services for smart cities
This research question investigates existing state-of-the-art concept drift detection and adaptation methods. Existing concept drift adaptation methods are examined for their integration with predictive models.

RQ2. What architecture is required to implement data pipelines supporting automated concept drift detection and adaptation?
Considering the existing challenges and limitations in the state-of-the-art. The research question aims to implement an architecture for data processing pipelines for continuous data collection, processing, storage, analysis, visualisation, results delivery, and concept drift detection and adaptation.

RQ3. What are the challenges for concept drift applications in real-world scenarios?
The research question focuses on challenges towards the real-world applicability of concept drift detection methods in different domains.

To summarise, **RQ1** reviews and identifies the current methods in the literature for concept drift detection and adaptation for smart services and use cases resembling smart city applications and **RQ2** implements and validates the proposed data pipeline architectures supporting concept drift detection and adaptation in real-world use cases. Whereas **RQ3.** outlines the potential challenges towards concept drift detection real-

Table 1. Contributions of the publications toward each research question.

Research Questions	Publications
RQ1: How do AI approaches support concept drift detection in smart services for smart cities?	II, III
RQ2: What architecture is required to implement data pipelines supporting automated concept drift detection and adaptation?	I, II, III
RQ3: What are the challenges for concept drift applications in real-world scenarios?	II,III, IV, V

world applicability. Table 1 shows how each publication contributes to each research question.

1.3 Research methodology

The research is implemented as constructive research (McGregor, 2018), where existing challenges in the areas of concept drift detection and adaptation and design considerations for developing data pipelines are investigated using the current state-of-the-art. This work proposes and implements predictive models integrated with concept drift detection and adaptation methods using real-world streaming data. Secondly, architectures implementing data processing pipelines are presented to address the existing challenges of data collection, processing, storage, and visualisation and support automated concept drift detection and adaptation for smart services in smart cities. Lastly, the challenges of the application of concept drift detection in real-world scenarios are discussed.

1.4 The publications and their contribution

In this thesis, the author has developed two cloud-based data pipeline solutions (Publications I, II) and one containerised architecture for computationally demanding data pipelines developed on top of the edge computing paradigm (Publication III). In addition, two different approaches were proposed for automated concept drift detection and adaptation (Publications II, III). Table 2 summarises the main technical contributions of this thesis. Publications IV, V discuss the applicability of concept drift detection in other domains, therefore are not added to the list in Table 2. Table 3 presents the author's contribution in each research publication included in this thesis .

A summary of how each publication contributes to the research questions is shown in Table 1. In addition, a separate contribution summary for each publication is provided below:

Table 2. Technical contributions of this thesis.

Publication	Description
I, II	Implementation of the cloud-based data pipeline for heterogeneous data sources supporting data collection, processing, storage, and visualisation.
III	A novel containerised architecture to tackle computationally demanding data pipelines built on edge computing paradigms
II	A concept drift detection and adaptation algorithmic process to benchmark concept drift detection methods and application to real-world use cases
III	An innovative feedback-based approach for automated concept drift detection and adaptation for real-time data streams

Publication I explores and addresses the existing challenges of collecting data from diverse sources with varying acquisition frequencies in smart cities. As global data is increasing enormously, it poses challenges for data collection, storage, management, analysis, and visualisation. In addition, production-ready smart city platforms require fault-tolerant, scalable, and efficient solutions to make data continuously available for processing and analysis and to support decision-making for industry, governance bodies, policymakers, etc.

The publication contributes by providing a robust cloud-based data pipeline solution enabling processes of continuous data monitoring, data collection, data processing, data storage, and both batch and real-time data analysis. The developed platform was tested against real-world streaming data for the city of Cork, in Ireland. Furthermore, the platform can handle data heterogeneity challenges, deploying predictive models, and based on the need, the models can be integrated with concept drift detection and adaptation approaches.

Publication II provides the first version of an automated concept drift detection and adaptation algorithm built on top of cloud-based data pipeline solutions. A new algorithmic process is developed and benchmarked against different predictive models and concept drift detection methods. The work utilises real-world streaming data related to Finland's power consumption to predict future electricity needs, considering other dependent variables such as weather. Four concept drift detection methods from the state-of-the-art are bench-marked against a synthetic dataset with known drifting indices to check their performance. Later, these are integrated with three different time series learners (base learners) to forecast the power consumption nationwide using streaming data. The publication contributes an automated concept drift detection and adaptation algorithm and architecture for a data processing pipeline that can be applied to streaming

Table 3. **Author contribution in each publication for the thesis. LEAD indicates that the author has primarily led the contribution item, CO-LEAD shows that the item was led together with another co-author, and CONTR indicates the author's contribution to the item.**

item	Publication author rank	I 1st	II 1st	III 1st	IV 2nd (shared)	V 1st
conceptualization		LEAD	CO-LEAD	LEAD	CONTR.	LEAD
architecture		LEAD	LEAD	CO-LEAD	CO-LEAD	LEAD
methodology		LEAD	CO-LEAD	LEAD	CONTR.	LEAD
data curation		LEAD	LEAD	LEAD	LEAD	–
analysis		LEAD	LEAD	LEAD	CO-LEAD	CO-LEAD
visualisation		LEAD	LEAD	LEAD	CO-LEAD	–
software		LEAD	LEAD	LEAD	CO-LEAD	–
validation		LEAD	LEAD	LEAD	–	–
original draft		LEAD	LEAD	LEAD	CONTR.	LEAD
revision		LEAD	LEAD	LEAD	–	LEAD

data from IoTs, sensor devices, and smart city sensing platforms to develop adequate decision-making systems.

Publication III proposes a novel containerised hybrid device-edge-cloud continuum based on edge micro data centres (EMDCs) for computationally demanding data pipelines. In addition, a feedback-based algorithm for detecting and adapting to concept drift in real-world scenarios is developed and evaluated against real-world streaming data from a network of sensors in a smart building. The gathered results contribute to understanding how the predictive performance of models can be improved using concept drift detection adaptation approaches. The publication provides recommendations for the seamless integration of tools for analytics-based resource allocation, data governance, and continuous data monitoring.

Publication IV presents an augmented reality system for social distancing that uses IoT-enabled data from the smart environment. The publication contributes to the design, implementation, and evaluation based on a user study. In the presented use case, such methodology can be utilised to evaluate predicted crowdedness with or without concept drift. In addition, recommendations are provided to employ predictive models capable of handling erroneous or drifting measurements from IoT devices.

Publication V contributes a novel hybrid edge-cloud architecture for smart excavators that are abundantly used in civil engineering, mining, and the forest industry. Furthermore, modern excavators have become more perceptive and are often integrated with sensors, 3D cameras, laser beams, and others. In such cases, AI-driven autonomous excavators can result in reduced performance issues, i.e., ineffective object detection and unable to access soil dynamics due to varying environmental and weather conditions, thus, exhibiting concept drift phenomena in deployed predictive models. The publication studies how such challenges can be avoided in real-world scenarios.

1.5 Structure of the thesis

The rest of the thesis is structured as follows. Chapter 2 provides the scientific and technological background needed to address the existing challenges in the scope of the thesis. Chapter three details the research contributions of this thesis and further revisits it in the discussion section along with limitations and future work in Chapter 4. Chapter 5 concludes the thesis.

2 Related Work

2.1 Smart cities

Modern cities play a pivotal role in highlighting the environmental, economic, and societal demands that stem from continuous urbanisation. During the 1950s, cities were inhabited by 30% of the world's population, the number reached 54% in 2014. According to estimations made by the United Nations, about 68.4% of people will live in modern-day cities by 2050 (Shrestha, Mitra, Rahman, & Marzen, 2023; Yin et al., 2015). Urbanisation is an irreversible process that causes significant economic, demographic, and social transformations. In addition, people experience improved living standards with provisions of continuous water supply, smart buildings, healthcare, transportation, sewage systems, up-to-date education facilities, and better economic prospects. However, urbanisation brings various challenges, such as uncontrolled usage of natural resources, impacts on the general climate, water pollution, increased carbon footprint, air pollution, health issues, safety concerns, overcrowding, and others (B. N. Silva, Khan, & Han, 2018; Yin et al., 2015). The above-mentioned challenges influence governance bodies, policymakers, industrialists, and other key stakeholders towards achieving more sustainable and smarter cities aided by cutting-edge technologies.

The emergence of modern technologies has improved connectivity between disjointed segments of cities and has also reshaped how cities were built and defined. Such developments not only contribute towards smarter cities but also assist in improving the life quality of its citizens, for example, well-being applications, route planning through smartphones, e-learning, situation-aware transportation and others (Mehmood, Kostakos, et al., 2021; Puliafito, Tricomi, Zafeiropoulos, & Papavassiliou, 2021). The concept of smart cities first appeared in the early 1990s (Yin et al., 2015), but it remains obscure. The connotations of smart cities differ from technological and people's perspectives. However, it includes the enhancement of the overall quality of life in cities, how cities are built and reshaped with dependence on evolving technologies such as cloud computing, cyber-physical systems, big data, edge and fog computing, the internet of things (IoT), and others(Mehmood, Kostakos, et al., 2021; Puliafito et al., 2021; Yin et al., 2015). As per recent estimations, the global smart cities market for its solutions and services will grow by approximately 2 trillion USD (*Smart City Adoption Timeline*, 2018).

The vast usage of sensor networks, economic platforms, social networks, and healthcare platforms in smart cities makes data highly available for smart city management to improve public safety, waste management, mobility and governance systems (Mehmood, Kostakos, et al., 2021; Puliafito et al., 2021). Similarly, the innovation ecosystem for disruptive technologies like 5G, IoT, smartphones, and sensors provides tremendous opportunities in academia and industry. These technologies are being systematised to make smart city platforms in smart cities for better governance, requiring improved technological solutions to make cities safer and more accessible (Anagnostopoulos et al., 2021; Morello, Mukhopadhyay, Liu, Slomovitz, & Samantaray, 2017). However, with their many benefits, different challenges arise, such as security and privacy issues, data heterogeneity, connectivity of sensors, scalability, and others (Corchado et al., 2021; Syed, Sierra-Sosa, Kumar, & Elmaghraby, 2021). The following section details several smart city initiatives with different or similar goals from different countries around the globe.

Smart city initiatives

Smart cities aim to provide improved living standards after rapid urbanisation growth. The smartness concept in smart cities involves economic, societal, financial, and environmental betterment. In addition, measuring the smartness in cities can be correlated with situation-aware transportation, e-governance, increased employment, individual wealth, and others (B. N. Silva, Khan, & Han, 2018). Currently, many smart city initiatives have been launched, and data from these cities is being harnessed to develop new urban management rules, design long-term urban policies, and support government decision-making (Mehmood, Kostakos, et al., 2021). For example, multiple enabling techniques have been surveyed to reduce energy consumption, waste, and pollution hazards and minimise traffic to make cities more sustainable and eco-friendly (Almalki et al., 2021). One example is Tokyo, which created a plan in 2016 to make it environment friendly by 2020 through effective traffic management plans, reducing energy utilisation, and others (*The action plan for 2020*, 2016). Considering the constant increase in population, China intends to circulate funds worth billions to its approximately 500 cities, destined to incorporate smart services, big data analytics, e-regulation management, and improved infrastructure by 2020 (Ekman, 2020; Hu, 2019; Y. Yu & Zhang, 2019). The U-city project in Korea aims to provide ubiquitous and modern infrastructure as part of their smart city initiative (Mullins & Shwayri, 2016). In Hwaseong-dong tan, a project has been completed with a focus on automating urban spaces, smart traffic flows, and crime prevention. (Mullins & Shwayri, 2016).

In the USA, IoT-driven smart city infrastructure development has been finalised with the intention to invest about 41 trillion USD by 2035, supporting better air quality,

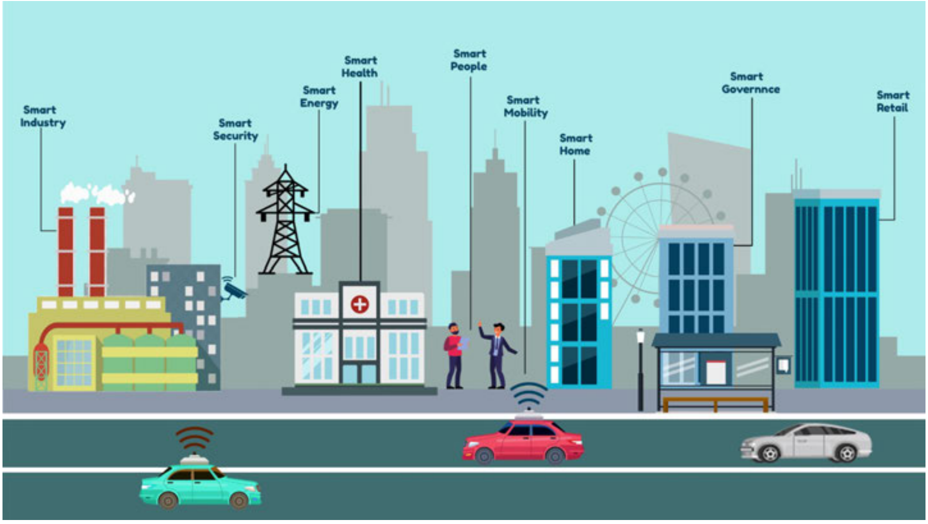

Fig. 1. Overview of a general smart city ecosystem, redrawn from B. N. Silva, Khan, Jung, et al. (2018).

smart transportation, sustainability, and others (*Smart America: Smart Cities USA*, 2014). Similarly, about 1 billion euros were allocated for 300 smart city projects as part of the European Innovation Partnership on Smart Cities and Communities (EIP-SCC). The initiatives primarily focus on improving living standards, smart buildings, climate sustainability, infrastructure digitisation, smart mobility, and others (*General Assembly of the European Innovation Partnership on Smart Cities and Communities (EIP-SCC)*, 2021). Another project called CUTLER targeted five European cities (Antalya, Thessaloniki, Antwerp, Cork, Vicenza) to provide evidence-based decision-making for policymakers in smart cities. The project included different use cases from each city to design better urban management policies, e.g. flood risk estimation and efficient parking systems (*CUTLER*, 2018). Similarly, the Padova Smart City project developed an urban IoT system to monitor the traffic flow, lighting, and pollution (Cenedese, Zanella, Vangelista, & Zorzi, 2014). All of these projects have utilised a combination of modern technologies such as big data, cloud computing, IoT, and others. For example, cloud-based infrastructures are being developed on top of big data technology solutions to store and process large-scale data in projects like CiDAP (Cheng, Longo, Cirillo, Bauer, & Kovacs, 2015), BASIS (Costa & Santos, 2016), Open IoT (Petrolo, Loscri, & Mitton, 2014), CUTLER (*CUTLER*, 2018). However, with growing trends and the adoption of IoT and 5G, edge and fog computing-based alternative are also being explored (Habibzadeh et al., 2019).

Intelligent smart city systems are usually built leveraging the data generated from smart cities and AI. Considering the application segments in smart cities, such as environmental sensing, situation-aware transportation, weather forecasting, flood risk estimation, energy consumption forecasting, and others, the ground truth of data may need to be updated in various cases. These days smart cities require intelligent systems to anticipate future urban issues. However, the understood patterns from historical data in such platforms may become outdated due to faulty devices, calibration issues in sensors, changes in context, sudden changes in weather conditions, and other factors, and this phenomenon is called concept drift (Gama et al., 2014). Examples of concept drift in smart cities can be found in applications such as weather forecasting, fraud detection, financial forecasting, studying mobility behaviours, and intelligent traffic systems (Mehmood, Kostakos, et al., 2021; Morello et al., 2017). In this regard, smart cities require platforms capable of handling computationally demanding processes in addition to concept drift in their forecasting systems.

2.2 Smart city platforms

Generally, smart cities aim to improve citizens' living standards, provide better governance and administration services, and sustainable development. To deliver the promise of smart cities, various salient themes such as smart transportation, healthcare, well-being, administration services, safety and security, and environmental monitoring rely on ICT-driven implementation. (Clohessy, Acton, & Morgan, 2014; Mehmood, Kostakos, et al., 2021). Consequently, safeguarding public assets is a vital obligation for cities, as certain areas may require frequent updates, such as power plants, legal systems, banking, transportation, and public administration systems, and universities may become vulnerable due to outdated systems (e.g. legacy systems) that have been used for decades. (Bellini, Nesi, Paolucci, & Zaza, 2018; Sengan et al., 2020). A typical smart city contains different segments equally contributing towards an intelligent society. A study by (B. N. Silva, Khan, Jung, et al., 2018) identified nine segments which together form a smart city ecosystem: smart homes, smart governance, smart health, smart retail, smart energy, smart people, smart mobility, smart security, and smart industry (see Fig. 1). Many initiatives have gone through great work to design case-driven systems such as parking management, waste disposal, and environmental monitoring (Cenedese et al., 2014; *CUTLER*, 2018). However, much less work is available incorporating all the solutions in a comprehensive smart city platform (Kubler et al., 2022; Luo et al., 2019; B. N. Silva, Khan, & Han, 2018).

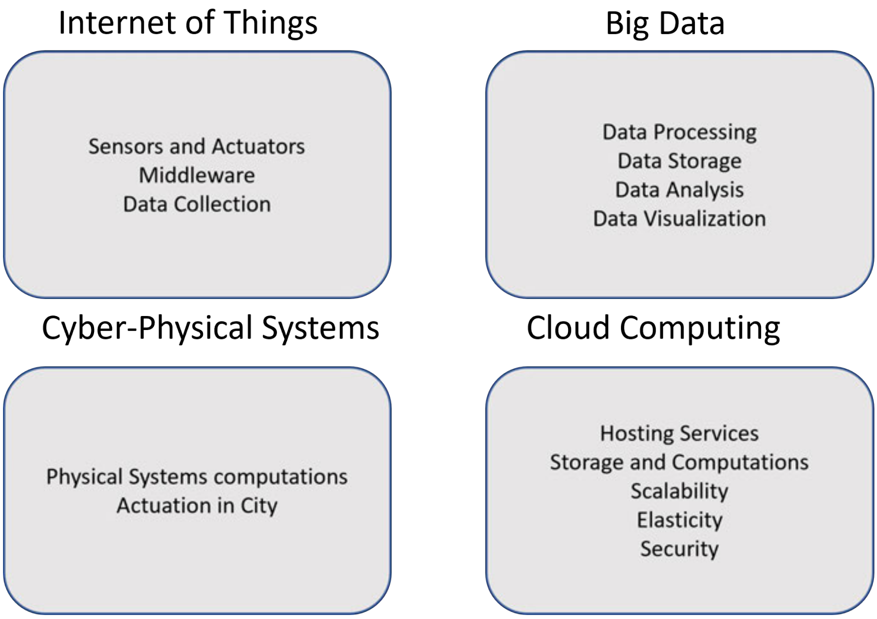

Fig. 2. Key Enabling Technologies in smart cities, redrawn from Santana, Chaves, Gerosa, Kon, and Milojicic (2017).

Managing and processing large-scale data is still challenging for the realisation of a smart city ecosystem (Mehmood et al., 2019). For example, the data in smart cities originates from diverse sources such as environmental sensors, economic platforms, traffic counters, analysers, and social platforms, making the collection, storage, processing, analysing, and visualising of the data demanding. Therefore, utilisation of big data technology solutions aided by distributed computing is well-suited for smart city platforms (Cheng et al., 2015; Costa & Santos, 2016; Mehmood et al., 2019). The need for efficient and innovative smart city platforms is still evident due to disintegrated progress and focus towards individual use cases, e.g. intelligent traffic platforms, environmental sensing, energy management, and others (Cenedese et al., 2014; Mehmood et al., 2019). In addition, the rapid increase in data from smart cities can also result in performance degradation issues on the platforms when it comes to continuously processing huge amounts of data. According to Santana et al. (2017), there are four major key enabling technologies (KETs) used on smart city platforms: i) cyber-physical systems, ii) cloud computing, iii) IoT, and iv) big data. However, some recent studies include fog and edge computing as KETs in smart cities (Khan et al., 2020; Perera, Qin, Estrella, Reiff-Marganiec, & Vasilakos, 2017). One of the reasons behind using fog/edge computing is to make the big data smaller and minimise the computational and communication cost of sending the sensed data to the cloud (Perera et al., 2017). The main KETs in smart city platforms are defined below and can be seen in Fig. 2.

Internet of Things. IoT is often defined as a set of uniquely identifiable networked physical objects connected to the Internet, capable of collecting and exchanging data with other devices (Santana et al., 2017). The physical objects can be of many types, varying from sensors and embedded devices to complex machines like vehicles and daily usage appliances. In smart cities, many devices contribute to data collection from cities and enable the use of IoT. The data accumulated from these devices is transferred and processed on platforms to develop different smart city services (Zanella, Bui, Castellani, Vangelista, & Zorzi, 2014).

Cyber-physical systems. Cyber-physical systems can be defined as systems that utilise computational and communication technologies to enhance physical system features seamlessly. The capabilities of cyber-physical systems are leveraged in many real-world applications such as traffic incident detection systems, medical devices, power grid control systems and others (Santana et al., 2017). According to (Wan, Hughes, Man, & Krilavičius, 2010), current information communication technologies (ICT) solutions do not support applications with continuously changing physical contexts. Employing cyber-physical systems in smart city applications can help mitigate such challenges.

Big data. Big data refers to large-scale and complex data sets that cannot be processed easily using conventional data processing approaches, requiring distributed computing paradigm-based technology solutions. Big data is often characterised by the 4Vs: volume, velocity, variety, and veracity(Santana et al., 2017). Big data applications can be found in smart cities, healthcare, government, transportation, etc. The data generated in smart cities is distributed across different sectors, making it complex and massive due to continuous data generation at varying frequencies. As data sources in smart cities are distributed in different sectors, they each generate data continuously with variable frequencies, thus making it complex and massive. Many big data techniques and tools are widely utilised on smart city platforms such as Apache Spark, HBase, Apache Storm, Apache Kafka, HDFS, Elastic Search, and others (Cheng et al., 2015; *CUTLER*, 2018; Mehmood et al., 2019; Santana et al., 2017).

Cloud computing. Cloud computing offers flexible, large, and highly available Internet infrastructure without needing an on-premise physical infrastructure for different use cases (Santana et al., 2017). In smart cities, cloud computing infrastructures manage, store, process, and deliver data in batch and real-time fashion (Mehmood et al., 2019; Santana et al., 2017). This is an ideal approach to contain data generated from diverse sources such as IoT, economic platforms, social networks and others in smart cities using big data technology solutions (e.g. Apache Hadoop (*Apache Hadoop*, 2006), Apache Spark (*Apache Spark*, 2018) etc.). Cloud computing capabilities also support non-functional security requirements, scalability, fault-tolerance, etc., on smart city platforms (Santana et al., 2017).

Edge Computing Edge computing is based on a distributed computing paradigm that enables the processing and analysing of data at or near the data source. Unlike cloud computing, the generated data from sources is not sent to acentral data repository or cloud to be processed (Khan et al., 2020). Edge computing addresses the limitations of high latency, zero support for mobility, and the extensive overhead of sending data to and from devices in smart cities. In addition, edge computing extends the many advantages of cloud computing, such as scalability, storage capability, multi-tenancy, and elasticity (Khan et al., 2020; Maltezos et al., 2021). Thus, edge computing is viable for modern smart city platforms, especially in IoT-driven smart environments. Examples of edge computing application areas in smart cities may involve smart buildings, waste plants, autonomous vehicles, smart grids, smart agriculture, smart waste bins, and smart streets, to list a few (Cicirelli, Guerrieri, Spezzano, & Vinci, 2017; Khan et al., 2020; Maltezos et al., 2021). Edge computing can facilitate real-time insights and data analysis development in such scenarios. This makes edge computing cost-efficient and essential to the smart city ecosystem.

Review of Smart city platform architectures

This section provides an overview of smart city platform architectures proposed in the state-of-the-art. The data generated from smart cities is enormous, and efficient platforms are needed to collect, store, process, and analyse data from different sources within the smart city ecosystem. Examples of smart city platforms, including their goals and key technologies utilised for their development, are provided in Table 4.

Table 4. Overview of Smart city platforms, adapted under CC BY 4.0 license from Publication III.

Project Name	Aim	Technologies
SmartSantander (Sanchez et al., 2014)	The platform's main goal was to integrate heterogeneous devices such as NFC, sensors, and actuators, to enable continuous monitoring, smart applications, and services	It was implemented using a three-tiered server architecture, including an IoT gateway, server tier, and IoT device tier.
Sii-Mobility Smart City (Zanella et al., 2014)	The primary purpose of this solution was to provide an efficient platform for sustainable mobility and transportation.	A multi-tier cloud-based solution to support big data analytics through tools like Apache Zeppelin, HDFS, and HBase
Padova Smart City (Zanella et al., 2014)	An urban IoT system developed to enable environmental monitoring using data related to noise, CO levels, air temperatures and others.	The architecture was designed based on a web service-like approach leveraging protocols such as EXI, CoAP, IPV4/IPV6, and others.
Snap4City (Badii et al., 2018)	To facilitate the development of multiple smart city applications enabled by heterogeneous data from IoT, big data analytics	A combination of different tools were configured on top of cloud-based infrastructure, such as Apache Spark, Apache Nifi, MQTT, Hbase, and others.
CiDAP (Cheng et al., 2015)	Big data technology-driven platform was developed to mitigate the challenges of data storage and processing from SmartSantander (Sanchez et al., 2014).	Tools like Apache Spark, HDFS, CouchDB, and others were used to develop the platform.

CUTLER (*CUTLER*, 2018)	The project was developed to shift intuition-based decision-making to data-driven decision-making for policymakers, especially in European coastal cities. Heterogeneous data from different smart cities were collected in the developed platform to enable the data-driven decision-making process.	Several technology solutions, such as HDFS, Apache Kafka, Elastic Search, and others, were configured on top of a hybrid cloud infrastructure.
Architecture for an IoT-based smart city (Hossain et al., 2018)	An edge computing paradigm-based layered approach was proposed for situational awareness in smart cities. However, computationally demanding processes like data processing are executed in the cloud.	MQTT, WSN, and Cassandra were used to implement the solutions.
iSapiens (Cicirelli et al., 2017)	A Multi-layered edge computing paradigm-based architecture was proposed for online analytics in smart city applications. However, advanced analytics or computationally demanding processing were done as an offline analysis leveraging cloud services.	The platform was designed using distributed edge computing to enable the connectivity of devices and analytics, along with an off-network database and a web interface.
CTwin platform (Xu et al., 2023)	This platform was developed to enable real-time situational awareness of urban transportation driven by cyber-physical controls, simulations, and analytic dashboards.	A cloud-based distributed platform was implemented using open-source tool sets, such as Angular, Kubernetes, and Docker.

ATCLL data platform (Vítor, Rito, Sargento, & Pinto, 2022)	A data platform was proposed for Aveiro Tech City Living Lab to facilitate data collection, processing, and visualisation for environmental data, mobility, and network data. The platform enables open access for third parties for experimentation and application development.	The platform was implemented using the FIWARE environment, Apache Kafka, and MongoDB.

Table 4 shows that much effort has been put into implementing and deploying several smart city projects. However, in most cases, case-specific platforms are designed to address different challenges separately, such as integrating heterogeneous data, sustainable mobility and transportation, environmental monitoring, big data analytics, and computationally demanding processes. Thus, the need for more comprehensive smart city platforms is evident that can address different smart city use cases from a unified solution. In addition, most smart city platforms leverage the cloud computing paradigm for platform development and deployment. With the adoption of 5G and progressive growth in the edge computing paradigm, smart city architectures must be implemented closer to the data sources. In cloud computing, data is moved to a central entity for processing, storage and analysis. In edge computing, the computations occur at the data sources themselves, provisioning cost effectiveness, energy efficiency, mobility, and low latency (Hossain et al., 2018). The architectures from Cicirelli et al. (2017); Hossain et al. (2018) utilise the edge computing paradigm for platform development but handle computationally demanding processes either on off-network infrastructure or the cloud. The platform called CTwin proposed by Xu et al. (2023) addresses most of the challenges of providing a holistic smart city platform. However, it lacks measures for continuous monitoring system-level logs and analytics-driven resource allocation for large or complex jobs in the long run. Such requirements must be addressed with growing intelligence on smart city platforms.

Intelligence on smart city platforms requires the employment of predictive models. At this stage, most platforms carry out predictive modelling and forecasting. However, the statistical properties or the ground truth in the data do not remain in the same dynamic environments (e.g. COVID reshaped societal dynamics like changes in mobility patterns), requiring the integration of concept drift detection methods with predictive models. Therefore, more edge computing-based smart city platforms are needed to ensure computationally demanding processes at the edge, address low latency issues, enable analytics-driven resource allocation, and support concept drift detection and adaptation for efficient intelligence on smart city platforms.

2.3 Artificial intelligence in smart cites

Smart cities generate vast amounts of data from diverse sources such as IoT, environmental sensors, financial institutes, economic platforms, social networks, governance systems, and many others (Mehmood, Kostakos, et al., 2021). In addition, due to rapid urbanisation, cities need to focus on resolving issues related to traffic congestion, pollution, well-being, water management, and others. The stakeholders in smart cities are able to harness the data from the sources mentioned earlier to develop many smart applications that can help to address contemporary urban issues. Governments throughout the globe are initiating many smart city projects leveraging cloud computing, edge computing, IoT, big data, and artificial intelligence to achieve the objective of smart and sustainable cities (Ashwini, Savithramma, & Sumathi, 2022; Mehmood, Kostakos, et al., 2021).

The use of artificial intelligence in smart cities has been of great discussion lately. In addition, many underdeveloped countries are using AI to move forward in achieving the United Nations' sustainable development goals (Herath & Mittal, 2022). AI-driven smart cities bring out many benefits, such as ensuring an adequate water supply, forecasting energy needs, situation-aware traffic control, providing an eco-friendly environment and decision-making system for governance bodies, and others. (Herath & Mittal, 2022; Mehmood, Kostakos, et al., 2021). The broad prospects of AI and 5G have also enabled monitoring environmental health parameters, such as pollutant emissions, radiation levels, and water contamination. However, these opportunities in smart cities are data-driven and require fault-tolerant data collection platforms and mechanisms to ensure continuous monitoring (Ashwini et al., 2022). The rest of this section reviews the AI adoption in different dimensions of smart cities, such as smart governance, smart education, smart energy, smart transportation, smart waste management, and smart health.

AI in smart Governance. Decision-making in city governance is crucial for planned and sustainable urban development. Nowadays, the capabilities of AI are being exploited to design and develop efficient decision-making platforms in smart cities. In addition, AI and other modern technologies have reshaped the decision-making process from intuition-based to evidence-based, backed by the vast data in smart cities. In a study by (Androutsopoulou et al., 2019), an AI-based chat bot was introduced to ensure efficient communication between the government and its citizens using an Artificial Neural Network (ANN). A regression-based model outlined the decision-making process for building smart cities.

Furthermore, a study by (Allam & Dhunny, 2019) proposed a system that connects AI and cities, assuring the incorporation of prominent aspects in cities such as governance, culture, and others. During the COVID-19 epidemic, the South Korean government introduced protocols leveraging AI to ensure information exchange, assistance to citizens, and the implementation of safety measures. Similarly, an opinion mining approach was utilised to analyse public reactions to local ordinances for smart governance (Puri, Varde, Du, & De Melo, 2018). Recently, federated learning-based approaches to support smart governance were also studied in a study by (Pandya et al., 2023). Therefore, such AI-driven smart governance initiatives contribute towards better policy making, planning for possible city uncertainties, providing enhanced leadership, and more.

AI in smart education. AI-based applications have acquired a significant amount of attention in the education sector. In Finland, a smart education system called SysTech was launched for continuous learning in 2011 (Mäkelä, Helfenstein, Lerkkanen, & Poikkeus, 2018). Similarly, Australia cooperated with IBM to develop a multi-disciplinary education system to connect different educational institutes country-wide (Herath & Mittal, 2022). The adoption of smart devices has enabled the development of smart learning. A project called the Robobo Smart city education framework was introduced with two main components, i) Robobo, a smartphone-based robot ii) a real-world smart city model (Juanatey, Naya, Baamonde, & Bellas, 2021). In addition, the continuous development of smart education applications has provided many opportunities for students, enabling e-learning. A convolutional neural network-based levy flight distribution (CNN-LFD) method was proposed to predict students' learning styles in another study. The method included two dimensions: the automated prediction of learning style and a number of learning styles using a classification approach (Alshmrany, 2022). The role and capabilities of AI in the education sector are also discussed in a study by (Ashwini et al., 2022). A recent study by (Thurzo, Strunga, Urban, Surovková, & Afrashtehfar, 2023) explored an AI-based tool called ChatGPT to study the impact of AI in dental education and how the current curriculum can adapt to such developments.

AI in smart energy. With the exponential increase in population and industrialisation around the globe, reliance on energy and CO_2 emissions has expanded. Such developments require more energy and can lead to serious environmental issues. Integrating AI with traditional energy platforms has helped anticipate cities' consumption and production demands beforehand. Several AI developments have been examined in sustainable smart cities (ElHusseini, Assi, Moussa, Attallah, & Ghrayeb, 2020; Hernández-Ocaña, Hernández-Torruco, Chávez-Bosquez, Calva-Yáñez, & Portilla-Flores, 2019; N. M. Kumar et al., 2020; Lilliu, Loi, Recupero, Sisinni, & Vinyals, 2019). In addition, different AI-based energy forecasting approaches have been proposed, e.g. AI-driven energy cost optimisation (T. Lu et al., 2020), AI-based energy optimisation (Hernández-Ocaña et al., 2019), and others. Fully automated smart grid systems integrated with efficient prediction systems have become necessary. A study by (Selim, Zhou, Feng, & Quinsey, 2021) has proposed deep learning and gradient-boosting methods for measuring the short-term uncertainty in electricity demand. Additionally, deep neural networks have been utilised to forecast cool and heat loads in different structures (Sadeghi, Younes Sinaki, Young, & Weckman, 2020). At the same time, (Bourhnane et al., 2020) examined different models to predict energy consumption and finalised an artificial neural network and genetic algorithms approach. More AI initiatives regarding forecasting and automated smart grid systems are discussed in (Ashwini et al., 2022; Herath & Mittal, 2022).

AI in smart transportation. The exponential increase in population centres and rapid urbanisation has greatly affected transportation, especially in large cities. Such developments have yielded limited transportation infrastructure, inefficient traffic flow, traffic congestion, unavailability of parking spaces, and more (Ashwini et al., 2022; G. Chen & Zhang, 2022; Herath & Mittal, 2022). Therefore, designing smart and sustainable transportation systems using modern technologies is immensely needed. A study by (G. Chen & Zhang, 2022) proposed a deep belief network (DBN) to predict traffic congestion as a reference for intelligent transportation in smart cities. An innovative deep learning-based model called global spatial-temporal graph convolutional network (GSTGCN) was implemented to predict urban traffic speeds in a study by (Ge, Li, Wang, Chang, & Wu, 2020). In South Korea, an long short-term memory-recurrent neural network (LSTM-RNN) was deployed to predict and analyse traffic congestion issues on the Gyeongbu expressway (Yi, Bui, & Jung, 2019).

Furthermore, (P. Liu, Zhang, Kong, & Yin, 2019) proposed a neural network to foresee bus traffic flows using bus patterns and residual networks. Apart from the availability of efficient traffic infrastructures, bicyclists' driving behaviour and mobility also play a pivotal role towards smooth traffic flows, e.g. involving the exit trajectory from roads, use of mobile phones by drivers, and more. (Herath & Mittal, 2022) has outlined some research on driving behaviour, pedestrian risk events, and other safety issues. The use of ICT technologies together with AI for sustainable and reliable transport management systems is discussed in (Ashwini et al., 2022). However, forecasting traffic flows, generating map routes, and autonomous vehicles gradually coming to the market require low-latency traffic solutions with guaranteed trustworthiness (Ke et al., 2020; Pandya et al., 2023; Peyman et al., 2021; Wang, Zhang, Wang, Ma, & Liu, 2020).

AI in smart waste management. The adoption of AI in waste management and environmental protection has become a prominent discussion topic. The rising population and urbanisation have triggered economic growth and boosted waste generation (Herath & Mittal, 2022). The World Bank has forecasted that solid waste will weigh approximately 3.4 billion tons by 2050 (Waste, 2018). AI-based environmental monitoring systems have been proposed by (Ighalo, Adeniyi, & Marques, 2021; Shaikh, Naidu, & Kokate, 2021; Sunny, Zhao, Li, & Sakiliba, 2020). A study by (Abbasi & El Hanandeh, 2016) implemented four AI-based algorithms, adaptive neuro-fuzzy inference system (ANFIS), support vector machine (SVM), k-nearest neighbours (kNN), and ANN to predict monthly waste generation in Queensland, Australia. Similarly, a machine learning-based predictive platform called machine learning-driven predictive analytic framework (MLDPAF) to plan and predict waste for energy and waste management (Huang & Koroteev, 2021). (Abbasi, Abduli, Omidvar, & Baghvand, 2014) utilised SVM and wavelet transform-support vector machine (WT-SVM) models to forecast solid waste generation. In China, a construction and demolition waste forecasting system was proposed by (Song, Wang, Liu, & Zhang, 2017). A study by (J. Chen, Huang, BalaMurugan, & Tamizharasi, 2021) discusses the potential hazards of electronic waste. It proposes an artificial intelligence technique to analyse the effect of pollutants from electronic waste on the environment, human health, and management policies.

AI in smart health. The adoption of AI in different components of smart cities is wider than the aforementioned smart city dimensions. Its use is becoming prominent in the healthcare sector, for example. LSTM method was used in the decision support system for home care assistance, convolutional neural network (CNN) to predict diseases using imagery data, LSTM was employed to forecast patients' health with a focus on diabetes, and others (Juyal, Sharma, & Shukla, 2021; Massaro, Maritati, Giannone, Convertini, & Galiano, 2019; Tuli et al., 2020). Similarly, AI methods, such as fuzzy logic, RNN, LSTM, random forest in smart agriculture, and precision farming, are discussed in (Ashwini et al., 2022; Herath & Mittal, 2022; Pathan, Patel, Yagnik, & Shah, 2020; Sharma, Georgi, Tregubenko, Tselykh, & Tselykh, 2022).

Most smart cities are now integrating modern technologies to address the rising challenges of transportation, waste management, efficient education system, industrial automation, AI-driven healthcare, and many others (Herath & Mittal, 2022). A study by Ashwini et al. (2022) has reviewed the use of AI in other sectors of smart cities, such as smart manufacturing, smart town planning, and smart security. However, the employed

AI models in smart cities may produce ineffective predictions due to concept drift. As the data in smart cities is non-stationary and its characteristics are ever-changing, for example, in the case of smart houses, predicting energy consumption may become ineffective due to changes in family structure, having a secondary house, and others (Fenza, Gallo, & Loia, 2019). Similarly, deploying IoT devices or sensor devices to measure and forecast environmental parameters may be affected due to calibration issues, faulty devices, changes in mobility patterns in spaces, and other reasons (Elwell & Polikar, 2011; Gama et al., 2014; Mehmood, Kostakos, et al., 2021). Therefore, with the increasing adoption of AI in smart city ecosystems and other domains, integration of concept drift detection and adaptation methods are needed to ensure effective predictions and address the degradation of predictive models over time (Agrahari & Singh, 2022; Bifet & Gavalda, 2007; Cerquitelli, Proto, Ventura, Apiletti, & Baralis, 2019; Fenza et al., 2019; Naqvi, Rehman, & Islam, 2022; L. Yang, Manias, & Shami, 2021).

2.4 Concept drift

Concept drift is a multi-faceted problem, often informally presented as virtual or real drift. The virtual or temporary drift originates from a partial representation of data distribution instead of a change in the underlying concept. In comparison, real drift

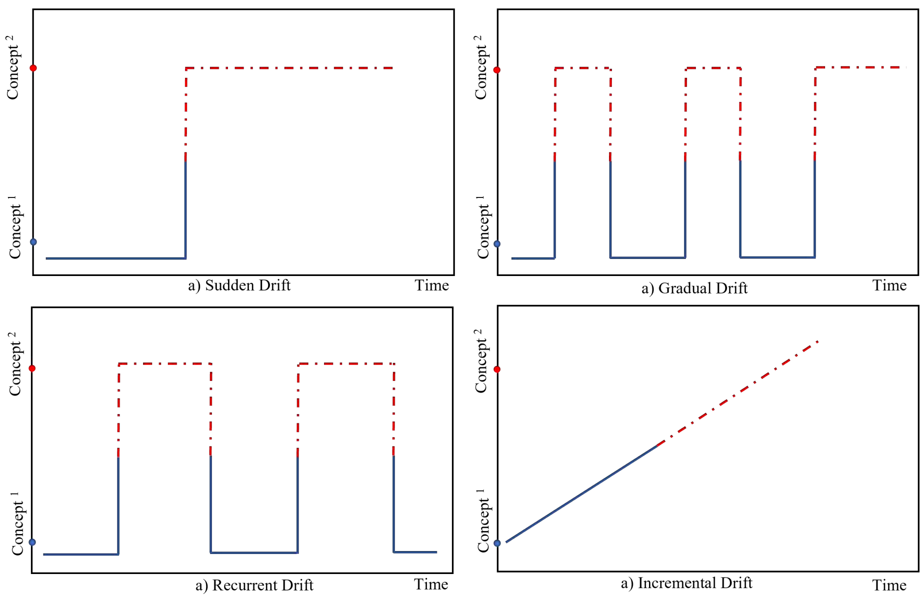

Fig. 3. Different types of concept drift, reprinted under CC BY 4.0 license from Publication III.

refers to the change in the conditional distribution of the target variable, regardless of any change in the dependent variable distribution (Elwell & Polikar, 2011; Gama et al., 2014). Real drift scenarios usually require feedback-based approaches to evaluate the learners' performance continuously for adaptation. On the other hand, virtual drift can be adapted without the need for such approaches. Various reasons can trigger a concept drift in the data, such as malfunctioning devices, changes in data collection mechanisms, calibration issues, context changes, and other issues. Depending upon the generation frequency of changes in data, a concept drift can be abrupt, gradual, or recurring. In an abrupt drift, the change in concept happens suddenly. In a gradual or incremental drift, the change in concept happens slowly over time (Gama et al., 2014; Krawczyk & Cano, 2018). In a recurring drift, the change in data can occur and vanish at a particular time and may return after a certain time (Krawczyk & Cano, 2018; Sun, Wang, Bai, Dai, & Nahavandi, 2018). A visual representation of different types of concept drifts can be seen in Fig. 3.

Concept drift algorithms for adaptive machine learning are usually categorised into two main approaches: passive approaches and active approaches. Passive approaches do not require explicit concept drift detection before updating learning models with the assumption of concept drift present with evolving data. On the other hand, active approaches require the detection of concept drift first and updating the model at a later stage. Table 5 provides an overview of concept drift methods with their type (active or passive), purpose (suitability for different types of drift), and support for classification or regression problems.

Table 5. Overview of Concept drift detection methods and their properties, adapted under CC BY 4.0 license from Publication II.

Method	Type	Purpose	Support
The Page Hinkley Test (PHT)	Active	This method was developed to identify change using Gaussian Signal data and is considered a suitable method for abrupt drift cases. (Gama et al., 2014).	Classification and Regression (Alberg, Last, & Kandel, 2012; Bifet & Gavalda, 2007)

Fuzzy competence model drift detection (FCM)	Active	A data distribution-based method that utilises a competence model to detect concept drift in abrupt and gradual drift scenarios u(Dong, Zhang, Lu, & Li, 2018).	Classification and regression (Dong et al., 2018).
Semi-parametric log-likelihood (SPLL)	Active	The method is suitable for abrupt drift cases. However, in case of gradual changes in data distribution, the method's efficiency may be less optimal (Kuncheva, 2011).	Classification (Kuncheva, 2011).
Drift detection method (DDM)	Active	This method can detect concept drift in both abrupt and gradual drift cases. However, it is considered more suitable for an abrupt drift (Kadwe & Suryawanshi, 2015).	Classification and regression (Cavalcante & Oliveira, 2015; Harel, Mannor, El-Yaniv, & Crammer, 2014).
QuantTree	Active	A recursive binary splitting strategy is applied for change detection. The method requires further experimentation for abrupt and gradual drift (Carrera, 2020; A. Liu, Lu, & Zhang, 2020).	classification (Boracchi, Carrera, Cervellera, & Maccio, 2018).
Early drift detection method (EDDM)	Active	This method is an extension of DDM, and it was developed to address the limitations of detecting gradual drift(Kadwe & Suryawanshi, 2015).	classification and regression (Harel et al., 2014).

Adaptive Windowing (ADWIN)	Active	A sliding windowing approach was introduced to detect concept drift with data sequence bounded to [0,1]. Experiments have been conducted for both abrupt and gradual drift (Gonçalves Jr, de Carvalho Santos, Barros, & Vieira, 2014; Patil, 2019).	Classification and regression (Kaneko, Miyaguchi, & Yamanishi, 2017).
Meta-ADD	Active	An active drift detection method was developed leveraging neural networks. The proposed method automatically detects different types of drifts without needing hypothesis testing (H. Yu et al., 2022).	Classification and regression (H. Yu et al., 2022).
Very fast decision tree (VFDT)	Passive	A decision tree-based method that adapts to concept drift in streaming data. However, retraining can be time crucial for large-scale streaming data; therefore, more than basic incremental learning methods are needed to handle concept drift in non-stationary environments(Brzezinski & Stefanowski, 2013).	Classification and regression (Brzezinski & Stefanowski, 2013; Domingos & Hulten, 2000).
Concept drift very fast decision tree (CVFDT)	Passive	This method is an extension of VFDT to adapt to concept drift in streaming data. However, the method is inefficient for abrupt drift scenarios (Brzezinski & Stefanowski, 2013; Nguyen, Woon, Ng, & Wan, 2012).	Classification and regression (Brzezinski & Stefanowski, 2013; A. Kumar, Kaur, & Sharma, 2015).

Streaming ensemble algorithm (SEA)	Passive	This method performs inefficiently in abrupt drift cases as the learners created for past data may still be valid with inaccurate weights (Brzezinski & Stefanowski, 2013).	Classification and regression (Ditzler, 2016; Elwell & Polikar, 2011).
Online information network(OLIN)	Passive	This model works by generating a network using a windowing approach, while concept drift is detected using the increase in error rate. However, the method has a higher computational cost (Cohen, Avrahami-Bakish, Last, Kandel, & Kipersztok, 2008).	Classification (Cohen et al., 2008).
Online non-stationary boosting	Passive	A boosting approach-based method capable of learning in a non-stationary environment. The method has been tested against hidden contexts (e.g. abrupt or gradual drifts) (Pocock, Yiapanis, Singer, Luján, & Brown, 2010).	Classification (Pocock et al., 2010).
Dynamic adaptation to concept changes (DACC)	Passive	This model follows a deletion strategy to discard outdated learners, enabling higher reactivity to concept drift in non-stationary environments. It supports abrupt and gradual drifts(Cano & Krawczyk, 2020).	classification (Cano & Krawczyk, 2020).

Online sequential extreme learning machine(OS-ELM)	Passive	The proposed method has demonstrated higher efficiency in detecting several kinds of drifts (abrupt, recurrent, and others) (Z. Yang, Al-Dahidi, Baraldi, Zio, & Montelatici, 2019). The method also generates a notification when the model needs to be updated.	Classification&regression (Z. Yang et al., 2019).
Tiny machine learning for concept drift (TML-CD)	Active Passive	This algorithm can learn directly from IoT devices or embedded systems and continuously adapt to evolving data. The algorithm operates in supervised settings and supports active, passive, and hybrid adaptation strategies (Disabato & Roveri, 2022).	Classification and regression (Disabato & Roveri, 2022)

In the existing literature, both active and passive approaches have been utilised to address the problem of concept drift in predictive modelling. The reviewed methods in Table 5 show that, in most cases, the implemented methods are designed for classification problems with less support for regression problems. In addition, some of the methods are only suitable for particular types of drift. For example, PHT has been found to be more appropriate for abrupt drift scenarios (Gama et al., 2014); FCM supports both abrupt and gradual drift cases in classification and regression tasks (Dong et al., 2018); SPLL is suitable for abrupt drift cases with less efficiency for gradual drift scenarios supporting only classification tasks (Kuncheva, 2011); DDM supports abrupt and gradual drift cases, and it was extended as an upgraded version called EDDM to improve the efficiency for gradual drift scenarios. In addition, a recent active concept drift detection method called Meta-ADD was developed based on neural network concepts. The method is suitable for different types of concept drift without the additional need for hypothesis testing to ensure the correctness of the detected concept drift, and both classification and regression tasks are supported by the method (H. Yu et al., 2022).

On the other hand, passive approaches like OLIN and DACC can adapt to concept drift in the data stream but do not support regression problems (Cano & Krawczyk, 2020; Cohen et al., 2008). Whereas OS-ELM by Z. Yang et al. (2019) showed higher performance when detecting and adapting to different kinds of concept drift (abrupt, gradual, and recurrent) with support for classification and regression tasks. In addition, a hybrid method based on active and passive approaches called TML-CD was developed for concept drift adaptation from the data source directly in both classification and regression cases (Disabato & Roveri, 2022). Passive and active approaches provide different opportunities to ensure the performance of predictive models. However, passive approaches work well in recurrent and gradual concept drift cases where the learners are retrained based at a predefined frequency (Ditzler, Roveri, Alippi, & Polikar, 2015). However, adjusting the adaptation speed of methods based on passive approaches can be onerous. Furthermore, continuous retraining of the learners could be expensive in the context of constraints like memory and computational resources (Disabato & Roveri, 2022; Ditzler et al., 2015). In contrast, active approaches are considered more optimal for sudden drift cases and cases where a large amount of data is unlabelled (J. Lu et al., 2018; Mehmood, Kostakos, et al., 2021). In addition, unlike passive approaches, active approaches require the detection of concept drift before retraining (Disabato & Roveri, 2022), resulting in reduced model degradation over time. The detection of concept drift before retraining can be beneficial to investigate the root cause and ensure model accuracy (Kabir, Keung, Bennin, & Zhang, 2019). Passive approaches can be less effective when it comes to identifying concept drift. Both approaches are efficient and have been studied in the literature widely. They can be chosen based on requirements, the computational resources at hand, and other factors.

Much effort has been made to address concept drift in data. However, in most cases, the methods are tested against synthetic datasets, where the data may not frequently change compared to real-world scenarios. Therefore, more work is needed to experiment with these methods with real-world streaming data (Gulcan & Can, 2023; Iwashita, de Albuquerque, & Papa, 2019; Karimian & Beigy, 2023). Most of the proposed methods are focused on classification problems, requiring more exploration of regression problems (Lima, Neto, Silva Filho, & Fagundes, 2022). Moreover, the current literature lacks experimentation of concept drift detection and adaptation in real-world cases that incorporate large datasets and resemble smart application use cases (Mehmood, Kostakos, et al., 2021).

3 Research contributions

This chapter summarises contributions from the publications included in this thesis, which can be seen in Table 1. Section 3.1 investigates existing concept drift detection methods from the current state-of-the-art. In addition, it introduces new algorithms, considering the factors affecting the performance of models, such as time definition to retrain the model, adaptation strategies, and failures. Section 3.2 presents different platform architectures for data pipelines in smart cities for smart services and the development of predictive models with concept drift detection and adaptation methods in different real-world use cases. Finally, Section 3.3 describes the challenges for the concept drift application in real-world scenarios.

3.1 Concept drift detection methods for smart services in smart cities

The research question **RQ1** seeks different concept drift detection methods for efficient predictive models that can be applied in real-world scenarios. Publication **II & III** contributes towards answering this question. Generally, smart city ecosystems are built by leveraging modern technologies such as cyber-physical systems, cloud and edge infrastructures, sensors, AI, and other alternatives. Cities integrate different AI methods for predictive modelling to support the decision-making process and smart services development (Ashwini et al., 2022; Herath & Mittal, 2022). However, smart cities are dynamic environments, and data from smart cities continuously changes over time. Concept drift can be found in different areas of smart cities, for example, a city's forecast for power consumption demand may fluctuate with a sudden or gradual change in weather conditions. Additionally, extended winters can affect the mobility behaviour of citizens; enclosed market spaces may experience high sales and traffic congestion issues. Furthermore, algorithmic biases driven by demo-graphical and socio-economic variations at the municipality level can affect public decision-making. (J. Lu, Liu, Song, & Zhang, 2020; Mehmood, Hiltunen, et al., 2021; Morello et al., 2017). Therefore, addressing concept drift is essential for better smart services in smart city applications.

Exploration

To explore what kind of state-of-the-art solutions are available for concept drift detection and adaptation, we ran a number of benchmarks in Publication **II**. The methods included

Algorithm 1 Concept drift adaptation algorithm, reprinted under CC BY 4.0 license from Publication II.

Def: T: Training Data, $\mathbf{x}_{new}$: Streaming data, $\mathbf{y}_{new}$: Target variable, $\hat{\mathbf{y}}_{new}$: Predictions

Call learner ▷ *e.g., Prophet, TBATS, LSTM*
train learner with T ▷ *i.e., T as observations to date*
if Page Hinckley Test **then**
    for each new observation $\mathbf{x}_{new}$ **do**
        learner receives $\mathbf{x}_{new}$ and generates $\hat{\mathbf{y}}_{new}$
        compute the error for $(\hat{\mathbf{y}}_{new}, \mathbf{y}_{new})$
        if No drift is detected **then**
            save $(\mathbf{x}_{new}, \mathbf{y}_{new})$ ▷ *i.e., $\mathbf{x}_{new}$ added to T*
        else if drift is detected **then**
            retrain with T
            reset saved data ▷ i.e.,$\hat{\mathbf{y}}_{new}$
            Save the learner
        else
            Send the current state of learner
        end if
    end for
else
    for each new observation $\mathbf{x}_{new}$ **do**
        learner receives $\mathbf{x}_{new}$ and generates $\hat{\mathbf{y}}_{new}$
        compute the error for $(\hat{\mathbf{y}}_{new}, \mathbf{y}_{new})$
        if drift detector generates warning alert **then**
            save $(\mathbf{x}_{new}, \mathbf{y}_{new})$ ▷ *i.e., $\mathbf{x}_{new}$ added to T*
        else if drift is detected **then**
            retrain with T
            reset the saved data ▷ i.e.,$\hat{\mathbf{y}}_{new}$
            Save the learner
        else
            Send the current state of learner
        end if
    end for
end if

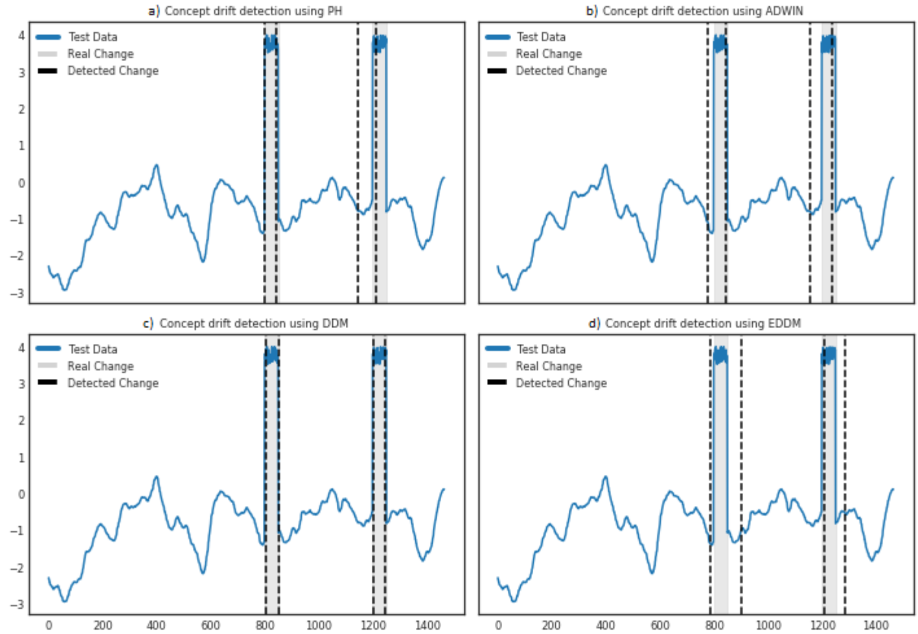

Fig. 4. Detected abrupt concept drift using synthetic dataset, reprinted under CC BY 4.0 license from Publication II.

PHT, ADWIN, DDM, and EDDM. The accuracy of these methods was tested against synthetic datasets with known abrupt and gradual drifting points shown in Fig. 4 and 5. The datasets were developed using the Gaussian process with Matern kernel 3/2 to evaluate abrupt drift cases and a Sinosoidal signal for gradual drift cases (Mehmood, Kostakos, et al., 2021). Later, we used the real-world dataset ELEC2 on scheduled electricity transmissions between different states in Australia (Mehmood, Kostakos, et al., 2021) for experiments. In addition, streaming data about power consumption in Finland from Fingrid (Fingrid, 2017) was used to benchmark the concept drift detection methods with real-time data streams. The concept drift detection methods were integrated with three different base learners: TBATS, fbprophet, and LSTM (Mehmood, Kostakos, et al., 2021), using a distributed computing paradigm-based data pipeline architecture described in Fig. 10 and described algorithm in Algorithm 1. The base learner was first trained with historical observations using data from ELEC2 (Mehmood, Kostakos, et al., 2021) and Fingrid (Fingrid, 2017); refer to line 1 & 2 in Algorithm 1. Then, each concept drift detection method was called to detect the underlying change in data distribution; if drift is detected, an alarm is generated by the method; refer to line 9 & 20-23 in Algorithm 1. Upon detection of concept drift, the learner is retrained; refer

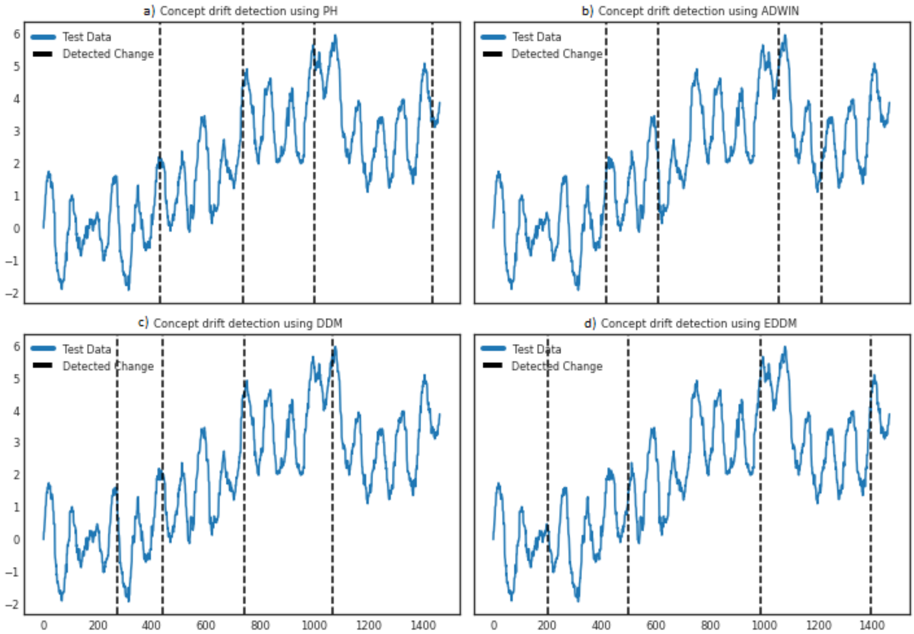

Fig. 5. Detected concept gradual drift using synthetic data, reprinted under CC BY 4.0 license from Publication II.

to lines 9-12 & 20-26 in Algorithm 1. If no concept drift is found, the learner's state is saved, and the process continues; refer to line 14 & 28 in Algorithm 1. More details can be seen in Publication **II** (Mehmood, Kostakos, et al., 2021).

Publication **III** proposes a novel feedback-based concept drift detection and adaptation algorithm built on top of edge micro data centre (EMDC) settings over a hybrid cloud-edge continuum, refer to Fig. 12 and 11. The proposed algorithm in Publication **III** addresses the limitations of the introduced algorithm in Publication **II**. The algorithm in Publication **II** uses a standalone LSTM implemented in a distributed data pipeline architecture. In contrast, Publication **III** uses a distributed LSTM based on a distributed learning paradigm for large-scale data streams. The implementation of the distributed LSTM is adapted from (Filonov, Lavrentyev, & Vorontsov, 2016; Mehmood, Kostakos, et al., 2021). The algorithm uses three workers and one master node inheriting the distributed learning capabilities of Apache Spark in the EMDC. The algorithm uses real-world streaming data about CO_2 from a sensor network installed in a smart campus (University of Oulu). The distributed LSTM is trained using historical data first stage and real-time data streams at a later stage. PHT, ADWIN, and Kolmogorov-smirnov windowing (KSWIN) are integrated with LSTM in the algorithm for concept drift

Algorithm 2 Feedback-based Concept drift detection and adaptation process, reprinted under CC BY 4.0 license from Publication III.

Historical data to train hd_T Streaming data ds_x, Target Variable ds_t, Predictions ds_o, Learner l_n, Worker Node W_n
Kafka topic for concept drift alert q_{t1}, Kafka topic for exceptions and messages q_{t2}, Kafka consumer for $\mathrm{q}_{t1} \rightarrow \mathrm{kc}_1$,
Kafka consumer to deploy model kc_2
Kafka producer for $\mathrm{q}_{t1} \rightarrow \mathrm{kp}_1$, Kafka producer to publish model kp_2
Initialize l_n
Train $\mathrm{l}_n \rightarrow hd_T$ with W_{n-1}
while $\mathrm{ds}_x = True$ **do**
    for each ds_x **do**
        Compute $\mathrm{ds}_o \leftarrow \mathrm{l}_n$
    end for
    if len($\mathrm{ds}_o > 0$) **then**
        Call Concept drift detector ▷ Kolmogorov Smirnov
        Compute Error (ds_t, ds_o)
        if CD is True **then**
            $\mathrm{q}_{t1} \leftarrow kp_1$
            Alert is sent
            Update model $\leftarrow kc_1$
            Send state $\leftarrow q_{t2}$
            $\mathrm{kp}_2 \leftarrow l_n$
            $\mathrm{hd}_t + ds_o = hd_t$
        else
            $\mathrm{kp}_2 \leftarrow \mathrm{l}_n$
            $\mathrm{hd}_t + ds_o = hd_t$
        end if
    else
        $\mathrm{q}t_2 \leftarrow kp_1$ (No values available)
    end if
end while

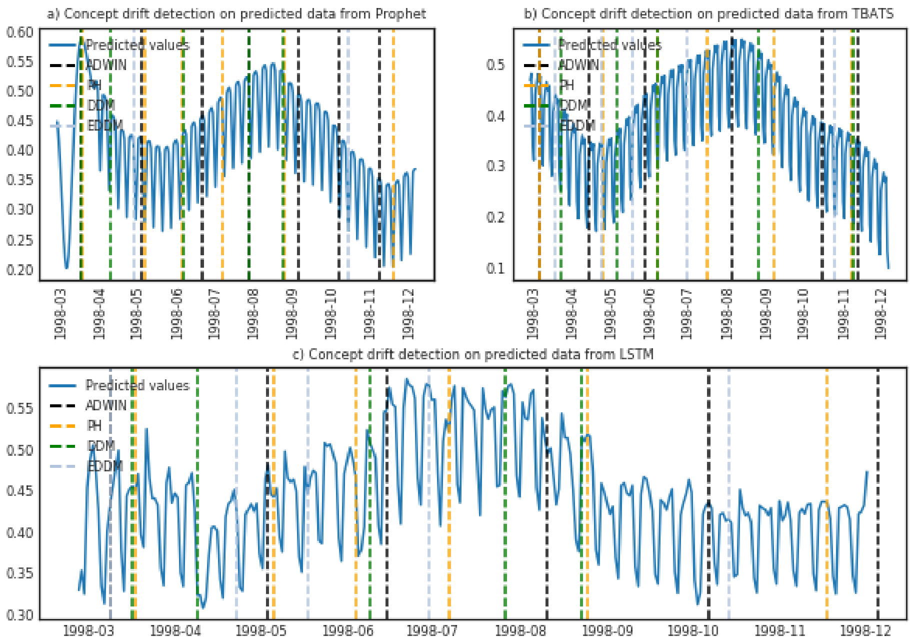

Fig. 6. Detected concept drift using data from ELEC2, reprinted under CC BY 4.0 license from Publication II.

detection. An alert is generated for defined Kafka topics (q_t1 & q_t2) if concept drift is detected or the data stream has no values, utilising Apache Kafka as a communication bus; refer to line 11, 17 in Algorithm 2. The Kafka topics are assigned to the Kafka producer kp_1, which generates the alert " concept drift is detected" and waits for the defined Kafka consumer kc_1 to receive the broadcast message; refer to line 11-17 in Algorithm 2. Later, the Kafka consumer kc_1 consumes the message to retrain the model and saves the current state of the learners; refer to line 16, 17 in Algorithm 2. Additionally, if the data stream contains no data, the Kafka producer kp_1 broadcasts a message, and the process is restarted to consume the data streams again; refer to line 25 in Algorithm 2. If no concept drift is detected, the model is saved directly for deployment by the Kafka producer kp_2, and the incoming data streams are evaluated for concept drift detection; refer to line 21, 22 in Algorithm 2. The complete implementation is described in Algorithm 2 and Fig. 8.

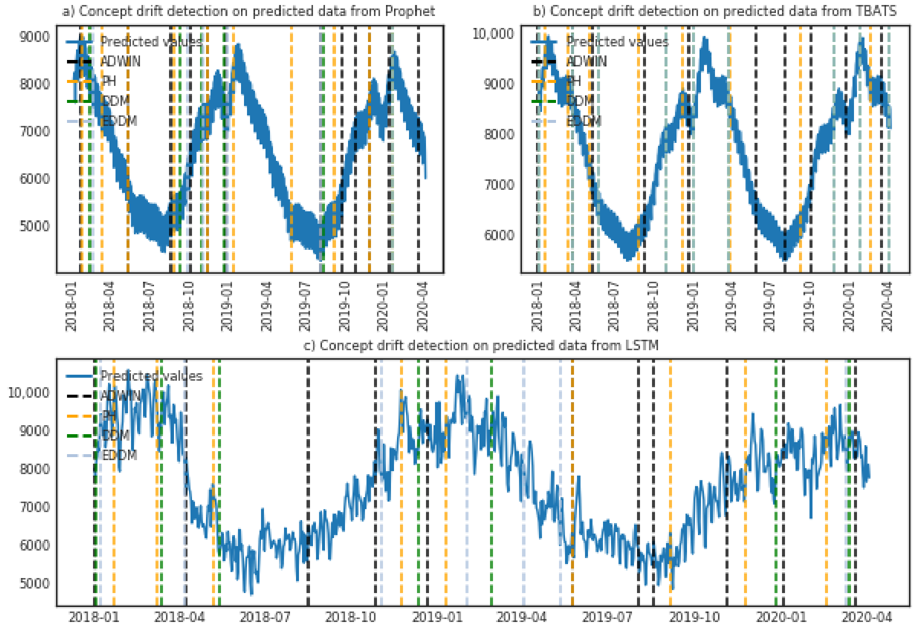

Fig. 7. Detected concept drift using data from Fingrid, reprinted under CC BY 4.0 license from Publication II.

Evaluation

The implemented algorithm in Publication **II** was evaluated using a state-of-the-art evaluation method called mean absolute percentage error (MAPE) = $\frac{100}{n}\sum_{t=1}^{n}\frac{|x_t - y_t|}{x_t}\%$, where x_t corresponds to the original observations and y_t corresponds to the predicted values, as seen in Table 6, among the combination of methods used for concept drift detection and adaptation integrated with time series base learner. LSTM and PHT significantly lower the MAPE for both the ELEC2 (0.97%) and Fingrid (0.24%) datasets. In addition, the predicted values from LSTM integrated with PHT could comply and follow the original observations comparatively to their standalone base learners. Fig. 6 and 7 show the detected concept drift integrated with a time series and base learner from dataset ELEC2 and Fingrid.

In Publication **III**, a distributed time series learner (LSTM) was integrated with a number of concept drift detection methods like PHT, ADWIN, and KSWIN. The evaluation of the algorithm employed on the edge platform was conducted using methods such as MAE= $(\frac{1}{n})\sum_{i=1}^{n}|x_t - y_t|$, RMSE= $\sqrt{(\frac{1}{n})\sum_{i=1}^{n}(x_t - y_t)^2}$, and MAPE. It can be observed from Table 7, that the employment of the concept drift detection method

Table 6. **Computed mean absolute percentage error (MAPE) for time series learners, reprinted under CC BY 4.0 license from Publication II.**

	Method	ELEC2	Fingrid
	standalone LSTM	5.10	0.57
	with PHT	0.97	0.24
LSTM	with ADWIN	5.75	0.45
	with DDM	2.28	0.48
	with EDDM	3.51	0.83
	standalone Prophet	15.58	6.1
	with PHT	9.86	5.68
Prophet	with ADWIN	20.66	6.48
	with DDM	10.06	6.80
	with EDDM	9.94	6.92
	standalone TBATS	21.35	7.0
	with PHT	17.25	6.45
TBATS	with ADWIN	17.37	7.05
	with DDM	16.94	7.08
	with EDDM	23.81	7.11

Table 7. **Evaluation of time series learner with integrated concept drift detection methods, where min observation is 341.9 and max is 1283.0, reprinted under CC BY 4.0 license from Publication III.**

Method	MAE	MAPE	RMSE
LSTM	42.8	8.5	59.2
LSTM with PHT	24.02	4.8	33.56
LSTM with ADWIN	23.91	4.78	32.22
LSTM with KSWIN	19.8	3.88	27.9

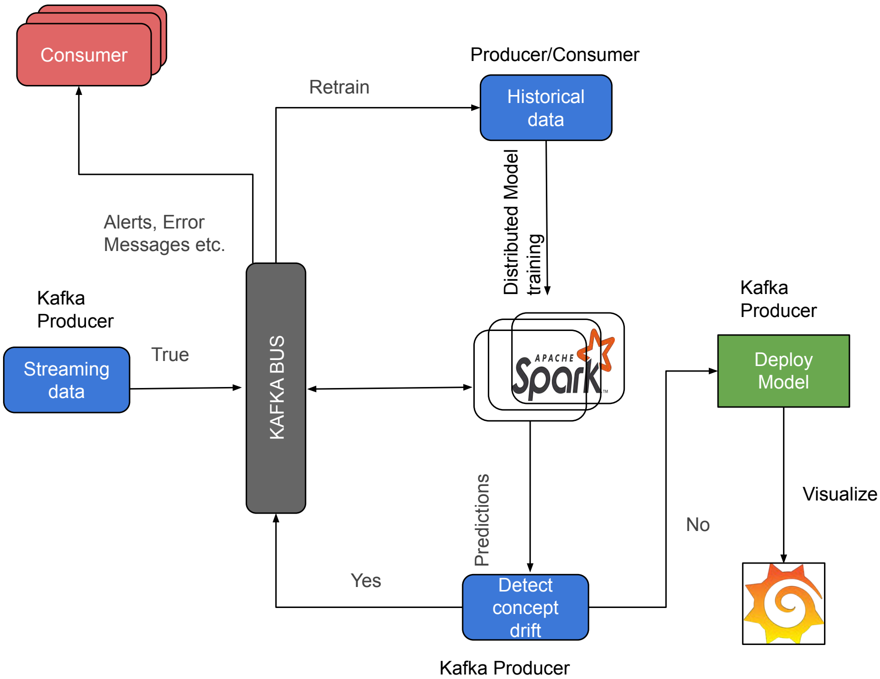

Fig. 8. Execution flow of algorithm, reprinted under CC BY 4.0 license from Publication III.

helped to improving MAPE with KSWIN from 8.5% to 3.88% when compared to a standalone distributed LSTM and its integration with PHT and ADWIN. In addition, the detected concept drifts in the predicted values are shown in Fig. 9.

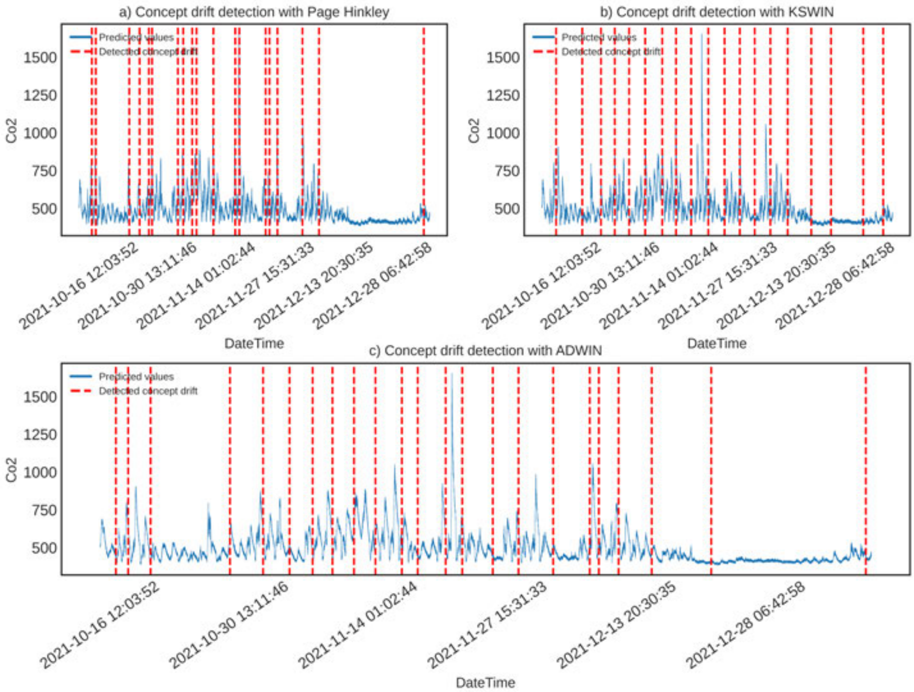

Fig. 9. Comparison of detect concept drift with different methods integrated with distributed time series learner, reprinted under CC BY 4.0 license from Publication III.

3.2 Architectures for data pipelines supporting automated concept drift detection and adaptation

The research question **RQ2** is focused on distributed computing paradigm-based architectures for data platforms supporting automated concept drift detection and adaptation for smart services and decision-making in smart cities. Publications **I, II, and III** propose cloud and edge computing paradigm-based distributed architectures for data collection, storage, analysis, visualisation, monitoring, ensuring ground truth of data, and others. In smart cities, trends for automating processes in modern cities are becoming evident daily to monitor water levels, air quality, energy demands, and street light management, and data collection from other sources to develop applications and services for citizens (Cheng et al., 2015; Cicirelli et al., 2017; *CUTLER*, 2018; Herath & Mittal, 2022; Sanchez et al., 2014). Many efforts have been made to develop efficient smart city platforms to address existing challenges and provide smart services in smart cities (see Table 4). However, much less work is available that provides a comprehensive solution for smart cities capable of handling the highlighted challenges.

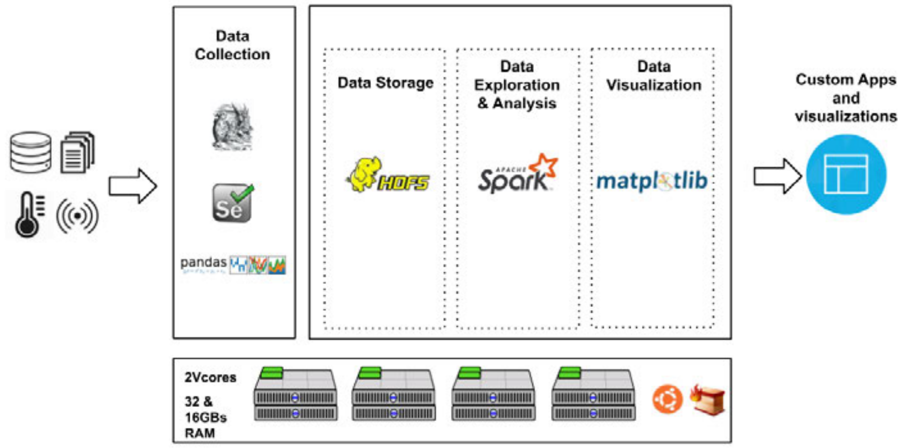

Fig. 10. Distributed architecture for data pipeline implementation, reprinted, with permission, from Publication I © 2019 IEEE.

Publications **I & II** contribute towards the **RQ 2.** by implementing a cloud-based distributed data pipeline architecture described in Fig. 10. The developed distributed data pipeline architecture addresses the highlighted challenges in the current state-of-the-art for data collection, storage, analysis, visualisation, and management for large-scale data in smart cities. In the data platform implementation, different constraints were taken into account. For example, data in smart cities can be stationary, non-stationary, structured, unstructured, and semi-structured; therefore, the processing of such data must be supported. In addition, in Publication **I**, it has been highlighted that the data platform should be highly available, reliable, fault-tolerant, scalable, and needs to support both batch and real-time processing and enable metadata management. To address these requirements, different technology solutions and big data tools were configured to develop the platform, such as selenium, pandas, beautifulsoup4, HDFS, Apache Spark, matplotlib, as well as support for custom applications through application interfaces. (Mehmood et al., 2019; Mehmood, Kostakos, et al., 2021). To facilitate data collection from heterogeneous sources, a combination of standalone frameworks such as pandas, beautfulsoup4, along with a distributed data collection framework called Apache Flume, were used; refer to Fig. 10. In addition, for data analysis, a distributed computing paradigm-based framework called Apache Spark was configured to provide batch and real-time data analysis. Later, for visualisation purposes, the standalone python framework matlplotlib is configured. The architecture implementation was first tested with real-world streaming data from Cork, Ireland, to perform sentiment analysis using a machine learning classification approach. The architecture was further

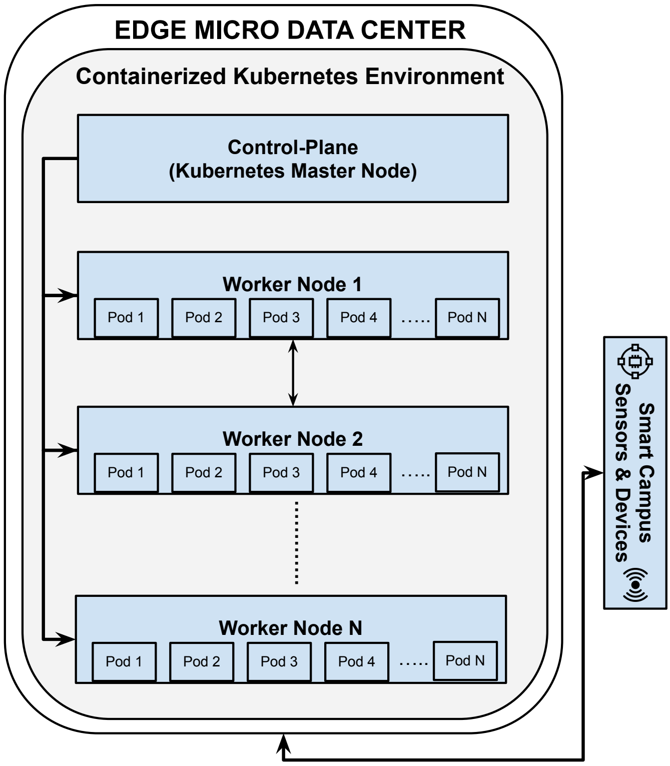

Fig. 11. Containerized edge platform for smart environments, reprinted under CC BY 4.0 license from Publication III.

evaluated by computing the training time for the model to be 51.1 seconds (Mehmood et al., 2019). In Publication **II**, the work was extended by implementing an automated concept drift detection approach for large-scale streaming data, described in Section 3.1. During the implementation, it was observed that concept drift detection in the distributed environment posed several challenges. For example, if the data is not distributed evenly, it may result in task failures due to data skewness and balancing issues. In addition, clusters often have asynchronous communication possibilities; in such cases, policies should be set carefully (e.g. a subset of the data stream should be read, or the entire data stream should be read) if data is not fully read or there are network failures.

Publication **III** is focused on providing a comprehensive and containerised edge data platform architecture for smart cities and environments. The deployment of 5G and IoT has enabled the development of smart city services and applications, insights generation, and decision-making processes which are closer to the data sources and end-user. In

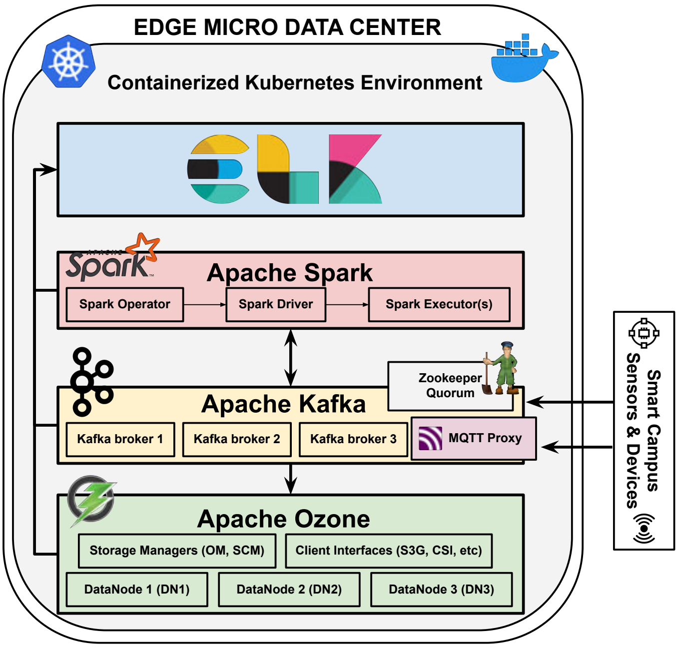

Fig. 12. Implemented component and services in edge platform, reprinted under CC BY 4.0 license from Publication III.

this regard, using the edge computing paradigm to develop data platforms closer to the users mitigates the challenges of cloud computing, such as high latency, mobility, reduced network overhead, and the development of low-latency applications (e.g. traffic applications and autonomous vehicles).

The containerised edge platform architecture is implemented for computationally demanding data pipelines in EMDC settings over a hybrid cloud-edge continuum. The implementation follows a containerised approach to meet the requirements of reliability, scalability, and diversity; refer to Fig. 12. Unlike in traditional environments, virtual machines distribute resources between configured components and resources, resulting in overhead to manage the virtual machines, reduced resource distribution flexibility and slowed processes and machine spawning. The implemented solution is portable, allows for computationally demanding analyses, and supports integrating and developing different smart city scenarios for smart services, applications, and decision-making. Furthermore, on-demand live migration of workloads (data processing jobs) is facilitated for resource allocation based on requirements. The proposed solution includes a telemetry layer for continuous monitoring and enables analytics-driven resource optimisation for workloads.

The designed architecture is based on Kubernetes, consisting of a controller node acting as a management node to handle the schedule, resource allocation, and other aspects. Worker nodes are used to execute the tasks. Furthermore, all configurations and environment variables are stored as Kubernetes objects, making them available at run time for other applications. In this regard, the application is more portable due to the decoupled configuration approach (Mehmood, Khalid, Kostakos, Gilman, & Pirttikangas, 2024).

Implementing a smart city platform requires different components to collect data from heterogeneous sources, services for data storage, process management, data analysis, and results delivery. For example, the implementation of automated concept drift detection on a smart campus described in Section 3.2, uses data from different sensors in real-time about environmental readings, noise levels, and from motion detectors. The proposed platform is suitable for such use cases in smart cities. The proposed design is implemented with four main components with scale-up options:

Storage System: Each worker node in EMDC needs persistent storage for the execution of stateful applications. A storage system capable of abstracting storage space, operating from applications, and supporting consumption from Kubernetes API is needed to enable dynamic and transparent resource allocation for applications and workloads. These requirements are met using Apache Ozone as a storage system. Apache Ozone addresses the challenges in Apache Hadoop, such as scalability issues, immutability, performance issues with small files, single namespaces, and integration with modern applications. Different components of Apache Ozone were deployed in the platform, such as storage container manager, ozone manager, container storage interface (CSI), and S3 gateway. CSI acts as an interface between external storage and container workloads, supporting the creation and configuration of persistent storage for application on an external storage system (Ozone). This simplification allows simple consumption of storage resources. In addition, if an application requests a storage volume, Ozone seamlessly creates a mount. The fault tolerance in Ozone is ensured through its self-healing properties in case of node failures, making the data highly available. Finally, integration with external applications and other clients is assured through the S3 gateway.

Messaging bus: The data in smart cities originates from diverse sources in varying frequency, requiring the process of data ingestion in a fault-tolerant and reliable fashion. In addition, modern AI predictive models require low latency and concurrent solutions to manage and process large-scale streaming data. Therefore, Apache Kafka is configured to support the process of data collection, building reliable data pipelines supporting predictive modelling, automated concept drift detection, and adaptation for missioncritical applications. Three Kafka brokers were deployed in the EMDC platform with two listeners. One listener handles the services and configuration inside the EMDC, and the other for consumers deployed outside the EMDC. The Zookeeper service is configured for Kafka functions within Kubernetes pods. In addition, sensors usually use the MQTT protocol for sending messages and readings from sensors due to its scalability features and lightweight interface. Therefore, an MQTT proxy is configured to interface the produced messages from sensor networks directly to Kafka.

Data analytics engine: Various frameworks support computationally demanding data analysis for large-scale data streams, such as Apache Spark, Apache Storm, and Apache Flink. Any of these frameworks can be configured seamlessly in EMDC depending on the requirements. In this setup, Apache Spark is used due to its support for distributed computing, efficient deployment of models, scalable processing, and support for library customisation. A Spark operator configures Apache Spark on the platform enabling the deployment of Spark jobs in a declarative manner through a Kubernetes manifest. The main application and additional dependencies are configured in the Spark Docker image. Furthermore, other configurations, management, and executions are maintained by the Spark operator and Kubernetes itself, making the overall implementation more straightforward for AI model development. When the user creates a Spark job using Kubernetes pods, the Spark operator automatically creates a spark driver and executors to complete the job.

Telemetry and monitoring: The implemented EMDC platform supports the integration of logging and visualisation frameworks, such as Elastic search, logstash, Kibana, Prometheus, Grafana, and influxdb. These frameworks can be leveraged for results delivery, insight generation, and dashboard developments in real-time. By default, on the EMDC platform Kubernetes supports some of the mentioned frameworks, making the configurations straightforward. In addition, the telemetry and monitoring layer can also support logging system-level events and developing insights into failures, security breaches, alerts, tests and others.

The feasibility of the proposed architecture is verified with the previously mentioned smart campus use case, where different sensors are configured to measure environmental parameters, noise levels, motion sensors, and lighting sensors. In such environments, multiple EMDCs can be installed in different locations in the vicinity due to their small form factor. In this regard, such an approach may help with low-latency issues, unlike in cloud-based infrastructures, online learning, continuous data delivery and monitoring, and data variability. The deployed sensors' data was collected and analysed to forecast the CO_2 footprint at different locations in the smart campus. In addition, the deployed predictive model is integrated with concept drift detection methods and

```
,breaklines
ubuntu@OULU----:~$ kubectl describe pod workload.spark
Name:         workload.spark
Namespace:    default
Node:         oulu2-work01/10.13.2.2
Annotations:  kubectl.kubernetes.io/last-applied-configuration:
                {"apiVersion":"v1","kind":"Pod","metadata"
                :{"annotations":{},"name":"workload.spark"
                ,"namespace":"default"},"spec":{"contai...
Status:       Running
IP:           10.244.1.14
IPs:
  IP:  10.244.1.14
Containers:
  workload:
    Container ID:   docker://e279b19b4fa1216ac98af6c2ea0b8510
                    e2e535f3126de4e7c24f6edc749935d0
    Image:          10.13.2.1:5000/workload/spark
    Image ID:       docker-pullable://10.13.2.1:5000/workload/
                    spark@sha256:dc70e
                    ba897ca8d311fba765bfd1a7e8eedf7f42d42a4502
                    2e310e1ccaaed1d0
    Mounts:         /var/run/secrets/kubernetes.io/serviceaccount
                    from default-token-xkhkj (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-xkhkj:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-xkhkj
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s

Events:
  Type    Reason     Age        From                   Message
  ----    ------     ----       ----                   -------
  Normal  Scheduled  <unknown>                         Successfully assigned
                                                       default/workload.spark
                                                       to oulu2-work01
  Normal  Pulling    3m4s       kubelet, oulu2-work01  Pulling image
                                                       "10.13.2.1:5000
                                                       /workload/spark"
  Normal  Pulled     2m18s      kubelet, oulu2-work01  Successfully pulled
                                                       image "10.13.2.1:5000
                                                       /workload/spark" in 46.473354841s
  Normal  Created    2m6s       kubelet, oulu2-work01  Created container
                                                       workload
  Normal  Started    2m6s       kubelet, oulu2-work01  Started container
                                                       workload
```

Fig. 13. Event logs from executed Spark job in containerised EMDC, reprinted under CC BY 4.0 license from Publication III.

adaptation strategies to ensure efficient forecasts. As in real-world cases, data may drift due to calibration issues or malfunctioning devices, resulting in data quality issues and ineffective predictions when supplied to AI learners. The implemented algorithm (workload) is executed as a container in the EMDC due to efficient resource allocation and portability. The complete implementation is described in Section 3.1. In addition, Kubernetes-based orchestration makes the overall system agile and easily manageable, as the operational functions such as fault-tolerance, scalability, and load balancing are independent of the main application and instead are handled by the orchestrator. Fig. 13 shows the logs from the submitted workload to the EMDC platform, where steps starting from image creation, event details and the state of the jobs are mentioned.

3.3 Challenges for real-world applicability of concept drift detection

The research question **RQ3** discusses the limitations and constraints for the real-world applicability of concept drift detection methods. The real-world systems performing data-driven predictions, forecasts, decisions, and others require additional procedures to detect and adapt to concept drift in data distribution. Publications **II-V** are focused on answering this question through a state-of-existing knowledge base, lessons from experimentation and implemented use cases in the work.

Data-driven decision-making platforms and applications are usually built on AI-based predictive models. In non-stationary environments, the data continuously changes with variable conditions, for example, changes in environmental parameters, changes in fuel average and driving patterns in vehicles), traffic situations, and many others. Therefore, real-world application areas require integrating concept drift detection approaches with AI approaches for effective and efficient results. In Publication **IV**, an augmented reality (AR) application called DistAR is proposed to assist visually

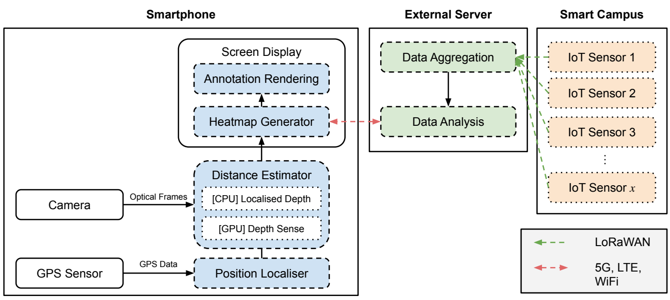

Fig. 14. Architecture for DistAR with AR and smart sensing analysis, reprinted under CC BY 4.0 license from Publication IV.

with social distancing in the post-COVID-19 situation. The application is developed by integrating a sensing network from the Smart Campus, at the University of Oulu. Various sensors are configured on the campus, continuously generating measurements related to environmental parameters, noise, illumination, and motion of individuals in the facility. DistAR leverages AR and smart sensing technology to support users in making safe navigation and social distancing choices to avoid possible exposure to airborne virus particles. Furthermore, depth estimations on user devices and data from passive infrared (PIR) inform users of the crowdedness in different areas of the campus. The complete implementation architecture is described in Fig. 14. When the user opens the application, a world-facing camera view is displayed. During the COVID-19 pandemic, a 2-meter safety distance was proposed globally, the application blurs the region outside a distance of 2 meters. In addition, the central reticle estimates the depth, and if the reticle moves to a larger region than 2m, haptic motors on the phone alert the user for social distancing. In addition, near real-time measurements from PIR sensors from the smart campus are embedded in the application's main display to provide an overview of populated regions. The complete application view can be seen in Fig. 17. The current implementation uses a custom drift detection mechanism to find erroneous values in the data stream from the smart campus that can arise due to calibration issues, changes in mobility patterns, and faulty devices. Considering the application's intensive nature, accurate sensor measurements are needed to provide insights and integrate predictive models with concept drift detection methods to forecast future crowdedness on the campus.

Publication **V** provides a hybrid data processing architecture based on edge and cloud computing paradigms for developing autonomous excavators. The excavators are popular in civil engineering, the forest industry, mining, space exploration, and disaster management. Machine control systems have widely been utilised to automate excavators for routine operations, especially in hazardous conditions where human involvement is limited. However, modern excavators have become more perceptive due to the integration of sensors, 3D cameras, lidar, laser beams and others. Therefore, data-driven solutions are needed to enable the excavation in variable weather conditions, adaptive movement and control, object detection, optimal path generation etc. The paper reviews the state-of-the-art and proposes a hybrid edge-cloud architecture and framework for AI-driven smart excavators, described in Fig. 16 and 15. Three main components are proposed in Fig. 15; i) the sensing and planning component is allocated for sensing and modelling in real-time with edge and cloud-based infrastructure, ii) the control and operation component facilitates autonomous and manual control for excavator, iii) communication and integration configure the sensors, enable wireless

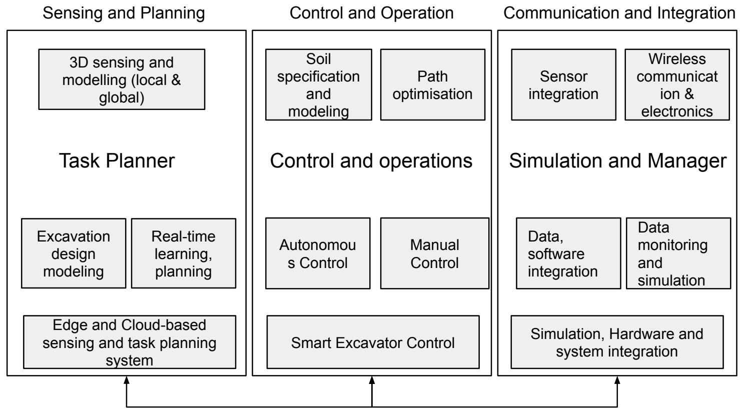

Fig. 15. Main components for AI-driven excavator, reprinted, with permission, from Publication V © 2021 Authors.

communication, and provide a monitoring and simulation environment. The hybrid edge and cloud architecture supports these components to perform analytics, modelling, and monitoring (see Fig. 16). The data generated from the sensors, such as 3D lidar, laser, cameras, and others equipped with the excavator, are leveraged to develop optimal paths, access dynamic soil characteristics and object detection. In this regard, due to variable environmental conditions, calibration errors, and malfunctioning devices, the learned data patterns may become obsolete due to possible concept drift in the data distribution. Similarly, in object detection on remote construction sites, the changing soil dynamics in the vicinity and crucial weather and storms can also result in drifting situations. Therefore, in non-stationary environments, ensuring the ground truth of the data is necessary for effective predictions and forecasts. The proposed framework supports the integration of predictive learners and concept drift detection methods.

The earlier presented real-world examples are appropriate where concept drift detection and adaptation approaches can be further applied. However, real-world implementation has challenges, as mentioned in work described in Publications **II-III**. For example, real-world streaming data about power consumption in Finland was used to perform forecasting, and was comprised of a base learner integrated with concept drift detection and an adaptation strategy (see 3.1). The data in smart cities is increasingly becoming heterogeneous and demands more computational power and fault-tolerant data distribution-based approaches for data fusion, predictive analysis, and concept drift detection. In a distributed environment, implementing concept drift detection and adaptation becomes challenging if the data across the nodes are not balanced. The experiment also showed that if a data record is skipped during computations

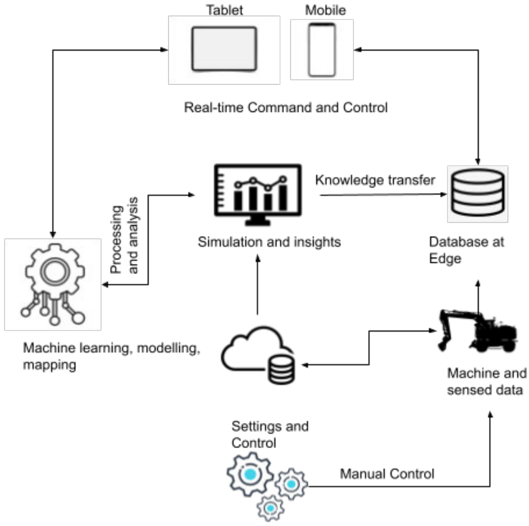

Fig. 16. Edge architecture for a smart excavator, reprinted, with permission, from Publication V © 2021 Authors.

or not fully read, it can truncate the available memory and result in task failure. In addition, employing predictive models in the real world requires continuous feedback for monitoring. Most modern distributed frameworks support asynchronous communication, but concrete assessment is needed before setting any rules, for example, in the case of a network failure, data should be restored partially or fully.

Similarly, concept drift detection in real-world settings often requires prior knowledge of data patterns with the envisioned changes. Such information is substantial for fine-tuning the learners and concept drift detectors. However, heuristic-based parameter selection can be made based on variations in streaming data. Data containing noise or complex patterns can obstruct the fine-tuning of the learner. In addition, imperfect settings of concept drift detection methods result in false alarms, especially in noisy data streams. Based on a defined strategy, base learners are usually retrained upon concept drift detection. However, adaptation strategies can be compromised if true and false alarms are not distinguished. In addition, the retraining frequency of models upon the detection of concept drift should be evaluated given the computational effort and selection subset or entire retraining data.

In Publication **III**, a feedback-based concept drift detection and adaptation algorithm is proposed to handle concept drift in real-world data streams. The implementation focuses on a real-world use case to forecast carbon emissions in a smart campus using sensor network data. Based on the pre-COVID-19 and post-COVID-19 hypotheses,

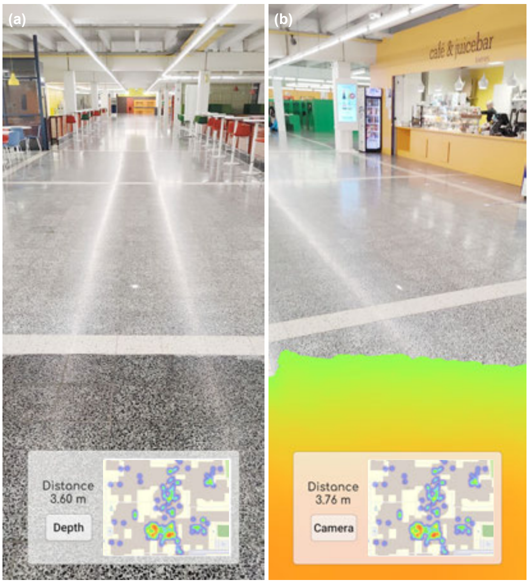

Fig. 17. Application example view, reprinted under CC BY 4.0 license from Publication IV.

changes in mobility patterns during lockdown would result in concept drift in the predictions. Results showed the detection of concept drifts in the COVID-19 time frame. However, in one case, the concept drift arose due to malfunctioning sensors, exhibiting a value of 6000ppm, i.e. larger than the ideal CO_2 value range of 400ppm to 800 ppm. In such cases, data and environmental details should be known a priori to address the root cause. Furthermore, fine-tuning of parameters and periodic manual intervention may be needed in this case. On the other hand, passive approaches do not require the direct detection of concept drift. Passive approaches utilise ensemble-based models in most cases with pre-defined retraining frequencies. However, such approaches can be computationally exhaustive due to unnecessary periodic retraining. In contrast, active approaches comprising base learners and concept drift detection methods are well-suited for real-world cases due to the dynamic nature of data, and opportunities to study why and how concept drift is detected.

4 Discussion

This chapter discusses and revisits the research aims and questions outlined in the first chapter of this thesis. Furthermore, the corresponding answers to each research question are briefly addressed. Finally, the Chapter is concluded with limitations and the future work of this thesis.

4.1 Revisiting research questions

RQ1 - How do AI approaches support concept drift detection for smart services in smart cities?

The research identified different concept drift detection approaches supported by AI smart services in smart cities. The smart city ecosystem driven by IoT, sensor networks and other ubiquitous technology has provided tremendous opportunities for smart services and applications for citizens and data-driven decision-making to governance bodies. Data and machine learning methods drive the analytics tasks in most smart city applications for forecasting and decision-making, posing a risk of ineffective predictive models due to variable changes in data ground truth. Therefore, when predictive modelling is performed, the supplied data to the models must be continuously monitored due to concept drift. During the investigation, it was observed that most concept drift detection methods were tested using synthetic data sets (Gulcan & Can, 2023; Iwashita et al., 2019; Karimian & Beigy, 2023; Mehmood, Kostakos, et al., 2021). In contrast, the large amount of produced data in smart cities is time series oriented. Various proposed concept drift detection methods were also studied for classification problems, requiring exploration for regression problems (Lima et al., 2022).

Publication **II** implemented and benchmarked four concept drift detection methods using synthetic and real-world datasets from the energy sector. Three time series learners (LSTM, Prophet, TBATS) were integrated with four concept drift detection methods, i.e. PHT, ADWIN, DDM, and EDDM. The concept drift detection methods were first evaluated for performance using synthetic datasets with known drift points. Later, the time series learners integrated with concept drift detection methods were supplied with real-world data about power consumption from Finland. The results from the proposed algorithm described in Section 3.1 showed that integration of concept drift detection methods with predictive models could enhance the predictive performance of the models. It was observed that ADWIN and PHT methods, when integrated with learners, improved the learners' computed MAPE compared to other methods (see Table 6).

Similarly, Publication **III** acknowledged that integrating concept drift detection approaches could enhance learners' efficiency. The study integrated distributed LSTM with three concept drift detection approaches (PHT, ADWIN, KSWIN) to detect and perform learners' adaptation. Section 3.1 explains the complete algorithm and implementation. The proposed algorithm was implemented on top of a novel edge platform with real-world data from a sensor network in a smart campus. It was hypothesised that CO_2 forecasting would be affected by pre and post Covid situations due to changes in mobility patterns of individuals on a smart campus, eventually resulting in ineffective predictions. Therefore, the training process of learners included data from the pre Covid situations, and testing was performed on post Covid data. The integrated concept drift detection methods with distributed learners detected concept drift in the generated predictions. However, Table 7 shows a significant decrease from %8 to %3.88 in the computed MAPE when the distributed LSTM with KSWIN, compared with other methods.

The experimentation shows that integrating concept drift detection with base learners and adaptation processes improves the performance of models. However, effort is needed to fine-tune the model parameters and define user-defined thresholds for concept drift detection methods. In addition, upon detection of concept drift, the adaptation or retraining of the model is a complicated process because of different constraints. For example, upon retraining, which segment of the data needs to be corrected, how much data do the models need to be retrained for, and how often should the parameters of learners and concept drift detectors be altered? The real-world applicability of concept drift detection and adaptation approaches, limitations, and other constraints are discussed in Section 4.1.

RQ2 - What architecture is required to implement data pipelines supporting automated concept drift detection and adaptation?

Many efforts have been made to develop smart city platforms to address the requirements for different smart city services and applications (see Table 4). However, most of the smart city platforms are tailored to specific use cases, requiring more work for a comprehensive solution that incorporates different requirements altogether (Luo et al., 2019; B. N. Silva, Khan, & Han, 2018). The **RQ2** focuses on addressing the challenges mentioned earlier and contributes to different distributed data pipeline architectures to support the process of data collection, storage, analysis, visualisation, results delivery, data quality, data management, concept drift detection, and predictive modelling.

Publications **I, II, and III** contribute different distributed data pipeline architectures to address the earlier-mentioned challenges in smart cities. The proposed platform in Publication **I & II** leverages the cloud computing paradigm and big data technology solution to develop distributed data pipeline platform supporting data collection, storage, analysis, results delivery, monitoring, and others (see Section 3.1). It was observed from the state-of-the-art experimentation that real-world data platforms require flexible, fault-tolerant, reliable, available, and scalable solutions to handle large-scale and streaming data (Corchado et al., 2021; Syed et al., 2021). In addition, the data-driven decision-making processes depend on predictive models and insights generated from data in smart cities. In such cases, the integrated tools and frameworks should support seamless integration and healing properties in case of network failures. In addition, most concept drift detection methods require feedback, resampling, maintenance etc. Therefore, integrated approaches and frameworks to carry out forecasting and predictions should support the integration of such methods. This work utilised Apache Spark's capabilities to develop predictive models integrated with concept drift detection methods.

In Publication **III**, a novel containerised edge computing paradigm-based EMDC was proposed to address computationally demanding processing and analyses. Considering the challenges of data collection, management and processing huge amounts of data at the edge, more solutions are needed. Table 4 shows that in most cases, current edge computing paradigm-based solutions lack support for handling computationally demanding processing at the network's edge. In addition, some edge computing paradigm-based platforms handle resource-demanding jobs off-network or leverage cloud middleware (Cicirelli et al., 2017; Hossain et al., 2018). The platforms built using virtual machines, e.g., (*CUTLER*, 2018; Pereira et al., 2022; Vítor et al., 2022), do not support the migration of systems components and applications. A notable issue with rapidly growing data also results in degraded performance issues with data processing platforms. As mentioned, handling concept drift in predictive modelling is crucial for decision-making processing apart from data collection, storage, management, and processing issues. A recent study by Xu et al. (2023) proposed a generalised architecture following a containerised approach, however, the platform showed limited capabilities for system logging and analytics-driven resource allocation.

The implemented solution in this study address the mentioned limitations in the state-of-the-art (see further in Section 3.2). The proposed platform supports on-demand live migration of resource-intensive jobs following a containerised architecture that facilitates the integration of different smart cities scenarios and workflows, concept drift detection and adaptation, and analytics-driven resource allocation through the telemetry layer. We used a real-world use case to test the implemented platform with a distributed LSTM with concept drift detection methods such as PHT, KSWIN, and ADWIN. It was noticed that integrating concept drift detection methods in distributed edge computing environments can be complex, for example data arriving from different edge nodes for processing may need balancing and partitioning, requiring additional processing. In addition, a workload during data collection or concept drift detection may fail or execute partially due to code errors, network issues, stale processes, and empty data streams. Such matters must be handled on production-grade platforms. To manage such losses and delays, we integrated Apache Kafka, which follows an event-driven approach. To summarise, the performed work addresses most of the limitations found for existing smart city platforms, contributes to the state-of-the-art, and provides opportunities for future exploration. The limitations and open issues in the implementation are discussed in the next Chapter 5

RQ3 - What are the challenges for concept drift applications in real-world scenarios?

With rapid digitisation and growing trends of AI, concept drift has become a crucial issue. The prominence of concept drift can be seen in different areas such as automobiles, power grid systems, environmental monitoring, transportation, and others (Manias, Shaer, Yang, & Shami, 2021; Mehmood et al., 2024; Mehmood, Kostakos, et al., 2021; Peña, Lanzarini, Cerrada, Cabrera, & Sánchez, 2021; Zhao, Xu, Zang, & Hu, 2021; Žliobaitė, Pechenizkiy, & Gama, 2016). However, applying concept drift detection and adaption is an effortful task, due to its different types.

Publications **IV & V** provide different use cases from smart construction and smart building sectors that are described in Section 3.3. In the first use case, an AR application using the Android platform was developed to anticipate the degree of crowdedness in a smart campus using features from handheld mobile devices and sensing networks installed in the building. Whereas the second use case from Publication **V** proposes a framework and edge architecture for smart excavators following recent advancements in the construction engineering area. Modern applications and machines have become more perceptive with integrated sensors, lasers, cameras, IMU, and many other features. Therefore, AI methods are being abundantly used for object detection in low light and dusty environments, movement of soil dumps to other locations, trajectory planning, and many others Fernando and Marshall (2020); Shariati, Yeraliyev, Terai, Tafazoli, and Ramezani (2019); Son, Kim, Kim, and Lee (2020). In both use cases, forecasting crowdedness on the campus or performing object detection and path planning on a construction site can be faced with concept drift issues in the data. For example, forecasting crowdedness due to malfunctioning devices or sudden changes in environmental parameters can result in ineffective predictions, therefore, leading to ineffective decision-making and degraded models and performance. Similarly, rapid changes in soil properties, dusty conditions, sensor calibration issues, and changing climate conditions can provide less accurate object detection and path planning results.

The real-world applicability of concept drift detection and adaptation brings many challenges, such as support for detecting gradual, sudden, incremental, and recurrent concept drift. Moreover, tuning parameters for concept drift detection methods are challenging and often require prior knowledge about data in real-world cases. Publications **II & III** discussed such challenges by applying concept drift detection and adaptation processes to real-world use cases (see Section 3.1). In most cases, the detection methods for concept drift can handle only a few types of concept drift (Mehmood, Kostakos, et al., 2021), which can be a potential challenge in real-world applications. The data in the real world is continuously changing and may contain one or more types of concept drift in the data, e.g. sudden or abrupt drift, recurrent drift, gradual drift etc.

Concept drift detection is usually done using active concept drift detection methods that require the definition of parameters, tuning of parameters, and integration with base learners. In several cases, passive approaches employ ensemble-based learners, which periodically retrain the learners without explicit concept drift detection. Therefore, choosing the method itself is challenging, but if concept drift needs to be studied and mitigation mechanisms are to be placed, active concept drift approaches are well-suited. In addition, active concept drift detection methods would lessen the cost of unnecessary retraining by the learners; however, the computational cost may increase due to continuous monitoring for possible concept drifts in data streams. In addition, upon concept drift detection, the defined parameters (delta, threshold etc.) may need to be changed if heuristic-based approaches are not fully used. Subsequently, these changes may require additional fine-tuning of the integrated base learners for accurate and precise results.

The efficient detection of concept drift is followed by an adaptation procedure, where the model is retrained for adaptation. However, the retraining process cannot be initiated without knowledge of the data, type of concept drift and deciding the portion of data to be used for retraining. Data resampling can be done before retraining when the deviations are small in the data distribution. If the detection concept drift is sudden, then the latest subset of data can be used for retraining. However, concept drift detection methods can result in false alarms. These may require additional evaluation, such as distinguishing between true and false alarms through Type I and Type II error measurements, or by using a request and verify approach (S. Yu, Wang, & Príncipe, 2018), and testing and retraining. In addition, the computational effort must be considered before retraining to avoid overusing disk space in the case of data storage and processing capabilities in low-latency applications where results are produced in real time (e.g. in autonomous vehicles, traffic forecasting, etc.).

5 Future work

The research presented in the thesis offers potential directions for future work. The concept drift detection and adaptation algorithms proposed in this study can be applied across various domains in smart cities, such as smart homes, smart health, smart traffic, and smart energy. Currently, the implemented algorithms employ active concept drift detection approaches, which allow for exploring passive approaches in real-world scenarios. Additionally, a combination of active and passive approaches could be utilised to develop an algorithm for concept drift detection and adaptation, as proposed by (Disabato & Roveri, 2022). The implementation of the algorithm can be extended to detect and adapt to specific types of concept drift (gradual, abrupt, and recurrent). Moreover, it is still challenging to distinguish between true and false alarms generated by concept drift detectors, especially in real-world cases where adaptation strategies often require prior knowledge about data. The proposed methods by Disabato and Roveri (2022); S. Yu et al. (2018) could be explored to formulate efficient ways to address this challenge. Applying concept drift detection methods in the real world is challenging and highly needed, as described in Section 4.1. Therefore, the use cases presented in Publications **IV and V** are well-suited to further experiment with the proposed algorithms and explore earlier mentioned future directions.

The study proposes cloud and edge computing paradigm-based distributed architectures to handle computationally demanding analyses, enable smart services, and support concept drift detection and adaptation in different smart cities scenarios. The proposed platforms are tested for limited use cases and require further exploitation in other real-world scenarios. For example, the containerised architecture proposed in Publication **III** supports integrating different smart city scenarios and workflows but has not been explored for digital twin applications, as proposed in (Xu et al., 2023). Additionally, the live migration of computationally demanding analyses for efficient resource allocation requires manual intervention at this stage. In future, automated mechanisms could be formulated where data from system-level logs from telemetry could be held for resource allocation for large-scale jobs. The developed platform allows seamless integration of other tools. Therefore, to enhance the data governance within the proposed EMDC solution, data management tools like Apache Ranger (*Apache Ranger – Introduction*, 2014) and Apache Atlas (*Apache Atlas – Data Governance and Metadata framework for Hadoop*, 2016) could be integrated. The implemented smart campus use case in Publication **III** provides opportunities to configure more EMDCs across the campus and explore federated learning-based predictive learners integrated with concept drift detection methods. Lastly, due to limited research on edge platform architectures built over a hybrid cloud-edge continuum, the implemented architecture was not benchmarked and compared to other architectures for its efficiency and performance. It would be interesting to explore such an evaluation for building efficient smart city platforms in the future.

6 Conclusions

Modern technologies like big data, AI, edge and cloud computing, the Internet of Things, and cyber-physical systems have transformed traditional urban space-building methods. City stakeholders aim to develop data-driven urban policies to achieve sustainable societal goals. Sensor networks, IoT, economic platforms, social networks, and others generate enormous amounts of data at variable frequencies. The data from smart cities needs to be harnessed to improve the quality of well-being, the environment, transportation, infrastructure, and others for improved living standards. However, the data from smart cities is diverse and produced at different frequencies, and managing such large-scale data is still challenging. This thesis studied the challenges for computationally demanding processes in smart cities, as well as concept drift detection in predictive modelling to support decision-making and its limitations for real-world applicability. In particular, the focus was on building distributed data pipeline architectures to support predictive modelling, concept drift detection and adaptation, and decision-making processes.

Three research questions were answered in the five publications included in this thesis. **RQ1** looked for methods supporting concept drift detection for smart services in smart cities. Publications **II & III** introduced algorithms with real-world streaming data and were evaluated using synthetic datasets with induced concept drifts. Later experiments were performed with real-world scenarios. **RQ2** explored and implemented architectures that can implement data pipelines supporting automated concept drift detection and adaptation. Considering the existing limitations, Publication **I, II** contributed a cloud computing paradigm-based architecture to support automated concept drift and adaptation. Publication **III** proposed a novel edge computing architecture developed over a hybrid cloud-edge continuum to address the existing challenges in the state-of-the-art and to support automated concept drift detection. The proposed architectures were tested with real-world use cases. **RQ3** outlined challenges for concept drift detection and adaptation methods in real-world scenarios. Publications **IV & V** highlighted the need and challenges for concept drift detection methods in predictive modelling through two use cases from the real world. Later Publications**II-V**, listed the challenges from the tests and provided possible improvements and future directions.

The results of this thesis highlight the importance of concept drift detection and adaptation methods for improved performance of predictive models, which can be affected due to concept drift in real-world cases. In addition, the experiments showed that integrated concept drift detection and adaptation methods could improve the performance of predictive models employed in the real world. Moreover, the implementation of comprehensive data pipeline architecture highlighted some demanding requirements for implementing such architectures in smart cities, such as integration of different smart cities scenarios, analytics-driven resource optimisation, support for automated concept drift detection and adaptation, and improved data governance.

References

Abbasi, M., Abduli, M., Omidvar, B., & Baghvand, A. (2014). Results uncertainty of support vector machine and hybrid of wavelet transform-support vector machine models for solid waste generation forecasting. *Environmental Progress & Sustainable Energy*, *33*(1), 220–228.

Abbasi, M., & El Hanandeh, A. (2016). Forecasting municipal solid waste generation using artificial intelligence modelling approaches. *Waste management*, *56*, 13–22.

The action plan for 2020. (2016). Retrieved from `https://www.metro.tokyo.lg.jp/english/about/plan/index.html`

Agrahari, S., & Singh, A. K. (2022). Disposition-based concept drift detection and adaptation in data stream. *Arabian Journal for Science and Engineering*, *47*(8), 10605–10621.

Alberg, D., Last, M., & Kandel, A. (2012). Knowledge discovery in data streams with regression tree methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(1), 69–78.

Allam, Z., & Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. *Cities*, *89*, 80–91.

Almalki, F. A., Alsamhi, S. H., Sahal, R., Hassan, J., Hawbani, A., Rajput, N., ... Breslin, J. (2021). Green iot for eco-friendly and sustainable smart cities: future directions and opportunities. *Mobile Networks and Applications*, 1–25.

Alshmrany, S. (2022). Adaptive learning style prediction in e-learning environment using levy flight distribution based cnn model. *Cluster Computing*, *25*(1), 523–536.

Anagnostopoulos, T., Kostakos, P., Zaslavsky, A., Kantzavelou, I., Tsotsolas, N., Salmon, I., ... Harle, R. (2021). Challenges and solutions of surveillance systems in iot-enabled smart campus: a survey. *IEEE Access*, *9*, 131926–131954.

Androutsopoulou, A., Karacapilidis, N., Loukis, E., & Charalabidis, Y. (2019). Transforming the communication between citizens and government through ai-guided chatbots. *Government information quarterly*, *36*(2), 358–367.

Apache Atlas – Data Governance and Metadata framework for Hadoop. (2016). Retrieved 2022-11-01, from `https://atlas.apache.org/#/`

Apache hadoop. (2006). Retrieved from `https://hadoop.apache.org/`

Apache Ranger – Introduction. (2014). Retrieved 2022-11-01, from `https://ranger.apache.org/`

Apache spark. (2018). Retrieved from `https://spark.apache.org/`

Ashwini, B., Savithramma, R., & Sumathi, R. (2022). Artificial intelligence in smart city applications: an overview. In *2022 6th international conference on intelligent computing and control systems (iciccs)* (pp. 986–993).

Badii, C., Belay, E. G., Bellini, P., Marazzini, M., Mesiti, M., Nesi, P., ... others (2018). Snap4city: A scalable iot/ioe platform for developing smart city applications. In *2018 ieee smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (smartworld/scalcom/uic/atc/cbdcom/iop/sci)* (pp. 2109–2116).

Bellini, P., Nesi, P., Paolucci, M., & Zaza, I. (2018). Smart city architecture for data ingestion and analytics: Processes and solutions. In *2018 ieee fourth international conference on big data computing service and applications (bigdataservice)* (pp. 137–144).

Bifet, A., & Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 siam international conference on data mining* (pp. 443–448).

Boracchi, G., Carrera, D., Cervellera, C., & Maccio, D. (2018). Quanttree: Histograms for change detection in multivariate data streams. In *International conference on machine learning* (pp. 639–648).

Bourhnane, S., Abid, M. R., Lghoul, R., Zine-Dine, K., Elkamoun, N., & Benhaddou, D. (2020). Machine learning for energy consumption prediction and scheduling in smart buildings. *SN Applied Sciences*, *2*, 1–10.

Brzezinski, D., & Stefanowski, J. (2013). Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 81–94.

Cano, A., & Krawczyk, B. (2020). Kappa updated ensemble for drifting data stream mining. *Machine Learning*, *109*, 175–218.

Carrera, D. (2020). Learning and adaptation to detect changes and anomalies in high-dimensional data. *Special Topics in Information Technology*, 63–75.

Cavalcante, R. C., & Oliveira, A. L. (2015). An approach to handle concept drift in financial time series based on extreme learning machines and explicit drift detection. In *2015 international joint conference on neural networks (ijcnn)* (pp. 1–8).

Cenedese, A., Zanella, A., Vangelista, L., & Zorzi, M. (2014). Padova smart city: An urban internet of things experimentation. In *Proceeding of ieee international symposium on a world of wireless, mobile and multimedia networks 2014* (pp. 1–6).

Cerquitelli, T., Proto, S., Ventura, F., Apiletti, D., & Baralis, E. (2019). Automating concept-drift detection by self-evaluating predictive model degradation. *arXiv preprint arXiv:1907.08120*.

Chen, G., & Zhang, J. (2022). Applying artificial intelligence and deep belief network to predict traffic congestion evacuation performance in smart cities. *Applied Soft Computing*, *121*, 108692.

Chen, J., Huang, S., BalaMurugan, S., & Tamizharasi, G. (2021). Artificial intelligence based e-waste management for environmental planning. *Environmental Impact Assessment Review*, *87*, 106498.

Cheng, B., Longo, S., Cirillo, F., Bauer, M., & Kovacs, E. (2015). Building a big data platform for smart cities: Experience and lessons from santander. In *2015 ieee international congress on big data* (pp. 592–599).

Cicirelli, F., Guerrieri, A., Spezzano, G., & Vinci, A. (2017). An edge-based platform for dynamic smart city applications. *Future Generation Computer Systems*, *76*, 106–118.

Clohessy, T., Acton, T., & Morgan, L. (2014). Smart city as a service (scaas): A future roadmap for e-government smart city cloud computing initiatives. In *2014 ieee/acm 7th international conference on utility and cloud computing* (pp. 836–841).

Cohen, L., Avrahami-Bakish, G., Last, M., Kandel, A., & Kipersztok, O. (2008). Real-time data mining of non-stationary data streams from sensor networks. *Information Fusion*, *9*(3), 344–353.

Corchado, J. M., Chamoso, P., Hernández, G., Gutierrez, A. S. R., Camacho, A. R., González-Briones, A., ... others (2021). Deepint. net: A rapid deployment platform for smart territories. *Sensors*, *21*(1), 236.

Costa, C., & Santos, M. Y. (2016). Basis: A big data architecture for smart cities. In *2016 sai computing conference (sai)* (pp. 1247–1256).

CUTLER. (2018). Retrieved 2022-01-31, from `https://www.cutler-h2020.eu/`

Disabato, S., & Roveri, M. (2022). Tiny machine learning for concept drift. *IEEE Transactions on Neural Networks and Learning Systems*.

Ditzler, G. (2016). A study of an incremental spectral meta-learner for nonstationary environments. In *2016 international joint conference on neural networks (ijcnn)* (pp. 38–44).

Ditzler, G., Roveri, M., Alippi, C., & Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, *10*(4), 12–25.

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the sixth acm sigkdd international conference on knowledge discovery and data mining* (pp. 71–80).

Dong, F., Zhang, G., Lu, J., & Li, K. (2018). Fuzzy competence model drift detection for data-driven decision support systems. *Knowledge-Based Systems*, *143*, 284–294.

Ekman, A. (2020). Smart cities: Chinese ambitions in the time of coronavirus. *Politique etrangere*(3), 141–151.

ElHusseini, H., Assi, C., Moussa, B., Attallah, R., & Ghrayeb, A. (2020). Blockchain, ai and smart grids: The three musketeers to a decentralized ev charging infrastructure. *IEEE Internet of Things Magazine*, *3*(2), 24–29.

Elwell, R., & Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, *22*(10), 1517–1531.

Fenza, G., Gallo, M., & Loia, V. (2019). Drift-aware methodology for anomaly detection in smart grid. *IEEE Access*, *7*, 9645–9657.

Fernando, H., & Marshall, J. (2020). What lies beneath: Material classification for autonomous excavators using proprioceptive force sensing and machine learning. *Automation in Construction*, *119*, 103374.

Filonov, P., Lavrentyev, A., & Vorontsov, A. (2016). Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. *arXiv preprint arXiv:1612.06676*.

Fingrid, o. (2017, May). *Fingrid.* Retrieved from `https://www.fingrid.fi/en/`

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, *46*(4), 44.

Ge, L., Li, S., Wang, Y., Chang, F., & Wu, K. (2020). Global spatial-temporal graph convolutional network for urban traffic speed prediction. *Applied Sciences*, *10*(4), 1509.

General assembly of the european innovation partnership on smart cities and communities (eip-scc). (2021, Jul). Retrieved from `https://smart-cities-marketplace.ec.europa.eu/news-and-events/events/2018/eip-scc-general-assembly-2018-0`

Gilman, E., Keskinarkaus, A., Tamminen, S., Pirttikangas, S., Röning, J., & Riekki, J. (2015). Personalised assistance for fuel-efficient driving. *Transportation Research Part C: Emerging Technologies*, *58*, 681–705.

Gonçalves Jr, P. M., de Carvalho Santos, S. G., Barros, R. S., & Vieira, D. C. (2014). A comparative study on concept drift detectors. *Expert Systems with Applications*, *41*(18), 8144–8156.

Gulcan, E. B., & Can, F. (2023). Unsupervised concept drift detection for multi-label data streams. *Artificial Intelligence Review*, *56*(3), 2401–2434.

Habibzadeh, H., Kaptan, C., Soyata, T., Kantarci, B., & Boukerche, A. (2019). Smart city system design: a comprehensive study of the application and data planes. *ACM Computing Surveys (CSUR)*, *52*(2), 1–38.

Halstead, B., Koh, Y. S., Riddle, P., Pears, R., Pechenizkiy, M., Bifet, A., … Coulson, G. (2022). Analyzing and repairing concept drift adaptation in data stream classification. *Machine Learning*, *111*(10), 3489–3523.

Harel, M., Mannor, S., El-Yaniv, R., & Crammer, K. (2014). Concept drift detection through resampling. In *International conference on machine learning* (pp. 1009–1017).

Herath, H., & Mittal, M. (2022). Adoption of artificial intelligence in smart cities: A comprehensive review. *International Journal of Information Management Data Insights*, *2*(1), 100076.

Hernández-Ocaña, B., Hernández-Torruco, J., Chávez-Bosquez, O., Calva-Yáñez, M. B., & Portilla-Flores, E. A. (2019). Bacterial foraging-based algorithm for optimizing the power generation of an isolated microgrid. *Applied Sciences*, *9*(6), 1261.

Hossain, S. A., Rahman, M. A., & Hossain, M. A. (2018). Edge computing framework for enabling situation awareness in iot based smart city. *Journal of Parallel and Distributed Computing*, *122*, 226–237.

Hu, R. (2019). The state of smart cities in china: The case of shenzhen. *Energies*, *12*(22), 4375.

Huang, J., & Koroteev, D. D. (2021). Artificial intelligence for planning of energy and waste management. *Sustainable Energy Technologies and Assessments*, *47*, 101426.

Ighalo, J. O., Adeniyi, A. G., & Marques, G. (2021). Artificial intelligence for surface water quality monitoring and assessment: A systematic literature analysis. *Modeling Earth Systems and Environment*, *7*(2), 669–681.

Iwashita, A. S., de Albuquerque, V. H. C., & Papa, J. P. (2019). Learning concept drift with ensembles of optimum-path forest-based classifiers. *Future Generation Computer Systems*, *95*, 198–211.

Juanatey, D., Naya, M., Baamonde, T., & Bellas, F. (2021). Developing a simulation model for autonomous driving education in the robobo smartcity framework. *Engineering Proceedings*, *7*(1), 49.

Juyal, S., Sharma, S., & Shukla, A. S. (2021). Smart skin health monitoring using ai-enabled cloud-based iot. *Materials Today: Proceedings*, *46*, 10539–10545.

Kabir, M. A., Keung, J. W., Bennin, K. E., & Zhang, M. (2019). Assessing the significant impact of concept drift in software defect prediction. In *2019 ieee 43rd annual computer software and applications conference (compsac)* (Vol. 1, pp. 53–58).

Kadwe, Y., & Suryawanshi, V. (2015). A review on concept drift. *Iosr J. Comput. Eng*, *17*(1), 20–26.

Kaneko, R., Miyaguchi, K., & Yamanishi, K. (2017). Detecting changes in streaming data with information-theoretic windowing. In *2017 ieee international conference on big data (big data)* (pp. 646–655).

Karimian, M., & Beigy, H. (2023). Concept drift handling: A domain adaptation perspective. *Expert Systems with Applications*, *224*, 119946.

Ke, R., Cui, Z., Chen, Y., Zhu, M., Yang, H., & Wang, Y. (2020). Edge computing for real-time near-crash detection for smart transportation applications. *arXiv preprint arXiv:2008.00549*.

Khan, L. U., Yaqoob, I., Tran, N. H., Kazmi, S. A., Dang, T. N., & Hong, C. S. (2020). Edge-computing-enabled smart cities: A comprehensive survey. *IEEE Internet of Things Journal*, *7*(10), 10200–10232.

Krawczyk, B., & Cano, A. (2018). Online ensemble learning with abstaining classifiers for drifting and noisy data streams. *Applied Soft Computing*, *68*, 677–692.

Kubler, S., Främling, K., Zaslavsky, A., Doukas, C., Olivares, E., Fortino, G., ... others (2022). Iot platforms initiative. In *Digitising the industry internet of things connecting the physical, digital and virtualworlds* (pp. 265–292). River Publishers.

Kumar, A., Kaur, P., & Sharma, P. (2015). A survey on hoeffding tree stream data classification algorithms. *CPUH-Res. J*, *1*(2), 28–32.

Kumar, N. M., Chand, A. A., Malvoni, M., Prasad, K. A., Mamun, K. A., Islam, F., & Chopra, S. S. (2020). Distributed energy resources and the application of ai, iot, and blockchain in smart grids. *Energies*, *13*(21), 5739.

Kuncheva, L. I. (2011). Change detection in streaming multivariate data using likelihood detectors. *IEEE transactions on knowledge and data engineering*, *25*(5), 1175–1180.

Lilliu, F., Loi, A., Recupero, D. R., Sisinni, M., & Vinyals, M. (2019). An uncertainty-aware optimization approach for flexible loads of smart grid prosumers: A use case on the cardiff energy grid. *Sustainable Energy, Grids and Networks*, *20*, 100272.

Lima, M., Neto, M., Silva Filho, T., & Fagundes, R. A. d. A. (2022). Learning under concept drift for regression—a systematic literature review. *IEEE Access*, *10*, 45410–45429.

Liu, A., Lu, J., & Zhang, G. (2020). Concept drift detection via equal intensity k-means space partitioning. *IEEE transactions on cybernetics*, *51*(6), 3198–3211.

Liu, P., Zhang, Y., Kong, D., & Yin, B. (2019). Improved spatio-temporal residual networks for bus traffic flow prediction. *Applied Sciences*, *9*(4), 615.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*.

Lu, J., Liu, A., Song, Y., & Zhang, G. (2020). Data-driven decision support under concept drift in streamed big data. *Complex & Intelligent Systems*, *6*(1), 157–163.

Lu, T., Chen, X., McElroy, M. B., Nielsen, C. P., Wu, Q., & Ai, Q. (2020). A reinforcement learning-based decision system for electricity pricing plan selection by smart grid end users. *IEEE Transactions on Smart Grid*, *12*(3), 2176–2187.

Luo, H., Cai, H., Yu, H., Sun, Y., Bi, Z., & Jiang, L. (2019). A short-term energy prediction system based on edge computing for smart city. *Future Generation Computer Systems*, *101*, 444–457.

Mäkelä, T., Helfenstein, S., Lerkkanen, M.-K., & Poikkeus, A.-M. (2018). Student participation in learning environment improvement: Analysis of a co-design project in a finnish upper secondary school. *Learning Environments Research*, *21*, 19–41.

Maltezos, E., Karagiannidis, L., Dadoukis, A., Petousakis, K., Misichroni, F., Ouzounoglou, E., ... Amditis, A. (2021). Public safety in smart cities under the edge computing concept. In *2021 ieee international mediterranean conference on communications and networking (meditcom)* (pp. 88–93).

Manias, D. M., Shaer, I., Yang, L., & Shami, A. (2021). Concept drift detection in federated networked systems. In *2021 ieee global communications conference (globecom)* (pp. 1–6).

Massaro, A., Maritati, V., Giannone, D., Convertini, D., & Galiano, A. (2019). Lstm dss automatism and dataset optimization for diabetes prediction. *Applied Sciences*, *9*(17), 3532.

McGregor, C. (2018). Using constructive research to structure the path to transdisciplinary innovation and its application for precision public health with big data analytics. *Technology Innovation Management Review*, *8*(8).

Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., ... Riekki, J. (2019). Implementing big data lake for heterogeneous data sources. In *2019 ieee 35th international conference on data engineering workshops (icdew)* (pp. 37–44).

Mehmood, H., Hiltunen, M., Makkonen, T., Immonen, M., Pirttikangas, S., & Heikkilä, R. (2021). Road map for implementing ai-driven oulu smart excavator. In

Isarc. proceedings of the international symposium on automation and robotics in construction (Vol. 38, pp. 819–826).

Mehmood, H., Khalid, A., Kostakos, P., Gilman, E., & Pirttikangas, S. (2024). A novel edge architecture and solution for detecting concept drift in smart environments. *Future Generation Computer Systems*, *150*, 127-143. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167739X23003230` doi: https://doi.org/10.1016/j.future.2023.08.023

Mehmood, H., Kostakos, P., Cortes, M., Anagnostopoulos, T., Pirttikangas, S., & Gilman, E. (2021). Concept drift adaptation techniques in distributed environment for real-world data streams. *Smart Cities*, *4*(1), 349–371.

Morello, R., Mukhopadhyay, S. C., Liu, Z., Slomovitz, D., & Samantaray, S. R. (2017). Advances on sensing technologies for smart cities and power grids: A review. *IEEE Sensors Journal*, *17*(23), 7596–7610.

Mullins, P. D., & Shwayri, S. T. (2016). Green cities and "it839": A new paradigm for economic growth in south korea. *Journal of Urban Technology*, *23*(2), 47–64.

Naqvi, N., Rehman, S. U., & Islam, M. Z. (2022). Windrift: Early detection of concept drift using corresponding and hierarchical time windows. In *Data mining: 20th australasian conference, ausdm 2022, western sydney, australia, december 12–15, 2022, proceedings* (pp. 73–89).

Nguyen, H.-L., Woon, Y.-K., Ng, W.-K., & Wan, L. (2012). Heterogeneous ensemble for feature drifts in data streams. In *Advances in knowledge discovery and data mining: 16th pacific-asia conference, pakdd 2012, kuala lumpur, malaysia, may 29–june 1, 2012, proceedings, part ii 16* (pp. 1–12).

Pandya, S., Srivastava, G., Jhaveri, R., Babu, M. R., Bhattacharya, S., Maddikunta, P. K. R., ... Gadekallu, T. R. (2023). Federated learning for smart cities: A comprehensive survey. *Sustainable Energy Technologies and Assessments*, *55*, 102987.

Park, C. H., & Kang, Y. (2016). An active learning method for data streams with concept drift. In *2016 ieee international conference on big data (big data)* (pp. 746–752).

Pathan, M., Patel, N., Yagnik, H., & Shah, M. (2020). Artificial cognition for applications in smart agriculture: A comprehensive review. *Artificial Intelligence in Agriculture*, *4*, 81–95.

Patil, M. M. (2019). Handling concept drift in data streams by using drift detection methods. In *Data management, analytics and innovation: Proceedings of icdmai 2018, volume 2* (pp. 155–166).

Peña, M., Lanzarini, L., Cerrada, M., Cabrera, D., & Sánchez, R.-V. (2021). Data-driven gearbox fault severity diagnosis based on concept drift. In *2021 ieee fifth ecuador*

technical chapters meeting (etcm) (pp. 1–6).
Pereira, J., Batista, T., Cavalcante, E., Souza, A., Lopes, F., & Cacho, N. (2022). A platform for integrating heterogeneous data and developing smart city applications. *Future Generation Computer Systems*, *128*, 552-566. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167739X2100426X` doi: https://doi.org/10.1016/j.future.2021.10.030
Perera, C., Qin, Y., Estrella, J. C., Reiff-Marganiec, S., & Vasilakos, A. V. (2017). Fog computing for sustainable smart cities: A survey. *ACM Computing Surveys (CSUR)*, *50*(3), 1–43.
Pesaranghader, A., Viktor, H. L., & Paquet, E. (2018). Mcdiarmid drift detection methods for evolving data streams. In *2018 international joint conference on neural networks (ijcnn)* (pp. 1–9).
Petrolo, R., Loscri, V., & Mitton, N. (2014). Towards a smart city based on cloud of things. In *Proceedings of the 2014 acm international workshop on wireless and mobile technologies for smart cities* (pp. 61–66).
Peyman, M., Copado, P. J., Tordecilla, R. D., Martins, L. d. C., Xhafa, F., & Juan, A. A. (2021). Edge computing and iot analytics for agile optimization in intelligent transportation systems. *Energies*, *14*(19), 6309.
Pocock, A., Yiapanis, P., Singer, J., Luján, M., & Brown, G. (2010). Online nonstationary boosting. In *Multiple classifier systems: 9th international workshop, mcs 2010, cairo, egypt, april 7-9, 2010. proceedings 9* (pp. 205–214).
Puliafito, A., Tricomi, G., Zafeiropoulos, A., & Papavassiliou, S. (2021). Smart cities of the future as cyber physical systems: Challenges and enabling technologies. *Sensors*, *21*(10), 3349.
Puri, M., Varde, A., Du, X., & De Melo, G. (2018). Smart governance through opinion mining of public reactions on ordinances. In *2018 ieee 30th international conference on tools with artificial intelligence (ictai)* (pp. 838–845).
Sadeghi, A., Younes Sinaki, R., Young, W. A., & Weckman, G. R. (2020). An intelligent model to predict energy performances of residential buildings based on deep neural networks. *Energies*, *13*(3), 571.
Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., ... others (2014). Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, *61*, 217–238.
Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojicic, D. S. (2017). Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *ACM Computing Surveys (Csur)*, *50*(6), 1–37.

Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojicic, D. S. (2018). Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *ACM Computing Surveys (CSUR)*, *50*(6), 78.

Selim, M., Zhou, R., Feng, W., & Quinsey, P. (2021). Estimating energy forecasting uncertainty for reliable ai autonomous smart grid design. *Energies*, *14*(1), 247.

Sengan, S., Subramaniyaswamy, V., Nair, S. K., Indragandhi, V., Manikandan, J., & Ravi, L. (2020). Enhancing cyber–physical systems with hybrid smart city cyber security architecture for secure public data-smart network. *Future generation computer systems*, *112*, 724–737.

Shaikh, R. A. J., Naidu, H., & Kokate, P. A. (2021). Next-generation wsn for environmental monitoring employing big data analytics, machine learning and artificial intelligence. In *Evolutionary computing and mobile sustainable networks: Proceedings of icecmsn 2020* (pp. 181–196).

Shariati, H., Yeraliyev, A., Terai, B., Tafazoli, S., & Ramezani, M. (2019). Towards autonomous mining via intelligent excavators. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops* (pp. 26–32).

Sharma, A., Georgi, M., Tregubenko, M., Tselykh, A., & Tselykh, A. (2022). Enabling smart agriculture by implementing artificial intelligence and embedded sensing. *Computers & Industrial Engineering*, *165*, 107936.

Shrestha, M., Mitra, C., Rahman, M., & Marzen, L. (2023). Mapping and predicting land cover changes of small and medium size cities in alabama using machine learning techniques. *Remote Sensing*, *15*(1), 106.

Silva, B. N., Khan, M., & Han, K. (2018). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable cities and society*, *38*, 697–713.

Silva, B. N., Khan, M., Jung, C., Seo, J., Muhammad, D., Han, J., … Han, K. (2018). Urban planning and smart city decision management empowered by real-time data processing using big data analytics. *Sensors*, *18*(9), 2994.

Silva, R. M., & Almeida, T. A. (2021). How concept drift can impair the classification of fake news. In *Anais do ix symposium on knowledge discovery, mining and learning* (pp. 121–128).

Smart america: Smart cities usa. (2014). Retrieved from `https://smartamerica.org/teams/smart-cities-usa/`

Smart city adoption timeline. (2018, Jan). Frost Sullivan. Retrieved from `https://www.giiresearch.com/report/fs604427-smart-city-adoption-timeline.html`

Son, B., Kim, C., Kim, C., & Lee, D. (2020). Expert-emulating excavation trajectory planning for autonomous robotic industrial excavator. In *2020 ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 2656–2662).

Song, Y., Wang, Y., Liu, F., & Zhang, Y. (2017). Development of a hybrid model to predict construction and demolition waste: China as a case study. *Waste Management*, *59*, 350–361.

Suárez-Cetrulo, A. L., Cervantes, A., & Quintana, D. (2019). Incremental market behavior classification in presence of recurring concepts. *Entropy*, *21*(1), 25.

Sun, Y., Wang, Z., Bai, Y., Dai, H., & Nahavandi, S. (2018). A classifier graph based recurring concept detection and prediction approach. *Computational intelligence and neuroscience*, *2018*.

Sunny, A. I., Zhao, A., Li, L., & Sakiliba, S. K. (2020). Low-cost iot-based sensor system: A case study on harsh environmental monitoring. *Sensors*, *21*(1), 214.

Syed, A. S., Sierra-Sosa, D., Kumar, A., & Elmaghraby, A. (2021). Iot in smart cities: A survey of technologies, practices and challenges. *Smart Cities*, *4*(2), 429–475.

Thurzo, A., Strunga, M., Urban, R., Surovková, J., & Afrashtehfar, K. I. (2023). Impact of artificial intelligence on dental education: A review and guide for curriculum update. *Education Sciences*, *13*(2), 150.

Tuli, S., Basumatary, N., Gill, S. S., Kahani, M., Arya, R. C., Wander, G. S., & Buyya, R. (2020). Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments. *Future Generation Computer Systems*, *104*, 187–200.

Vítor, G., Rito, P., Sargento, S., & Pinto, F. (2022). A scalable approach for smart city data platform: Support of real-time processing and data sharing. *Computer Networks*, *213*, 109027.

Wan, K., Hughes, D., Man, K. L., & Krilavičius, T. (2010). Composition challenges and approaches for cyber physical systems. In *2010 ieee international conference on networked embedded systems for enterprise applications* (pp. 1–7).

Wang, F., Zhang, M., Wang, X., Ma, X., & Liu, J. (2020). Deep learning for edge computing applications: A state-of-the-art survey. *IEEE Access*, *8*, 58322–58336.

Waste, W. (2018). 2.0: A global snapshot of solid waste management to 2050/silpa kaza. In *Lisa yao, perinaz bhada-tata and frank van woerden. world banc group.*

Wojtkiewicz, J., Katragadda, S., & Gottumukkala, R. (2018). A concept-drift based predictive-analytics framework: Application for real-time solar irradiance forecasting. In *2018 ieee international conference on big data (big data)* (pp. 5462–5464).

Xu, H., Berres, A., Yoginath, S. B., Sorensen, H., Nugent, P. J., Severino, J., … Sanyal, J. (2023). Smart mobility in the cloud: Enabling real-time situational awareness and cyber-physical control through a digital twin for traffic. *IEEE Transactions on Intelligent Transportation Systems*.

Yang, L., Manias, D. M., & Shami, A. (2021). Pwpae: An ensemble framework for concept drift adaptation in iot data streams. In *2021 ieee global communications conference (globecom)* (pp. 01–06).

Yang, Z., Al-Dahidi, S., Baraldi, P., Zio, E., & Montelatici, L. (2019). A novel concept drift detection method for incremental learning in nonstationary environments. *IEEE transactions on neural networks and learning systems*, *31*(1), 309–320.

Yi, H., Bui, K.-H. N., & Jung, H. (2019). Implementing a deep learning framework for short term traffic flow prediction. In *Proceedings of the 9th international conference on web intelligence, mining and semantics* (pp. 1–8).

Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., & David, B. (2015). A literature survey on smart cities. *Sci. China Inf. Sci.*, *58*(10), 1–18.

Yu, H., Zhang, Q., Liu, T., Lu, J., Wen, Y., & Zhang, G. (2022). Meta-add: A meta-learning based pre-trained model for concept drift active detection. *Information Sciences*, *608*, 996–1009.

Yu, S., Wang, X., & Príncipe, J. C. (2018). Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels. *arXiv preprint arXiv:1806.10131*.

Yu, Y., & Zhang, N. (2019). Does smart city policy improve energy efficiency? evidence from a quasi-natural experiment in china. *Journal of Cleaner Production*, *229*, 501–512.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, *1*(1), 22–32.

Zhao, Z., Xu, J., Zang, Y., & Hu, R. (2021). Adaptive abnormal oil temperature diagnosis method of transformer based on concept drift. *Applied Sciences*, *11*(14), 6322.

Žliobaitė, I., Pechenizkiy, M., & Gama, J. (2016). An overview of concept drift applications. *Big data analysis: new algorithms for a new society*, 91–114.

Original publications

I Mehmood, H., Gilman, E., Cortes, M., Kostakos, P., Byrne, A., Valta, K., ... & Riekki, J. (2019, April). Implementing big data lake for heterogeneous data sources. In 2019 ieee 35th international conference on data engineering workshops (icdew) (pp. 37-44). IEEE.

II Mehmood, H., Kostakos, P., Cortes, M., Anagnostopoulos, T., Pirttikangas, S., & Gilman, E. (2021). Concept drift adaptation techniques in distributed environment for real-world data streams. Smart Cities, 4(1), 349-371.

III Mehmood, H., Khalid, A., Kostakos, P., Gilman, E., Pirttikangas, S. (2024). A novel edge architecture and solution for detecting concept drift in smart environments. Future Generation Computer Systems, 150, 127-143. doi: https://doi.org/10.1016/j.future.2023.08.023

IV Cao, J., Mehmood, H., Liu, X., Tarkoma, S., Gilman, E., & Su, X. (2022). Fighting Pandemics with Augmented Reality and Smart Sensing-based Social Distancing. IEEE Computer Graphics and Applications.

V Mehmood, H., Hiltunen, M., Makkonen, T., Immonen, M., Pirttikangas, S., & Heikkilä, R. (2021). Road map for implementing AI-driven Oulu Smart excavator. In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 38, pp. 819-826). IAARC Publications.

Original publications are not included in the electronic version of the dissertation.

887. Alorwu, Andy (2023) User perceptions of personal data in healthcare : ethics, reuse, and valuation

888. Nyameke, Emmanuel (2023) Project identity formation and maintenance in a temporary organization environment

889. Koivikko, Niina (2023) Silica-titania supported vanadia catalysts in the utilization of industrial sulfur-contaminated gaseous methanol streams

890. Ye, Liang (2023) Automated multi-modal recognition of school violence

891. Nguyen, Phong (2023) Neural scene representations for learning-based view synthesis

892. Muhammad, Usman (2023) Video representation and deep learning techniques for face presentation attack detection

893. Opoku Asare, Kennedy (2023) Assessing mental disorders with digital biomarkers

894. Hashemi, Ramin (2023) Reconfigurable intelligent surface in URLLC wireless systems

895. Bounab, Yazid (2023) Social media mining for affective and business cues

896. Taimisto, Marjaana (2023) Preparation, characterization, and formation pathways of ruthenium(II) complexes with chalcogenoether ligands

897. Moeen Taghavi, Ehsan (2023) Network optimization in RIS-assisted communications

898. Baubekova, Aziza (2023) Catchment-estuary-coastal systems under climate change and anthropogenic pressure

899. Akbari, Mahdi (2023) Application of remote-sensing and machine-learning in studying the climatic and anthropogenic drivers of water bodies drying in data-scarce transboundary basins

900. Kaikkonen, Pentti (2023) Characteristics of ultrafine/nanostructured bainite formation in low-temperature ausformed medium-carbon steels

901. Su, Zhuo (2023) LBP inspired efficient deep convolutional neural networks for visual representation learning

902. Cui, Yawen (2023) Few-shot learning for image classification

903. Bhayani, Snehal (2023) Sparse resultant-based methods with their applications to generalized cameras

Book orders:
Virtual book store
https://verkkokauppa.omapumu.com/fi/

UNIVERSITY OF OULU P.O. Box 8000 FI-90014 UNIVERSITY OF OULU FINLAND

ACTA UNIVERSITATIS OULUENSIS

UNIVERSITY OF OULU

ISBN 978-952-62-3851-7 (Paperback)
ISBN 978-952-62-3852-4 (PDF)
ISSN 0355-3213 (Print)
ISSN 1796-2226 (Online)