Deep Learning-Based Sustainable and Secure Communications for Next-Generation Wireless Networks

Wu, Qirui

Awarding institution:
King's College London

Deep Learning-Based Sustainable and Secure Communications for Next-Generation Wireless Networks

Qirui Wu

Supervisor: Prof. Mohammad Shikh-Bahaei

The Department of Engineering
King's College London

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2024

I would like to dedicate this thesis to my loving husband and parents.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 100,000 words including footnotes.

Qirui Wu
September 2024

Acknowledgements

I extend my heartfelt gratitude to my supervisor, Prof. Mohammad Shikh-Bahaei, whose guidance and expertise shaped this thesis. I am deeply thankful to Dr. Yirun Zhang and Prof. Zhaohui Yang for their invaluable insights and guidance. My family's unwavering support and sacrifices have been my constant motivation. I also extend my thanks to my friends, colleagues, and participants for their encouragement and contributions. This work wouldn't have been possible without their collective support and belief in my abilities.

Abstract

Deep learning methods have seen increasing importance and rapid advancements in time-series forecasting. These methods, which leverage the power of neural networks, have proven to be highly effective in capturing complex patterns and dependencies in data, offering significant improvements over traditional forecasting techniques. With the ability to model nonlinear relationships and learn from big data, deep learning has revolutionised time-series forecasting, leading to more accurate and robust predictions across a diversity of domains.

In wireless communications, numerous time-series forecasting problems arise, e.g., predicting channel states and user mobility. Addressing these challenges is crucial for optimising network performance, enhancing energy efficiency, and ensuring robust communications. Deep learning provides powerful tools to tackle these problems by learning from historical data and making precise predictions, enabling adaptive network management.

This thesis presents two significant applications of deep learning for solving time-series forecasting problems in wireless communications:

First, we propose a novel deep learning-based algorithm for channel prediction and energy efficiency (EE) optimisation in an intelligent reflecting surface (IRS) aided Terahertz communication system. Specifically, a multi-antenna base station with an IRS with massive reflecting elements is designed to serve multiple moving users. A deep learning-based prediction-optimisation scheme is presented where we first propose a transformer encoder with channel index embedding (TE-CIE) deep learning model for time-varying channel prediction. With the help of channel prediction, an EE optimisation algorithm is then designed to maximise the EE in advance based on the predicted channel state information (CSI). Finally, we combine both methods to construct a deep learning-based prediction-optimisation scheme. Specifically, the future CSI is predicted by TE-CIE and the IRS phase-shift and precoding matrices are optimised in advance. Simulation results demonstrate that our proposed channel prediction method achieves close-to-optimal performance in terms of low mean absolute error and a much faster speed than the two baseline models. We demonstrate that the proposed EE optimisation algorithm outperforms the baseline algorithms in terms of much better EE under diverse parameter settings. Finally, the

proposed prediction-optimisation scheme achieves at least twice the EE improvement compared to the baseline methods in the literature.

Second, we focus on designing a robust deep-learning model to predict user mobility under malicious Global Navigation Satellite System (GNSS) spoofing attacks for unmanned aerial vehicle (UAV) swarm position optimisation. UAV swarms have become a promising solution to enhance modern wireless communication in complicated environments. However, due to the existence of real-world malicious attacks, the performance of prediction and optimisation methods used for UAV swarms are easily degraded. In this paper, we propose a novel deep learning-based user mobility prediction, user assignment and drone position optimisation scheme for a UAV swarm-enabled wireless communication system with the existence of malicious GNSS spoofing attackers. Specifically, a robust deep learning-based user mobility prediction model, namely denoising autoencoder recurrent transformer (DART), is designed and various efficient user assignment and drone position optimisation methods are proposed. Simulation results show that the proposed deep learning-based prediction-optimisation scheme can provide up to 30% higher overall sum rate compared with the adversarial trained long short-term memory (LSTM) baseline and almost doubled the overall sum rate compared with the vanilla LSTM baseline.

To reduce the computational complexity of the DART model without compromising its performance, we employ a technique called knowledge distillation for sustainable purposes. By distilling the knowledge learned by the complex DART model into a simpler and more computationally efficient architecture, such as a smaller Gated Recurrent Unit (GRU) model, we aim to retain the essential information for user mobility prediction and drone position optimisation. This distilled model can offer much faster inference times and reduced resource requirements while preserving much of the performance achieved by the original DART model, making it more practical for real-time deployment in UAV swarm-enabled wireless communication systems under the threat of malicious GNSS spoofing attacks. Simulation results demonstrate that the optimised sum rate using the distilled GRU student model's predicted user locations can achieve almost 99% compared to the Transformer teacher model. Meanwhile, the inference time of the student model is only 4% compared to the teacher model.

In conclusion, our research emphasises the potential of deep learning for time-series forecasting in next-generation wireless communication scenarios. By addressing key forecasting problems, e.g., predicting channel states and user mobility, our deep learning-based algorithms demonstrate significant improvements in energy efficiency and network performance. The proposed solutions for Terahertz communications, IRS systems, and UAV swarm networks show the robustness and accuracy of deep learning models in complex and dynamic environments. Future research will continue to explore innovative

deep-learning techniques to solve additional time-series forecasting challenges and further optimise wireless communication systems.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

2D Two-Dimensional

3D Three-Dimensional

AI Artificial Intelligence

AoD Angle of Departure

AWGN Additive White Gaussian Noise

BCD Block Coordinate Descent

BCE Binary Cross-Entropy

BCS Block Coordinate Searching

BERT Bidirectional Encoder Representations from Transformers

BS Base Station

CE Cross-Entropy

CMA-ES Covariance Matrix Adaptation Evolution Strategy

CNN Convolutional Neural Network

CSA Cumulative Step-length Adaptation

CSI Channel State Information

DART Denoising Autoencoder Recurrent Transformer

DBS Drone Base Station

EE Energy Efficiency

EPR Exploration and PReferential

FGSM Fast Gradient Sign Method

FPO Floating-Point Overflow

GELU Gaussian Error Linear Unit

GM Gauss Markov

GNSS Global Navigation Satellite System

GPT Generative Pre-Trained Transformers

GRU Gated Recurrent Unit

ILP Integer Linear Programming

IoT Internet of Things

IRS Intelligent Reflecting Surface

LOS Line-Of-Sight

LS Local Search

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MBS Macro Base Station

MIMO Multiple Input Multiple Output

MLP Multi-Layer Perception

MSE Mean Squared Error

MU-MISO Multi-User Multiple-Input Single-Output

NLP Natural Language Processing

ONNX Open Neural Network Exchange

PGD Projected Gradient Descent

ReLU Rectified Linear Unit

RF Radio Frequency

RMSprop Root Mean Square propagation

RNN Recurrent Neural Network

RS Random Selection

RWP Random Waypoint

SCA Successive Convex Approximation

SDP Successive Differential Programming

SOTA State-Of-The-Art

T5 Text-to-Text Transfer Transformer

TE-CIE Transformer Encoder with Channel Index Embedding

THz Terahertz

UAV Unmanned Aerial Vehicle

ULA Uniform Linear Array

ViT Vision Transformer

Chapter 1

Introduction

In recent years, deep learning techniques have significantly transformed the landscape of wireless communications [1–3]. By exploiting the capabilities of neural networks, deep learning models can effectively address the complex and dynamic nature of wireless communication systems. These models excel in processing large volumes of data and adapting to various communication scenarios, thereby offering substantial enhancements over traditional methods. The application of deep learning in wireless communications has led to breakthroughs in optimising network performance, improving Energy Efficiency (EE), and enhancing security measures, making it a necessary tool for advancing next-generation communication technologies.

Some recent comprehensive surveys [1–3] on the application of deep learning in mobile and wireless networking highlight the substantial progress made in the field, demonstrating how deep learning models can improve various aspects of wireless communication, including signal detection, spectrum management, and network optimisation. These surveys also emphasise the effectiveness of deep learning methods in handling the intricate and variable conditions of wireless networks including tackling time-series forecasting problems.

In the era of modern wireless communications, a diversity of time-series forecasting challenges emerge, including the prediction of channel states and user mobility [1–3]. Solving these issues is essential for maximising network performance under the dynamic communication environment, boosting EE, and ensuring reliable communications. Deep learning offers effective solutions to these challenges by analysing historical data and generating accurate forecasts, which facilitate adaptive network management.

For example, accurately predicting channel states can significantly improve the allocation of resources and the overall efficiency of the network. Deep learning models, such as Recurrent Neural Networks (RNNs) [4] and Convolutional Neural Networks (CNNs) [5], have shown great promise in this area by capturing the temporal and spatial dependencies in wireless channel data. Additionally, forecasting user mobility patterns allows for better

planning and resource distribution in the network, leading to enhanced user experiences and reduced latency. Models like long short-term memory (LSTM) networks [6] and transformer-based architectures [7] excel at learning these mobility patterns from vast datasets, providing accurate and timely predictions.

Furthermore, deep learning offers promising solutions for robust prediction problems in wireless networks, such as mitigating the impact of adversarial attacks. By leveraging advanced techniques in anomaly detection and pattern recognition, deep learning models can identify and mitigate the effects of malicious activities, ensuring the security and reliability of communication systems. This capability is crucial in protecting the network from adversarial attacks that aim to disrupt network operations or compromise data integrity.

1.1 Aim

This thesis aims to leverage advanced deep-learning methodologies to address critical time-series forecasting challenges in wireless communication systems. By focusing on the prediction of dynamic parameters such as channel states and user mobility patterns, the research aims to enhance network performance by optimising EE and improving overall system efficiency. Additionally, the thesis seeks to develop robust predictive models capable of mitigating the impact of adversarial attacks carried out by malicious attackers and other security threats in wireless networks. Through comprehensive empirical evaluations and innovative algorithmic developments, the goal is to establish deep learning as a critical tool for achieving sustainable, secure, and reliable wireless communications in diverse environments.

1.2 Objectives

The first objective of the thesis centres on the development of a sophisticated deep learning-based algorithm tailored for Intelligent Reflecting Surface (IRS) aided Terahertz (THz) communication systems [8–10]. The first aspect of this objective involves creating a predictive model using deep learning techniques to forecast the behaviour of the wireless channel. THz frequencies are highly sensitive to environmental conditions and obstacles, demanding accurate channel prediction to optimise transmission reliability and throughput. Deep learning models such as Transformer-based architectures will be explored to capture the complex temporal dependencies of the Channel State Information (CSI). These models aim to predict how the channel conditions will change over time, enabling efficient network management and resource allocation. The second aspect focuses on enhancing EE within IRS-aided THz communication systems [11–13]. IRS technology utilises reconfigurable

reflecting surfaces to manipulate signal propagation, thereby enhancing coverage and capacity. A sophisticated EE optimisation algorithm will be employed to optimise the phase shift and precoding matrices of IRS elements based on predicted channel states. This optimisation seeks to maximise EE by minimising power consumption while maintaining or enhancing communication quality. Heuristic optimisation methods will be explored to achieve this objective effectively.

The second objective of the thesis focuses on designing a robust deep learning model specifically tailored to predict user mobility patterns within Unmanned Aerial Vehicle (UAV) swarm-enabled wireless communication systems [14–17], particularly under the threat of malicious Global Navigation Satellite System (GNSS) spoofing attacks [18–20]. The primary challenge lies in ensuring accurate user mobility prediction under potential disruptions caused by GNSS spoofing attacks. GNSS spoofing can manipulate user location information, leading to erroneous positioning of UAVs and impacting communication reliability and efficiency. The deep learning model aims to mitigate these effects by learning from historical data and identifying patterns that are reliable to such adversarial disruptions. Key components of this objective include developing a deep learning architecture, such as the denoising autoencoder recurrent transformer (DART), capable of learning and predicting complex user mobility behaviours in dynamic and uncertain environments. The model will be trained on extensive datasets to capture diverse scenarios and adapt to varying levels of spoofing intensity. Additionally, techniques like knowledge distillation [21] will be explored to transfer the learned knowledge from the complex DART model to a more computationally efficient architecture such as the Gated Recurrent Unit (GRU) [22], ensuring scalability and real-time applicability. Furthermore, this objective entails exploring efficient user assignment strategies and optimising drone position planning algorithms based on deep learning predictions. By integrating these predictive and optimisation capabilities, this objective seeks to enhance the reliability and effectiveness of UAV swarm-enabled wireless communication systems under the persistent threat of GNSS spoofing attacks.

1.3 Contributions

In this thesis, we have studied deep learning applications to IRS-aided THz and UAV swarm communication systems.

Specifically, Chapter 3 proposes a novel deep learning-based algorithm for channel prediction and EE optimisation in an IRS-aided THz communication system. The contributions of this chapter are as follows:

- We extend the commonly used IRS system models [23–25] with fixed user position and static fading channel in the literature to a novel time-varying setting. More specifically, user mobility and time-varying THz channel characteristics are considered in our system model. Our user-mobility-based system model is more realistic for THz IRS systems since even a slow-moving user will cause a significant change in the CSI. Such a severe scenario greatly impacts CSI-based optimisation and cannot be neglected in THz IRS system models.

- We design a deep learning-based channel prediction method to predict the channel matrix between the IRS and the users for the following time slot so that the BS and IRS can optimise the precoding matrix and phase shifts in advance. The proposed low-complexity multi-channel CSI prediction deep learning model, namely Transformer Encoder with a Channel Index Embedding (TE-CIE), outperforms the literature's conventional multi-layer perception (MLP) and RNN-based channel prediction methods. The proposed TE-CIE model minimises the amount of sequential operation by the attention mechanism and allows parallelised prediction of multiple channels by using the CIE technique.

- We study the EE optimisation problem for IRS-aided multi-user multiple-input single-output (MU-MISO) wireless systems with THz communications and propose a covariance matrix adaptation evolution strategy (CMA-ES) and Dinkelbach's method-based EE optimisation algorithm. Our proposed CMA-ES-based optimisation method can maintain the same complexity as the cross-entropy (CE) method in the literature by using the same number of iterations and candidates whilst providing a much better EE optimisation performance than all three baselines.

- We propose a novel deep learning-based channel prediction and EE optimisation scheme which outperforms the baseline methods by at least doubling the EE. The proposed prediction-optimisation scheme guarantees high EE optimisation performance with low overall computational complexity compared with the methods in the literature. To the best of our knowledge, this is the first work that predicts the CSI of a time-varying channel in an IRS-aided THz network using a deep learning-based technique and then optimises the network EE.

Chapter 4 proposes a novel deep learning-based user mobility prediction, user assignment and drone position optimisation scheme for a UAV swarm-enabled wireless communication system in the presence of malicious GNSS spoofing attackers. We further distil the proposed Transformer-based teacher model into a smaller GRU model based on the knowledge distillation method to reduce the time complexity of the model while

maintaining its prediction power. The main contributions of this chapter are summarised as follows:

- We propose a deep learning-based user mobility prediction, user assignment and drone position optimisation scheme which is robust to malicious GNSS spoofing attacks. The proposed deep learning model forecasts user locations, on which we construct and solve assignment and position optimisation problems. It is worth noting that we measure the spoofing success hit ratio as a spoofing probability instead of considering a specific spoofing activity. Therefore, our proposed algorithm can work in all cases of spoofing activities in general.

- We apply a realistic user mobility model, i.e., the exploration and preferential (EPR) model, which can better represent real-world human mobility than the widely used Random Waypoint (RWP) model. This setting enhances the authenticity and reliability of our simulation results and theoretical analyses.

- We design a deep learning-based user mobility prediction method, namely DART, to predict the user locations for the next time slot so that the UAV swarm can optimise the user assignment and drone positions for the next time slot in advance. Based on the proposed deep learning architecture, we design an adversarial pre-training and fine-tuning scheme where the model learns to detect noisy locations and reconstructs them to enhance the robustness of our deep learning against malicious GNSS spoofing attacks.

- We distil the Transformer teacher model into a smaller GRU model based on the knowledge distillation method to reduce the time complexity of the model while maintaining its prediction power.

- We construct a robust optimisation problem which takes the user mobility prediction error into account. We have added a mathematical expression for the robust version of the optimisation problem considering the uncertainty of user location estimation to enhance the novelty of the paper and demonstrate the difference between the optimisation studied in this paper and those studied in the closely relevant prior works.

- We investigate two user assignment and drone position optimisation methods, successive convex approximation (SCA) and successive differential programming (SDP), and demonstrate the superiority of the latter over the former in terms of a larger maximum turning angle range, better numerical stability and lower overall computational complexity through convergence and complexity analysis. Our proposed SDP

method has proven to be more efficient and robust compared with those traditional SCA-based methods in the closely relevant prior works such as [26–28].

1.4 Thesis Structure

The remainder of this thesis is organised as follows:

Chapter 2 is dedicated to providing the preliminaries and overview of IRS and THz communications, UAV swarm networks, GNSS malicious spoofing attacks, time-series deep learning algorithms, adversarial training and knowledge distillation techniques. Chapters 3 and 4 are the main body of the thesis, and in each chapter, a novel solution to a specific deep learning application problem in IRS-aided THz and UAV swarm communication systems is proposed and studied. The conclusions and some suggestions for future works are provided in Chapter 5.

1.5 Research Publications

1.5.1 Papers Included In This Thesis

The contributions and novelties of this thesis have been drawn from and are disseminated through the following technical papers:

Papers Published

Papers that have been published or accepted are listed as follows:

Chapter 3: [C1] **Q. Wu**, Y. Zhang, C. Huang, Y. Chau, Z. Yang and M. Shikh-Bahaei, "Energy Efficient Intelligent Reflecting Surface Assisted Terahertz Communications," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, Montreal, QC, Canada, 2021, pp. 1-6, doi: 10.1109/ICCWorkshops50388.2021.9473736.

Chapter 4: [C2] **Q. Wu**, Y. Zhang, Z. Yang and M. Shikh-Bahaei, "Knowledge Distillation-Based Robust UAV Swarm Communication Under Malicious Attacks," in *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, Denver, CO, USA, 2024, pp. 1023-1029, doi: 10.1109/ICCWorkshops59551.2024.10615342.

Chapter 3: [J1] **Q. Wu**, Y. Zhang, Z. Yang and M. Shikh-Bahaei, "Deep Channel Prediction-Based Energy-Efficient Intelligent Reflecting Surface-Aided Terahertz Communications," in *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 2946-2960, April 2024, doi: 10.1109/TWC.2023.3304597.

Chapter 4: [J2] **Q. Wu**, Y. Zhang, Z. Yang and M. Shikh-Bahaei, "Deep Learning for Secure UAV Swarm Communication Under Malicious Attacks," in *IEEE Transactions on Wireless Communications, In Press*, doi: 10.1109/TWC.2024.3419923.

1.5.2 Papers Not Included In This Thesis

Papers that are not included in this thesis are listed as follows:

[C1] Y. Zhang, **Q. Wu** and M. Shikh-Bahaei, "Ensemble Learning Based Robust Cooperative Sensing in Full-Duplex Cognitive Radio Networks," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, Dublin, Ireland, 2020, pp. 1-6.

[C2] Y. Zhang, **Q. Wu** and M. Shikh-Bahaei, "Vertical Federated Learning Based Privacy-Preserving Cooperative Sensing in Cognitive Radio Networks," in *2020 IEEE Globecom Workshops (GC Wkshps)*, Taipei, Taiwan, 2020, pp. 1-6.

[J1] Y. Zhang, **Q. Wu** and M. R. Shikh-Bahaei, "On Ensemble Learning-Based Secure Fusion Strategy for Robust Cooperative Sensing in Full-Duplex Cognitive Radio Networks," in *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6086-6100, Oct. 2020.

[J2] Y. Zhang, **Q. Wu** and M. Shikh-Bahaei, "A Pointer Network Based Deep Learning Algorithm for User Pairing in Full-Duplex Wi-Fi Networks," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12363-12368, Oct. 2020.

Chapter 2

Preliminaries

This chapter briefly lays the background behind the research that will be presented later in the thesis. The starting point is the introduction of IRS and THz communications, followed by the concept of UAV swarm networks and GNSS malicious spoofing attacks. After that, we briefly introduce the contents of deep learning algorithms for time-series modelling, adversarial training and knowledge distillation techniques.

2.1 Intelligent Reflecting Surface

An IRS is an innovative technology poised to revolutionise wireless communication networks by enhancing signal strength, coverage, and overall efficiency [8, 29, 30, 23, 31, 32]. At its core, an IRS consists of a large number of small, reconfigurable reflecting elements that can intelligently manipulate electromagnetic waves. These elements, often implemented as passive reflecting surfaces or meta-surfaces, are strategically deployed to control and optimise the propagation of wireless signals in various environments. The concept draws inspiration from concepts in meta-materials, beamforming, and Multiple Input Multiple Output (MIMO) technologies, aiming to overcome the limitations of traditional wireless systems and pave the way for next-generation wireless communication.

At the heart of an IRS lies its ability to passively manipulate radio waves [9, 31, 29, 33]. The individual elements within an IRS, equipped with tunable properties, can alter the phase, amplitude, and polarisation of incident electromagnetic waves. By doing so, they can effectively redirect, focus, or shape the wireless signals in a manner that optimises the transmission and reception characteristics of the overall communication system. This precise control allows for the mitigation of signal blockages, reductions in interference, and the creation of custom signal patterns tailored to specific communication requirements.

The deployment scenarios for IRSs span a wide range of applications, encompassing both indoor and outdoor environments [8, 34]. In indoor settings such as offices, malls, or

smart homes, IRS can mitigate signal attenuation caused by obstacles, improving signal quality and coverage. Similarly, in outdoor scenarios like urban areas or stadiums where signal interference and fading are common challenges, IRS can enhance connectivity by intelligently redirecting signals around obstacles or toward desired locations. This technology has the potential to significantly enhance the performance of wireless networks, enabling seamless connectivity in environments previously considered problematic for wireless communication.

Moreover, the adaptability and programmability of IRS elements contribute to their versatility [8, 35–37]. Through advanced algorithms and machine learning techniques, these surfaces can dynamically adjust their configurations in response to changing environmental conditions, user demands, or network requirements. This adaptive capability allows the IRS to continuously optimise signal propagation, offering a high degree of flexibility and efficiency in wireless communication networks.

In summary, IRSs represent a paradigm shift in wireless communication technologies. By harnessing the principles of wave manipulation and employing an array of tunable elements, IRS has the potential to revolutionise signal propagation, significantly improve network performance, and enable the seamless connectivity demanded by the evolving landscape of wireless communication.

2.2 Terahertz Communications

THz communication systems represent a cutting-edge technology poised to revolutionise wireless communication [10, 11]. Operating in the electromagnetic spectrum between microwave and infrared frequencies, THz waves span from 0.1 to 10 THz, offering huge potential for high-speed data transmission, imaging, and sensing applications. The technology utilises the unique properties of THz waves, promising unprecedented data rates and capabilities beyond the limitations of existing wireless technologies.

At the core of THz communication lies its remarkable bandwidth capacity [38, 39]. With frequencies much higher than those used in current wireless systems, THz waves enable data transmission rates reaching multiple terabits per second. This huge bandwidth potential holds the promise of mitigating network congestion and meeting the escalating demands of data-intensive applications such as high-definition video streaming, virtual reality, and the Internet of Things (IoT).

Moreover, THz waves have exceptional abilities to penetrate various materials while being non-ionising, making them suitable for imaging and sensing applications [40]. This characteristic makes THz communication systems invaluable for applications like medical imaging, security screening, and material characterisation, enabling non-invasive

and high-resolution imaging through materials that are opaque to visible light or other electromagnetic waves.

However, deploying THz communication systems comes with its set of challenges [10, 11, 38, 39]. One significant challenge is the propagation loss of THz waves due to absorption by atmospheric gases and moisture, limiting their range and reliability for long-distance communication. Scientists and engineers are actively exploring innovative solutions such as antenna design, signal processing techniques, and material advancements to mitigate these propagation challenges and expand the practical applications of THz technology.

In conclusion, THz communication systems represent a paradigm shift in wireless communication, offering unparalleled data rates and various applications in a diversity of fields. Despite facing challenges related to propagation limitations, ongoing research and technological advancements continue to drive the potential of THz waves, paving the way for a new era of high-speed, high-capacity wireless communication and transformative applications across industries.

2.3 UAV Swarm Communication Networks

A UAV swarm communication network represents an innovative paradigm in modern technological landscapes, leveraging the collaborative potential of multiple drones to establish a robust and versatile communication infrastructure [15, 41]. Comprising interconnected drones operating in unison, this network facilitates seamless data exchange, offering a dynamic solution across various industries and applications. Through coordinated efforts, UAV swarm communication networks redefine the conventional methods of information dissemination, surveillance, and connectivity.

At its core, a UAV swarm communication network is characterised by the interconnectivity and collective intelligence of multiple unmanned aerial vehicles [16, 14]. These drones communicate with each other using sophisticated algorithms and protocols, enabling them to act as nodes in a decentralised network. By utilising this collective intelligence, the swarm can adapt to changing environments, dynamically reconfigure, and efficiently distribute tasks among individual units, ensuring optimal performance and coverage.

One of the fundamental advantages of a UAV swarm communication network lies in its flexibility [17, 42]. Unlike traditional communication systems constrained by fixed infrastructure, these networks are highly mobile and adaptable. Drones within the swarm can immediately reposition themselves to establish and maintain connectivity, even in challenging or remote environments where conventional networks may be inaccessible or impractical.

Moreover, the scalability of UAV swarm communication networks is a key asset. By integrating additional drones into the swarm, the network's capabilities can be expanded, allowing for increased coverage, data throughput, and redundancy. This scalability is particularly advantageous in scenarios that demand extensive coverage or rapid response, such as disaster management, search and rescue operations, or large-scale surveillance initiatives.

In essence, UAV swarm communication networks represent a cutting-edge approach to communication and information exchange. Their ability to autonomously organise, adapt, and collaborate enables a wide array of applications across industries, including but not limited to disaster relief, precision agriculture, infrastructure inspection, and military operations. As technology continues to evolve, the potential for UAV swarm communication networks to revolutionise connectivity and data transmission is vast, promising innovative solutions to contemporary challenges.

2.4 GNSS Malicious Spoofing Attacks

GNSS has become integral to various sectors, providing precise positioning, navigation, and timing services worldwide [43, 18]. However, the increasing reliance on GNSS for critical infrastructure, transportation, finance, and military applications has faced vulnerabilities, notably the threat of malicious spoofing attacks. A GNSS spoofing attack involves the transmission of falsified signals to deceive GNSS receivers, leading to incorrect positioning, navigation, or timing information. This deceptive manipulation poses severe risks, potentially causing significant disruptions and security breaches across diverse sectors.

The mechanism of a GNSS spoofing attack involves generating counterfeit signals that mimic authentic satellite signals received by GNSS receivers [19, 20]. Attackers can produce fake signals using sophisticated equipment, broadcasting fake data to override authentic signals. By imitating real GNSS signals, attackers mislead receivers into calculating erroneous location information, leading to inaccurate positioning or navigation guidance. Unlike traditional jamming attacks that disrupt signal reception, spoofing attacks manipulate the receiver into accepting false location data as genuine, making them particularly challenging to detect.

The impact of a successful GNSS spoofing attack is multifaceted and far-reaching [44, 18]. In transportation, such attacks can cause chaos by misleading autonomous vehicles, drones, or maritime vessels, leading to accidents or deliberate deviations from intended routes. Within critical infrastructure, such as power grids or telecommunication networks, reliance on accurate timing synchronisation via GNSS makes them susceptible

to disruptions, potentially resulting in service outages or compromised data integrity. Moreover, financial systems and stock markets heavily dependent on precise timing from GNSS could face vulnerabilities, enabling fraudulent activities or market manipulations.

Detecting and mitigating GNSS spoofing attacks present considerable challenges. The complex and dynamic nature of GNSS signals, coupled with the sophistication of spoofing techniques, makes it difficult to distinguish between real and fake signals. Existing receiver technologies often lack robust security measures, leaving them vulnerable to exploitation. Additionally, as spoofing attacks do not disrupt signals outright, but rather manipulate received data, traditional detection methods designed for signal interference may prove ineffective, demanding the development of advanced detection and authentication mechanisms.

The existence of GNSS spoofing attacks causes a significant threat to various sectors heavily reliant on accurate positioning, navigation, and timing services. Understanding the mechanisms, potential impacts, and challenges associated with detecting and mitigating these attacks is crucial in developing robust solutions to protect critical systems and infrastructure against malicious exploitation. Addressing these vulnerabilities demands collaborative efforts from technology developers, policymakers, and security experts to fortify GNSS receivers, enhance detection capabilities, and establish resilient defence mechanisms against potential spoofing threats.

2.5 Deep Learning Algorithms For Time-Series Modelling

Time-series data involves sequential observations indexed by time, found in various domains such as finance, weather forecasting, healthcare, and more. Analysing and predicting patterns within time-series data requires models capable of capturing temporal dependencies and understanding sequential information. Deep learning techniques have emerged as powerful tools for handling time-series data due to their ability to learn intricate patterns and relationships within sequences.

2.5.1 Recurrent Neural Network

RNNs are a class of neural networks designed for sequential data processing, enabling the modelling of temporal dependencies [6, 22]. The architecture of RNNs involves recurrent connections that allow information to persist over time, which is shown in Fig. 2.1 where $\mathbf{x}_t$ is the input and $\mathbf{o}_t$ is the output vector to the RNN cell; $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ refer to the hidden state vectors of the $(t-1)$th and tth time slots, respectively; $\mathbf{W}_i$ and $\mathbf{b}_i$ are learnable weights and biases in the RNN cell. Each node in an RNN processes an input along with information

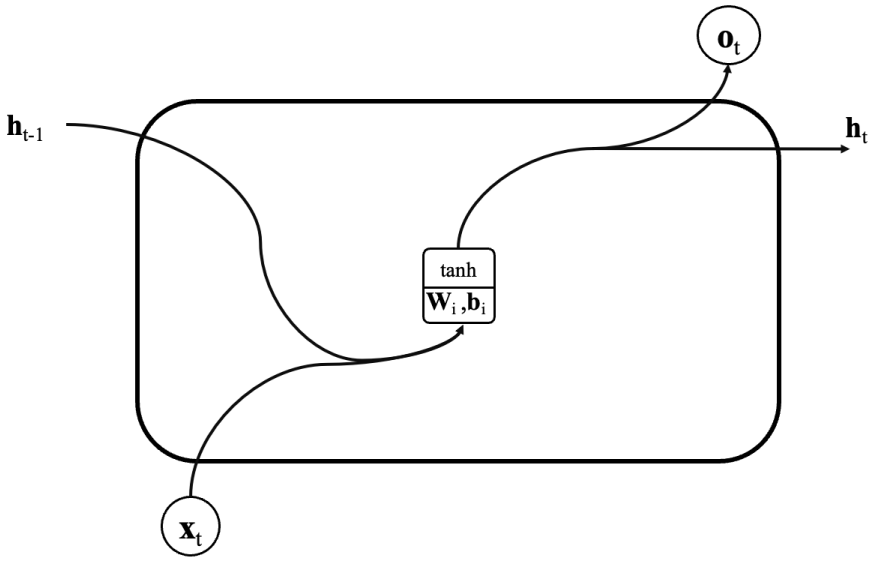

Fig. 2.1 RNN architecture.

from the previous node, forming a chain-like structure that is well-suited for time-series modelling.

However, traditional RNNs suffer from the vanishing or exploding gradient problem. As information traverses through many time steps, gradients can become extremely small (vanishing) or large (exploding), leading to difficulties in capturing long-term dependencies. This limitation hinders the effectiveness of RNNs in scenarios requiring the understanding of context over extended sequences.

Despite their limitations, RNNs have found applications in various domains, including natural language processing, speech recognition, and time-series analysis with short-term dependencies. Researchers have developed extensions to address the issues faced by traditional RNNs, such as LSTM networks and GRUs.

2.5.2 Long Short-Term Memory

LSTMs were introduced to mitigate the shortcomings of traditional RNNs in capturing long-range dependencies within sequential data [6]. The architecture of LSTM is shown in Fig. 2.2, where $\mathbf{x}_t$, $\mathbf{f}_t$, $\mathbf{i}_t$ and $\mathbf{o}_t$ are the input vector, the activation vector of the forget gate, the activation vector of the input gate and the activation vector of the output gate, respectively; $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ refer to the hidden state vectors of the $(t-1)$th and tth time slots; $\mathbf{c}_{t-1}$ and $\mathbf{c}_t$ refer to the cell state vectors of the $(t-1)$th and tth time slots; $\widetilde{\mathbf{c}}_t$ is the cell input activation vector; $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_c$ and $\mathbf{W}_o$ are the learnable weight matrices in the forget gate, input gate and output gate, respectively; $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_c$ and $\mathbf{b}_o$ are the learnable

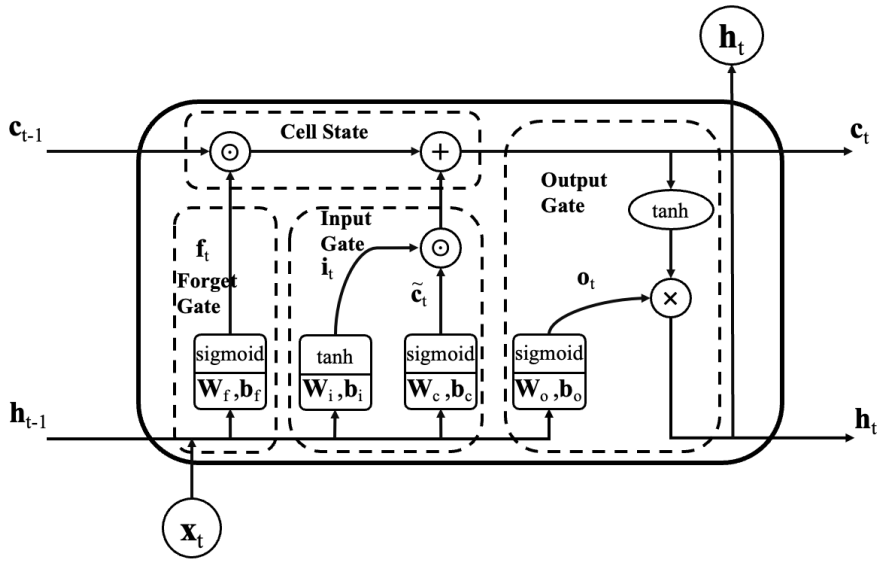

Fig. 2.2 LSTM architecture.

bias vectors in the forget gate, input gate and output gate, respectively. LSTMs incorporate specialised memory cells with gating mechanisms that control the flow of information. These gates (i.e., input, forget, and output gates) enable LSTMs to selectively retain or discard information over time, allowing them to maintain long-term dependencies more effectively.

The design of LSTMs, with their ability to remember information over long sequences and mitigate the vanishing gradient problem, has made them a popular choice in various time-series modelling tasks. They excel in scenarios where capturing and learning from complex temporal relationships across multiple time steps is crucial, such as in financial forecasting, speech recognition, and natural language understanding.

LSTMs have proven effective in handling time-series data with irregular time intervals or gaps and have been extensively employed in tasks involving sequential information processing, such as stock market predictions, energy load forecasting, and medical signal analysis. Their robustness in handling long sequences makes them a go-to choice for many time-series modelling applications.

2.5.3 Gated Recurrent Unit

GRUs represent another advancement in the realm of RNNs, offering a simpler architecture compared to LSTMs while maintaining competitive performance [22]. The architecture of GRU is shown in Fig. 2.3, where $\mathbf{x}_t$, $\mathbf{r}_t$, $\mathbf{z}_t$ and $\widetilde{\mathbf{h}}_t$ are the input vector, the activation vector of the reset gate, the activation vector of the update gate and the candidate activation vector,

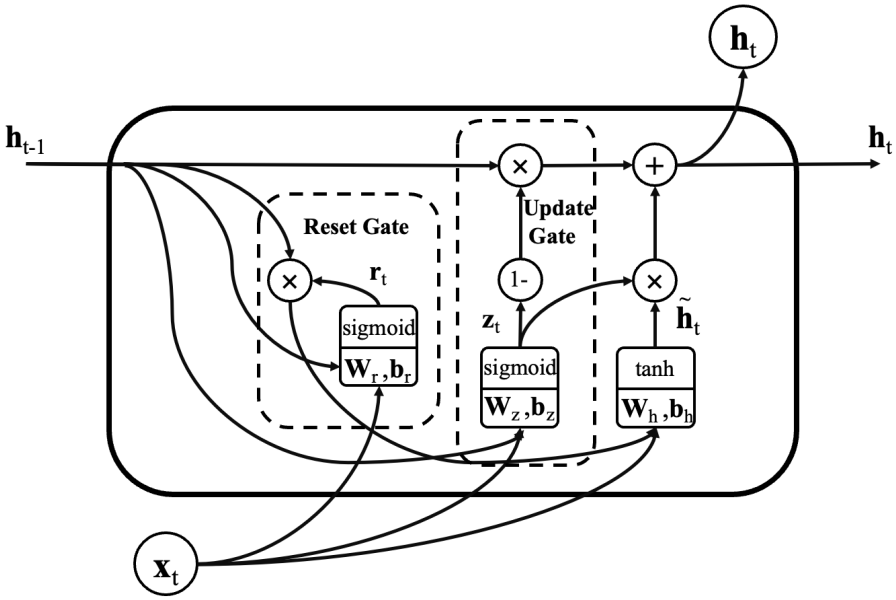

Fig. 2.3 GRU architecture.

respectively; $\mathbf{h}_{t-1}$ and $\mathbf{h}_t$ refer to the hidden state vectors of the $(t-1)$th and tth time slots, respectively; $\mathbf{W}_r$, $\mathbf{W}_z$, and $\mathbf{W}_h$ are the learnable weight matrices in the reset gate, update gate and candidate activation unit, respectively; $\mathbf{b}_f$, $\mathbf{b}_r$, $\mathbf{b}_z$ and $\mathbf{b}_h$ are the learnable bias vectors in the reset gate, update gate and candidate activation unit, respectively. GRUs aim to address the same challenges of modelling long-term dependencies but with fewer parameters and less complex computations compared to LSTMs.

GRUs merge the functionalities of the input and forget gates in LSTMs into a single update gate, simplifying the network architecture. This streamlined design reduces the computational burden, making GRUs faster to train and more suitable for datasets with limited resources or smaller sizes.

Despite their reduced complexity, GRUs have demonstrated effectiveness in various time-series applications, particularly when handling shorter sequences or scenarios where computational efficiency is a primary concern. They have been employed in tasks such as human activity recognition, speech synthesis, and music generation, showcasing their versatility in sequential data analysis.

2.5.4 Transformer Architecture

The Transformer architecture [7], introduced in the context of natural language processing (NLP), has reshaped sequence modelling by deviating from the recurrent structure of RNNs. Unlike RNN-based models, Transformers rely on self-attention mechanisms, which

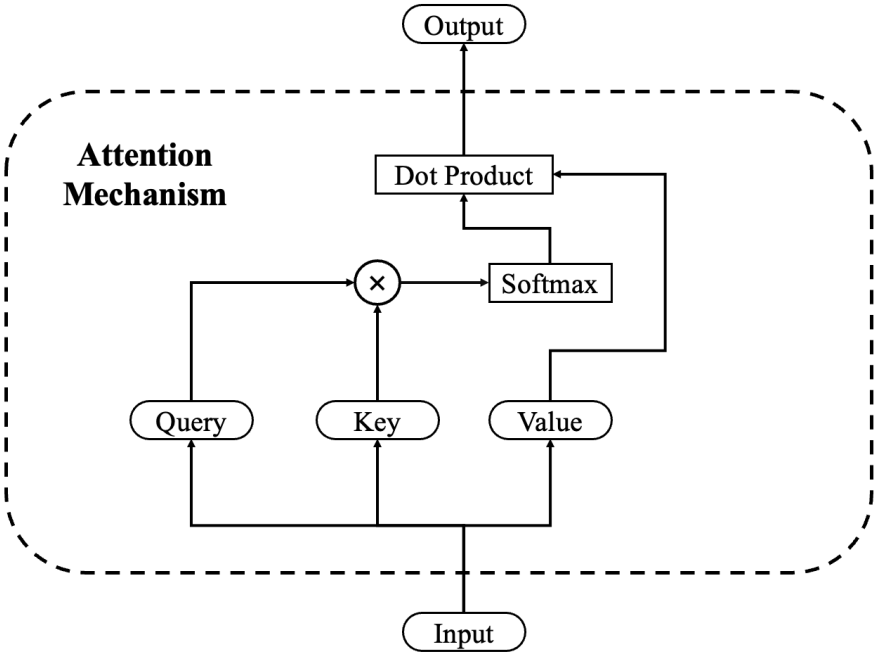

Fig. 2.4 Attention mechanism in the Transformer architecture.

is illustrated in Fig. 2.4, to capture relationships between different positions in the input sequence without recurrent connections.

Transformers leverage attention mechanisms that allow the model to weigh the significance of each element in the sequence concerning all other elements, enabling the extraction of complex dependencies and patterns from the entire sequence simultaneously. This parallel processing capability makes Transformers highly efficient for handling long sequences without the limitations of sequential computation.

Since the first Transformer [7] was proposed, there have been several significant improvements. Bidirectional Encoder Representations from Transformers (BERT) [45] introduced bidirectional training, allowing models to understand the context from both directions, which significantly enhanced performance across various NLP tasks. Generative Pre-Trained Transformers-3 (GPT-3) [46], with its 175 billion parameters, demonstrated the power of large-scale language models to perform numerous tasks with minimal task-specific training, showcasing the scalability of Transformer models. The Vision Transformer (ViT) [47] expanded the application of Transformers to image recognition, outperforming traditional convolutional neural networks when pre-trained on large datasets. Text-to-Text Transfer Transformer (T5) [48] unified NLP tasks into a text-to-text framework, simplifying the model architecture and improving performance across diverse benchmarks. Finally, the development of Multi-modal Large Language Models [49] integrated multiple modalities, enhancing the ability of models to process and generate

content across different data types, moving towards more comprehensive AI systems. These advancements collectively highlight the versatility and expanding capabilities of Transformer models in various domains.

Originally designed for language tasks, Transformers have been adapted for time-series modelling by considering the temporal nature of sequential data. Modifications such as Temporal Transformers have been proposed, tailoring the architecture to capture temporal dependencies within time-series data effectively.

In summary, RNNs, LSTMs, GRUs, and Transformers are prominent deep-learning architectures used in time-series modelling, each offering distinct advantages in capturing temporal dependencies and patterns within sequential data.

2.6 Adversarial Training

Adversarial training [50–52] is a fundamental technique in machine learning aimed at enhancing the robustness and performance of models, especially in the domain of deep learning. This approach involves training a model in an environment where it's exposed to adversarial examples—subtly modified inputs crafted to deceive the model. The primary goal is to fortify the model against such perturbations, thereby improving its ability to generalise and make accurate predictions on unseen data.

One aspect of adversarial training involves the creation of adversarial examples through methods like the fast gradient sign method (FGSM) or Projected Gradient Descent (PGD). These techniques manipulate input data by introducing imperceptible perturbations, leading the model to misclassify or produce erroneous outputs. By incorporating these examples into the training process, the model learns to recognise and adapt to these perturbations, bolstering its resilience to adversarial attacks.

Adversarial training serves as a proactive defence mechanism against potential vulnerabilities in machine learning models. Subjecting the model to adversarial examples during training fosters a robust learning process that encourages the model to detect and counteract potential threats. This approach helps mitigate the risk of exploitation by adversarial attacks in real-world scenarios, contributing to more reliable and secure AI systems.

Despite its effectiveness in improving model robustness, adversarial training comes with its challenges and trade-offs. Incorporating adversarial examples during training can increase computational complexity and training time since generating these examples requires additional iterations and computations. Moreover, there might be a trade-off between accuracy on clean data and robustness against adversarial attacks, where overly aggressive adversarial training could lead to decreased performance on normal data.

Adversarial training continues to be an active area of research in the pursuit of developing more resilient and trustworthy machine learning models. Ongoing studies focus on refining techniques, exploring novel defence mechanisms, and understanding the underlying vulnerabilities to create models that can better generalise across diverse and adversarial scenarios. As AI systems become increasingly integrated into various domains, the importance of adversarial training in fortifying these models against potential threats cannot be understated.

2.7 Knowledge Distillation

Knowledge distillation is a technique used in machine learning to transfer knowledge from a large, complex model (teacher) to a smaller, more lightweight model (student) [21]. This process involves training the student model to mimic the behaviour and predictions of the larger model by leveraging the information it provides. By distilling the knowledge from the teacher model, the student model can achieve comparable performance with reduced computational resources and memory requirements, making it suitable for deployment in resource-constrained environments such as mobile devices or edge devices.

The primary goal of knowledge distillation is to compress the knowledge contained within the teacher model into a more compact form that can be effectively learned by the student model. This compression is achieved by training the student model not only on the original dataset but also by incorporating additional information from the teacher model's predictions. The student model learns to generalise and capture the essential patterns, relationships, and decision-making processes present in the teacher model's predictions.

One of the key aspects of knowledge distillation is the use of soft targets during training. Instead of relying solely on the hard labels (ground truth) from the training data, the teacher model's softened probabilities or logits are used as targets for the student model. This enables the student model to learn not just the correct outputs but also the underlying probabilities and uncertainties associated with each prediction, allowing for a more nuanced understanding of the data distribution.

Knowledge distillation can significantly improve the performance of the student model by transferring the knowledge encapsulated in the teacher model. The distilled knowledge helps the student model generalise better, making it more robust against noise and variations in the data. Moreover, the computational efficiency gained through distillation enables faster inference times, making it practical for real-time applications where speed is crucial.

While knowledge distillation has shown promising results in various domains such as computer vision, natural language processing, and speech recognition, it is essential to note that the effectiveness of distillation depends on various factors, including the choice of

teacher model, student model architecture, hyperparameters, and the nature of the dataset. Researchers continue to explore and refine knowledge distillation techniques to enhance their applicability and effectiveness across different machine-learning tasks and scenarios.

Chapter 3

Deep Channel Prediction-Based Energy-Efficient Intelligent Reflecting Surface-Aided Terahertz Communications

3.1 Introduction

The heavy connectivity and spectral efficiency requirements of modern wireless networks have been attracting great attention to the energy consumption problems [53–55, 12, 56, 13, 57, 58]. Hence, EE has become an important performance indicator for designing green and sustainable future wireless networks.

One of the most recent emerging technologies for enhancing sustainable communications, namely IRS, shows significant potential for increasing EE for modern wireless communications [29, 30, 23, 31, 32]. An IRS consists of massive tiny reflecting elements where each element reflects electromagnetic waves by adjustable phase shifters. IRS adapts phase shifts to enhance beamforming to suppress interference among multiple users [31, 29, 33]. The IRS technology has been attracting researchers because of its potential to forward the receiving wireless signals without the help of power amplifiers, from an energy consumption perspective [35–37]. The IRS-assisted wireless systems have been researched by various works in the literature such as [30, 59, 60]. For example, the authors in [60] have proposed a hardware-efficient channel modelling method for IRS systems considering the coupling effect induced by the excessively large number of closely spaced patch antennas. Meanwhile, the authors in [30] have studied the EE optimisation problem for IRS-aided wireless systems. The results in [30] prove that the EE of IRS systems can be much better than that of conventional Amplify-and-Forward relaying systems if the

IRS phase shifts are properly designed. On the other hand, an energy-efficient IRS system has been proposed in [59] where the scenario of an IRS with an infinitely large number of reflecting elements serving a single user by passive beamforming is considered.

3.1.1 Prior Works

Researchers have been investigating the IRS-assisted communication systems for both indoor and outdoor environments in recent years [8, 34]. However, a recent paper [34] highlights the outperforming of IRS-aided communications in indoor environments as compared to outdoor use cases due to the presence of fewer scatterers in the former, which suggests the higher potential of deploying IRS-aided systems in indoor environments. On the other hand, it is also promising to combine THz communications with IRS systems to further improve the system performance of indoor communication networks. THz communications, which enables various novel applications, has become one of the potential technologies for ultra-high data-rate transmission [10, 11]. Due to its physical constraints, e.g., THz waves have poorer penetration and diffraction capabilities than those of millimetre waves and microwaves, THz systems depend more on reflection transmission compared with conventional systems [11]. As a consequence, introducing THz technology into the IRS systems can both overcome the shortages of THz communications and improve data rates in conventional IRS wireless systems. However, limited works have been published on the topic of THz-IRS systems in the literature. Two different algorithms are applied in [23] to iteratively optimise the hybrid precoding and IRS phase-shift matrices for the IRS-assisted THz system for maximising the sum rate of the system. Besides, a block coordinate searching (BCS) algorithm is proposed in [24] to jointly optimise the IRS's coordinates, phase shifts, THz sub-bands allocation and power control for sum-rate maximisation. However, maximising the sum rate of the system may cause an increase in energy consumption and degrade the EE. To design a green and sustainable THz IRS system, it is critical to optimise the EE of the system. To the best of our knowledge, the EE optimisation problem for IRS-assisted THz communications has not been thoroughly studied in the literature, and all the aforementioned works assume fixed user positions with static fading channels. Although the authors in the recent paper [61] propose a novel design of a distributed IRS framework that enhances the EE of indoor THz wireless communication systems, the mobility issue in THz communications caused by moving users is not taken into account.

In practice, the maximum radian Doppler frequency would be large even if the user moves slowly due to the high-frequency nature of THz communications. For example, with a 10 THz carrier frequency and a user moving at a speed of 1 m/s, the maximum radian Doppler frequency is approximately as high as 33 kHz. This indicates that the THz channel

is extremely sensitive to user motion. With such a severe Doppler effect, the fading channel changes over time dramatically, which makes optimisation results far away from optimum since the previously collected CSI becomes outdated in a short time. This motivates us to develop a strong algorithm which has the capability of capturing the temporal correlation of the THz fading channels to predict the future CSI. With the predicted CSI, the IRS-aided MU-MISO system can optimise the phase-shift and precoding matrices in advance. Traditional channel prediction methods include deterministic parameter-based models [62], auto-regressive predictive models [63, 64] and adaptive filtering techniques [65]. However, these methods either fail in non-stationary and fast-varying environments [63, 64, 62] or require high computational complexity in computing the second-order statistics of the time-varying channel [65].

With the rapid increase of computation power and data availability in recent years, deep learning algorithms have become a vital part of next-generation time-series forecasting strategies. Different from traditional domain expertise-informed parametric models, deep learning-based forecasting methods do not require any *a priori* knowledge and can provide an approach to learning temporal dynamics in a purely data-driven manner. The non-linear transformation and the universal approximation capability enable deep learning algorithms to learn complicated implicit patterns from massive data that traditional expertise-informed parametric models fail to capture. In the case that the underlying data pattern is extremely complicated, using a data-driven deep learning algorithm can significantly reduce the human supervision required for designing a reliable expertise-informed parametric model. Numerous successful deep learning-based time-series forecasting models have been proposed in the literature including both classic models such as LSTM networks and modern models such as Transformers [7].

The time-varying Rayleigh fading channel prediction problem has been studied in various papers [66–68]. For example, in [66], an MLP neural network model is implemented to predict the Rayleigh fading channel based on the conventional Jake's model. Moreover, an RNN model is proposed to further improve the prediction performance [67]. The authors in [68] propose a deep learning-based method for time-varying IRS channel prediction, which adopts an RNN-based novel architecture for joint channel decomposition and prediction. However, there are several drawbacks to these proposed methods. First, the existing studies only focus on single-channel prediction and require training one model for each channel separately, which is inefficient because the number of channels is huge. Second, the conventional Jake's model [69] used in [66] is deterministic, which cannot effectively represent the randomness of real-world fading channels. Third, the MLP model treats all past CSI equally due to its nature, which cannot extract the temporal correlation information effectively for time-series data. Although RNN models in [67, 68] have been proven to have a strong capability of capturing temporal correlation information, both their

training and inference speeds are slow due to the sequential input nature of the recurrent architecture [7]. All these drawbacks motivate us to develop an efficient and strong multi-channel prediction model that can predict the CSI of multiple channels simultaneously in real-time.

3.1.2 Contributions

In this chapter, we propose a novel deep learning-based time-varying fading channel prediction and EE optimisation scheme for indoor IRS-aided THz MU-MISO systems, where the base station (BS) and IRS have fixed positions whilst the users are moving slowly. A modified Clarke's Rayleigh fading model is used instead of the conventional Jake's model to better represent the randomness of real-world fading channels. The proposed deep learning channel prediction method collects previous CSI to predict the next CSI of the IRS-user channel. A TE-CIE deep learning model is developed to capture the temporal correlation of the fading channels and predict the future CSI. Our proposed TE-CIE model can predict the CSI of multiple channels in parallel in one model for IRS-assisted MU-MISO systems, which is much more efficient than MLP and RNN models. In addition, different from the sequential operation of RNN models, our proposed TE-CIE applies the self-attention mechanism to capture the temporal correlation of the history of the CSI data, which can be parallelised to minimise the amount of sequential computation in the model. Thus, the proposed TE-CIE model leads to a lower time complexity for sequential prediction than the RNN architectures proposed in [67, 68]. Based on the predicted CSI, we study the EE optimisation problem. To maximise the EE of the system for the next time slot in advance, a CMA-ES [70] and Dinkelbach's method-based EE optimisation algorithm are proposed to jointly optimise the BS's transmit power, the precoding matrix and the phase-shift matrix for IRS. Simulation results demonstrate that our proposed channel prediction method achieves close-to-optimal performance in terms of low mean absolute error (MAE) and much faster inference speed than MLP and RNN models. Moreover, the proposed EE optimisation algorithm outperforms three baseline algorithms on static fading channels from the existing works, i.e., the random selection (RS), local search (LS) and CE algorithms, in terms of much better EE under diverse parameter settings. Finally, our proposed deep learning-based prediction-optimisation scheme achieves at least two times EE improvement compared to the baseline methods in the literature.

3.1.3 Organisation and Notations

The remainder of this chapter is organised as follows. Section 3.2 introduces the IRS-assisted THz MU-MISO system model. Section 3.3 explains our proposed deep learning-

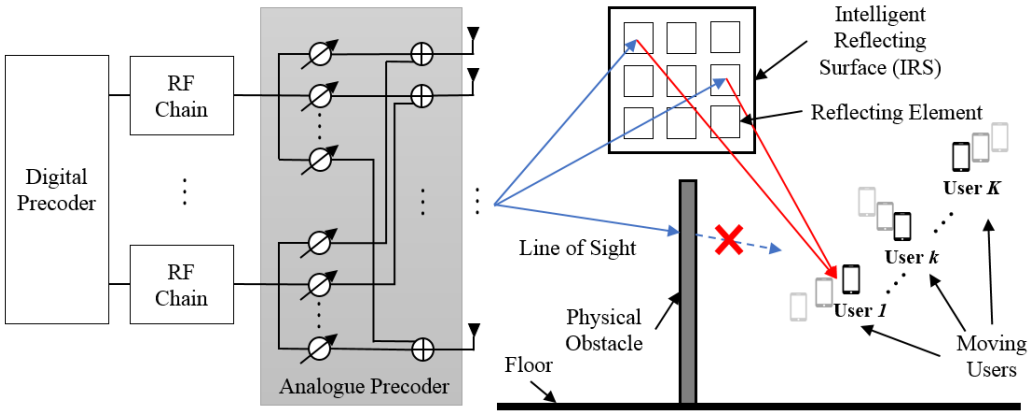

Fig. 3.1 The system model structure where a multi-antenna BS equipped with N_t antennas simultaneously serves K mobile users through an IRS with N reflecting elements.

based prediction-optimisation scheme, in which a deep learning-based channel prediction method is proposed and the CMA-ES-based EE optimisation algorithm is studied. Finally, the simulation results are illustrated in Section 3.4 whilst the conclusions are drawn in Section 3.5.

Notations: $(.)^H$, $(.)^{-1}$ and $\|.\|_F$ represent the conjugate transpose operation, the inversion operation and the Frobenius norm of the matrix, respectively; x, $\mathbf{x}$, $\mathbf{X}$ and $\mathcal{X}$ represent scalar, vector, matrix and high-dimensional tensor, respectively.

3.2 System Model

Fig. 3.1 shows the system model. We consider an IRS-assisted MU-MISO THz system, where a multi-antenna BS equipped with N_t antennas simultaneously serves K mobile users through an IRS with N reflecting elements. According to the analysis [40], THz is extremely sensitive to molecular absorption and atmospheric attenuation. In this case, the THz signals can experience severe path losses, which results in a huge limitation on communication distance and quality. Therefore, only a single reflection signal by the IRS is considered in this work and other multi-reflection signals are all ignored due to the heavy propagation loss for THz waves [23, 40]. Moreover, for simplicity, only one data stream is considered for transmission to each user. Therefore, the received signal vector $\mathbf{y}$ of size $K \times 1$ for all users can be written as:

$$\mathbf{y} = \mathbf{H}_r \mathbf{\Theta} \mathbf{H}_t \mathbf{W} \mathbf{s} + \mathbf{n}, \tag{3.1}$$

where the complex matrix $\mathbf{H}_t = [\mathbf{h}_{t,1}, \mathbf{h}_{t,2}, ..., \mathbf{h}_{t,N}]^H$ with size $N \times N_t$ represents the channel matrix between the BS and the IRS with $\mathbf{h}_{t,n} \in C^{N_t \times 1}$ denoting the channel vector between

the IRS element n and the BS; the complex matrix $\mathbf{H}_r = [\mathbf{h}_{r,1}, \mathbf{h}_{r,2}, ..., \mathbf{h}_{r,K}]^H$ with size $K \times N$ denotes the channel matrix between the IRS and the users with $\mathbf{h}_{r,k} \in C^{N\times 1}$ representing the channel vector between the kth user and the IRS, $k = 1, 2, ..., K$. Moreover, $\mathbf{\Theta} = \text{diag}[e^{j\theta_1}, e^{j\theta_2}, ..., e^{j\theta_N}] \in C^{N\times N}$ is the phase-shift matrix of the IRS, and $\theta_n \in [0, 2\pi)$ are the phase shifts for the nth IRS reflecting element. For practicality of the design, we assume each IRS phase shift θ is chosen from a set of discrete values $\mathcal{F} = \{0, \Delta\theta, ..., \Delta\theta(2^b - 1)\}$ where b is the bit-quantisation number and $\Delta\theta = 2\pi/2^b$. The precoding matrix $\mathbf{W} \in C^{N_t\times K}$ satisfies a transmit power constraint $\|\mathbf{W}\|_F^2 \leqslant P_{\max}$, in which $P_{\max}$ denotes the maximum total transmit power. It is worth noting that the precoding matrix $\mathbf{W} = \mathbf{F}_{\text{RF}}\mathbf{F}_{\text{BB}}$ is a coupled hybrid precoding matrix, where $\mathbf{F}_{\text{RF}} \in C^{N_t\times N_{\text{RF}}}$ and $\mathbf{F}_{\text{BB}} \in C^{N_{\text{RF}}\times K}$ are the analogue precoder matrix and digital precoder matrix with N_{RF} radio frequency (RF) chains, respectively. Furthermore, the vector $\mathbf{n} \in C^{K\times 1}$ denotes the zero-mean and σ^2 variance additive white Gaussian noise (AWGN). The transmission signal vector $\mathbf{s} \in C^{K\times 1}$ meets the condition $\text{E}[\mathbf{s}\mathbf{s}^H] = \mathbf{I}_K$.

A THz indoor multi-path channel model, namely Saleh-Valenzuela model [71], is used to describe the indoor THz channel properties for the channel vector $\mathbf{h}_{t,n}$. In such a design, massive antennas are used to mitigate the severe path loss and molecular absorption problems of THz channels in which only a small number of paths are effective. As a consequence, the channel vector $\mathbf{h}_{t,n}$ is expressed as:

$$\mathbf{h}_{t,n} = \zeta \sum_{l=1}^{L_n} \alpha_n^{(l)}(f, d) \cdot \mathbf{a}(N_t, \phi^{(l)}), \tag{3.2}$$

where N_t is the number of transmit antennas; L_n is the number of paths between the IRS element n and the BS; $\zeta = \sqrt{\frac{N_t}{L_n}}$ is a normalisation factor [72]; $\alpha_n^{(l)}(f, d) = \left|\alpha_n^{(l)}(f, d)\right| e^{j\psi_n^{(l)}}$ represents the complex path gain of the lth ray which is a function of the distance between the IRS and the BS d; $\psi_n^{(l)}$ is the associated independent phase shift which is uniformly distributed over the range of $[0, 2\pi)$ [72], f is the frequency; the array steering vector of an uniform linear array (ULA) is denoted by $\mathbf{a}(N_t, \phi^{(l)})$, which can be further expressed as $\mathbf{a}(N_t, \phi^{(l)}) = \frac{1}{\sqrt{N_t}} \left[e^{j2\pi m(d_a/\lambda)\sin(\phi^{(l)})}\right]^T, m = 1, 2, ..., N_t$, where d_a is the antenna space and λ is the signal wavelength; $\phi^{(l)} \in [0, 2\pi)$ is the angle of departure (AoD) in the horizontal azimuth domain for the path l. Moreover, the space between adjacent reflecting elements d_r is assumed to be much smaller than the distance between the IRS and the BS. In this case, each $\mathbf{h}_{t,n}$ consisted of the same $\mathbf{a}(N_t, \phi^{(l)})$ whilst only one beam pattern from the BS serves the whole elements [23]. Since the THz wave operates at a relatively high frequency, the value of d_r is considered to be much larger than the signal wavelength λ. As a result, the mutual coupling effect is negligible in this work.

The channel response between the nth reflecting element and the kth user $h_{r,n,k}$ is modelled by a modified Clarke's two-dimensional (2D) isotropic scattering Rayleigh fading model [73] since the users are moving objects and Doppler effect on the phase shift should be considered. The channel response $h_{r,n,k}$ at time slot t is given as follows:

$$h_{r,n,k}(t) = \frac{1}{\sqrt{L_n}} \sum_{l=1}^{L_n} e^{j(w_d t \cos a_l + b_l)}, \tag{3.3}$$

where w_d is the maximum radian Doppler frequency; $a_l = \frac{2\pi l + \psi_l}{L_n}$ and b_l are, respectively, the angle of arrival and initial phase of the lth propagation path. Both ψ_l and b_l are uniformly distributed over $[0, 2\pi)$ for all l and they are mutually independent.

Eq. (3.3) shows that even a small user movement in the indoor scenario can result in a large Doppler spread due to the high operating frequency of THz channels. The Doppler spread is inversely proportional to the coherence time of the channel. Hence, for a large Doppler spread, the channel coherence time will be smaller, which in turn entails a lower correlation between a sample at time t and the predicted one at $t + \delta_t$, where δ_t is the length of a time slot. Such a large Doppler spread can cause the CSI to change rapidly over time and make the EE optimisation results far away from optimum since the previously collected CSI becomes outdated in a short time. Therefore, a strong time-series prediction algorithm should be employed to capture the temporal dependency of the channel and predict the CSI in the following time slot in advance.

3.3 Deep Learning-Based Prediction-Optimisation Scheme

In this section, we explain our proposed deep learning-based prediction-optimisation scheme in detail, whose flow chart is illustrated in Fig. 3.2. The proposed scheme consists of two parts, channel prediction and EE optimisation. In channel prediction, the TE-CIE model estimates and collects the CSI of the previous M slots and uses them to predict the CSI of the next slot. The CMA-ES algorithm and Dinkelbach's method are then applied to optimise EE in advance based on the predicted CSI. Accordingly, the BS and IRS can adjust the precoding matrix and phase shift matrix based on the optimisation result.

3.3.1 Deep Learning-Based Time-Varying Fading Channel Prediction

In this section, a TE-CIE deep learning model is developed to predict the CSI matrix between the IRS and the users of the next time slot given the channel responses of the previous M time slots. The input of the model includes the channel index vector $\mathbf{c} \in \mathbb{Z}^M$ and the CSI matrix $\mathbf{X} \in \mathbb{R}^{M \times 2}$ which contains a sequence of the last M CSI values, where the dimension 2 is due to the real part and imaginary part of the CSI. More specifically, $\mathbf{X}$

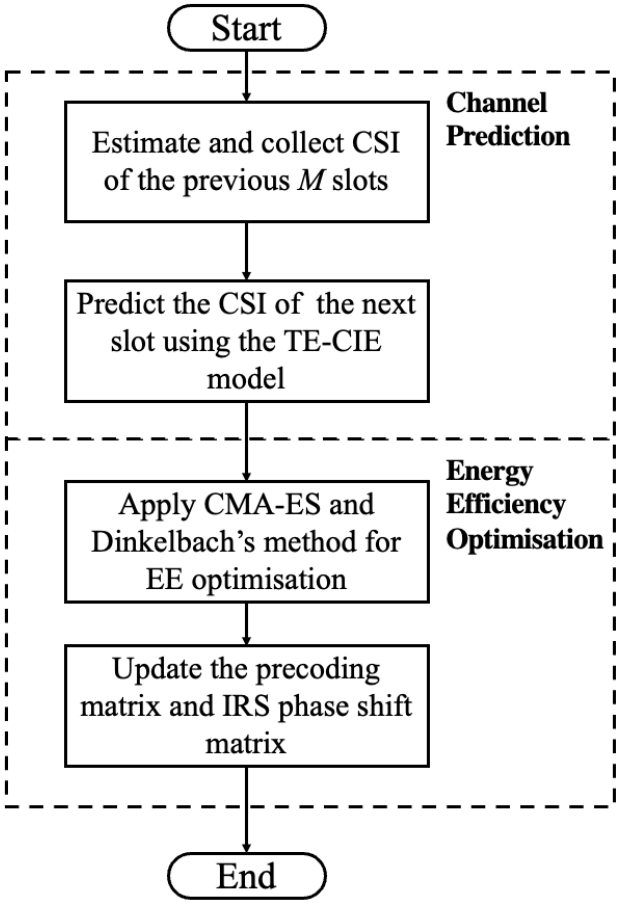

Fig. 3.2 The flow chart of the proposed deep learning-based prediction-optimisation scheme.

is a sequence of length M of the CSI of the channel between the nth IRS element and the kth user, $h_{r,n,k}$, as in Eq. (3), i.e., $[h_{r,n,k}(t-M+1),...,h_{r,n,k}(t-1),h_{r,n,k}(t)]^T$, where the real and imaginary parts are decoupled. The output of the model is a vector containing the real and imaginary parts of the next CSI $\mathbf{y}_{\text{CSI}} \in \mathbb{R}^2$.

TE-CIE Model Architecture

The TE-CIE model has four parts: input embedding layers, multiple stacked transformer encoder layers, feature aggregation layers, and the output layer.

The input embedding layers consist of three layers: the CIE layer, the CSI embedding layer, and the positional embedding layer. These three layers embed the channel index, CSI, and time-step index, respectively, into a high-dimensional matrix.

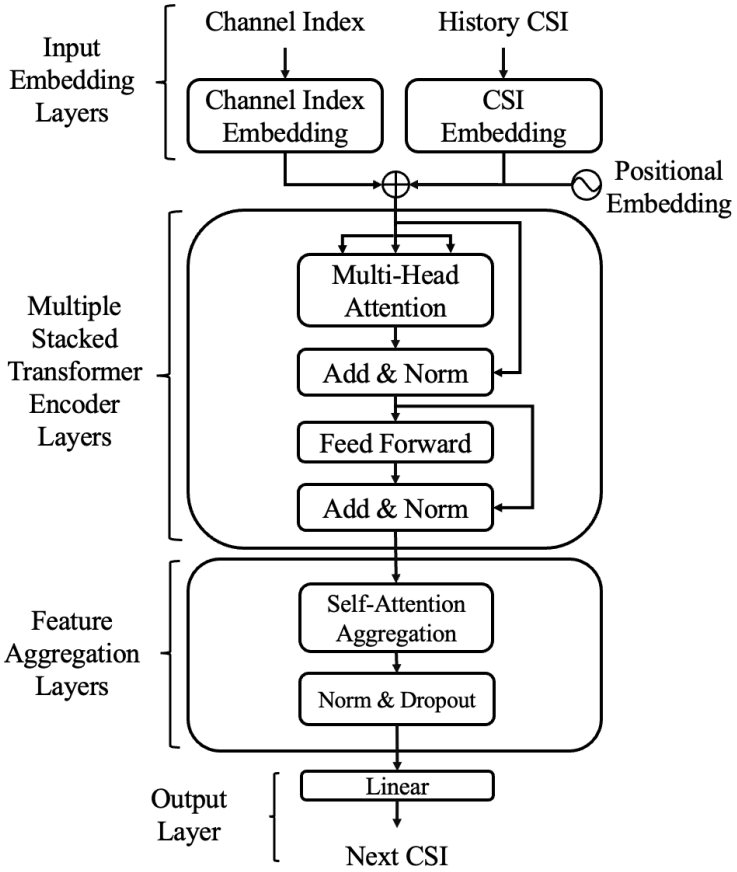

Fig. 3.3 TE-CIE model architecture.

Since there are multiple channels between the IRS and the users (i.e., NK channels) in our MU-MISO-IRS system, we first set an integer index c to each channel, ranging from 0 to $NK-1$. The channel index in our proposed TE-CIE model is used as an additional feature that enables the model to distinguish between different channels. It also allows the model to make predictions for all channels simultaneously without training one model for each channel separately, which significantly reduces the complexity. For each time step, the channel index is unchanged for the same channel. Thus, the channel index vector $\mathbf{c} = \underbrace{[c, c, ..., c]}_{M}$. The channel index embedding layer maps the channel index vector into a channel index matrix with embedding dimension d_{model}. The output of the CIE layer has a shape of $\mathbf{C}_{\text{CIE}} \in \mathbb{R}^{M \times d_{\text{model}}}$, which is calculated as follows:

$$\mathbf{C}_{\text{CIE}} = \text{OneHot}_{\text{CIE}}(\mathbf{c})\mathbf{W}_{\text{CIE}}, \tag{3.4}$$

where $\text{OneHot}_{\text{CIE}}(.)$ refers to the one-hot encoding operation which encodes the index vector to a sparse matrix with dimension $\mathbb{R}^{M\times NK}$; $\mathbf{W}_{\text{CIE}} \in \mathbb{R}^{NK\times d_{\text{model}}}$ is the weight matrix of CIE.

The CSI embedding layer is similar to the CIE layer but the input is the real part and imaginary part of the last M CSI. The CSI embedding layer maps the CSI matrix $\mathbf{X}$ to $\mathbf{C}_{\text{CSI}} \in \mathbb{R}^{M\times d_{\text{model}}}$, whose dimension is the same as that of $\mathbf{C}_{\text{CIE}}$. The output $\mathbf{C}_{\text{CSI}}$ can be expressed as:

$$\mathbf{C}_{\text{CSI}} = \mathbf{X}\mathbf{W}_{\text{CSI}}, \tag{3.5}$$

where $\mathbf{W}_{\text{CSI}} \in \mathbb{R}^{2\times d_{\text{model}}}$ is the weight matrix of CSI embedding.

The positional embedding layer embeds the time-step index to a high-dimension matrix. This is to add position information to the input since the TE-CIE model is not sequential such as RNN. Let $\mathbf{m} = [0, 1, ..., M]$ be the time-step indices of M steps, the output of positional embedding layer $\mathbf{C}_{\text{PE}} \in \mathbb{R}^{M\times d_{\text{model}}}$ is given by:

$$\mathbf{C}_{\text{PE}} = \text{OneHot}_{\text{PE}}(\mathbf{t})\mathbf{W}_{\text{PE}}, \tag{3.6}$$

where $\text{OneHot}_{\text{PE}}(.)$ is the one-hot encoding operation that encodes the time-step index vector to a sparse matrix with dimension $\mathbb{R}^{M\times M}$; $\mathbf{W}_{\text{PE}} \in \mathbb{R}^{M\times d_{\text{model}}}$ is the weight matrix of positional embedding.

The outputs of the CIE layer, CSI embedding layer and positional embedding layer are then added up and passed into the stacked transformer encoder layers. The output of the input embedding layers is:

$$\mathbf{C}_{\text{IE}} = \mathbf{C}_{\text{CIE}} + \mathbf{C}_{\text{CSI}} + \mathbf{C}_{\text{PE}}. \tag{3.7}$$

According to the implementation in the BERT paper in [45], the element-wise addition operation on embedding sequences is equivalent to first concatenating them together and then passing through a fully connected layer, where the simple summing is more efficient since the addition operation has much lower computational complexity than the matrix multiplication operation.

The stacked transformer encoder layers contain N_{TE} Transformer encoders, which are proposed initially in [7]. As shown in Fig. 3.3, each layer has two sub-layers, i.e., a multi-head self-attention mechanism and a position-wise fully-connected feed-forward network. In addition, a residual connection [74] with layer normalisation [75] is applied after each of the two sub-layers.

Let $\mathbf{X}_{\text{last}} \in \mathbb{R}^{M\times d_{\text{model}}}$ denote the output of the last layer, the output of the multi-head attention $\mathbf{Y}_{\text{MHA}} \in \mathbb{R}^{M\times d_{\text{model}}}$ of each TE layer can be expressed as:

$$\mathbf{Y}_{\text{head},i} = \text{softmax}\left(\frac{\mathbf{X}_{\text{last}}\mathbf{W}_i^Q(\mathbf{X}_{\text{last}}\mathbf{W}_i^K)^M}{\sqrt{d_k}}\right)\mathbf{X}_{\text{last}}\mathbf{W}_i^V, \tag{3.8}$$

$$\mathbf{Y}_{\text{MHA}} = \text{Concat}(\mathbf{Y}_{\text{head},1}, ..., \mathbf{Y}_{\text{head},h})\mathbf{W}^{O}, \tag{3.9}$$

where $d_k = d_{\text{model}}/h$ is the depth of each head and h is the number of heads; the projection are parameter matrices $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $\mathbf{W}^O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$; Concat(.) is the concatenation operation which concatenates the output of each head $\mathbf{Y}_{\text{head},i} \in \mathbb{R}^{M \times d_k}$ into a new matrix with dimension $\mathbb{R}^{M \times d_{\text{model}}}$.

In addition to the multi-head attention, each TE layer also applies a fully connected feed-forward network to each position identically and separately. It consists of two linear transformations with a Gaussian Error Linear Unit (GELU) activation function in between. Let $\mathbf{X}_{\text{FFN}} \in \mathbb{R}^{M \times d_{\text{model}}}$ denote the input of the fully-connected feed-forward network, whose output $\mathbf{Y}_{\text{FFN}} \in \mathbb{R}^{M \times d_{\text{model}}}$ is given by:

$$\mathbf{Y}_{\text{FFN}} = \text{GELU}(\mathbf{X}_{\text{FFN}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{3.10}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{model}} \times d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d_{\text{model}}}$, $\mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$ and $\mathbf{b}_2 \in \mathbb{R}^{d_{\text{model}}}$ are projection matrices and bias vectors, respectively; d_{ff} is the inner-layer hidden dimension.

The feature aggregation layer contains a self-attention aggregation layer, a batch normalisation [76] and a dropout layer [77] for regularisation. The self-attention aggregation layer provides a weighted-sum operation to the final output of the stacked transformer encoder layers for both dimensionality reduction and temporal feature aggregation. More specifically, since the final output of the stacked transformer encoder layers is a sequence, it is vital to apply a sequence-to-vector transformation so that only one vector representing the predicted CSI can be obtained.

Let $\mathbf{X}_{\text{STE}} \in \mathbb{R}^{M \times d_{\text{model}}}$ denote the final output of the stacked transformer encoder layers, the output of the self-attention aggregation layer $\mathbf{y}_{\text{SAA}} \in \mathbb{R}^{d_{\text{model}}}$ is given by:

$$\mathbf{Y}_{\text{SA}} = \text{softmax}\left(\mathbf{X}_{\text{STE}}\mathbf{W}_{\text{SA}} + \mathbf{b}_{\text{SA}}\right), \tag{3.11}$$

$$\mathbf{y}_{\text{SAA}} = \sum_{t=1}^{M} \left(\mathbf{x}_{\text{STE},i} \odot \mathbf{y}_{\text{SA},i}\right), \tag{3.12}$$

where $\mathbf{W}_{\text{SA}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ and $\mathbf{b}_{\text{SA}} \in \mathbb{R}^{d_{\text{model}}}$ are the weight matrix and bias vector for the linear transformation of $\mathbf{X}_{\text{STE}}$, respectively; $\odot$ refers to the element-wise multiplication.

The output layer provides a simple linear regression operation to the output of the feature aggregation layer, which is given by:

$$\mathbf{y}_{\text{CSI}} = \hat{\mathbf{y}}_{\text{SAA}}\mathbf{W}_o + \mathbf{b}_{\text{o}}, \tag{3.13}$$

where $\hat{\mathbf{y}}_{\text{SAA}} \in \mathbb{R}^{d_{\text{model}}}$ is the feature aggregation output after passing through the batch normalisation and dropout layers; $\mathbf{W}_o \in \mathbb{R}^{d_{\text{model}} \times 2}$ and $\mathbf{b}_{\text{o}} \in \mathbb{R}^2$ are the weight matrix and bias vector for the output linear regression layer, respectively.

Model Training

In the training phase, the training CSI sequence is split into multiple training samples by applying the sliding window approach [78]. Each sample contains M historical CSI as the input and one CSI of the next slot as the target.

For model parameter optimisation, the widely used Adam optimiser is applied due to its fast convergence and stability [79]. The cosine annealing method [80] is used as the learning rate scheduler for the training. With the i-th run, the scheduled learning rate is given by:

$$\eta_t = \eta_{\min}^i + \frac{1}{2}\left(\eta_{\max}^i - \eta_{\min}^i\right)\left(1+\cos\left(\frac{T_{\text{cur}}}{T_i}\pi\right)\right), \tag{3.14}$$

where $\eta_{\min}$ and $\eta_{\max}$ are minimum and maximum allowed learning rates, respectively; T_{cur} refers to the number of epochs since the last restart.

3.3.2 EE Optimisation Problem

In this section, we explain the proposed EE optimisation algorithms on the predicted CSI of the next time slot. First of all, an EE optimisation problem is formulated for the IRS-assisted THz MU-MISO system. Specifically, to solve the optimisation problem, the zero-forcing method is applied to cancelling the co-channel interference, the Dinkelbach's and Lagrange multiplier method is used to optimise the power allocation, and the CMA-ES algorithm is proposed to optimise the IRS phase-shift matrix.

Problem Formulation

We aim to jointly optimise the IRS phase-shift matrix $\mathbf{\Theta}$ and the hybrid precoding matrix $\mathbf{W}$ based on our system model in this chapter. From Eq. (3.1), the achievable sum-rate R for all users can be written as:

$$R = \sum_{k=1}^{K} \log_2\left(1+\frac{\left|\mathbf{h}_{r,k}^H \mathbf{\Theta} \mathbf{H}_t \mathbf{w}_k\right|^2}{\sum_{i=1,i\neq k}^{K}\left|\mathbf{h}_{r,k}^H \mathbf{\Theta} \mathbf{H}_t \mathbf{w}_i\right|^2+\sigma^2}\right), \tag{3.15}$$

where $\mathbf{w}_k$ is the kth column vector of $\mathbf{W}$. The total system transmit power is expressed as [30]:

$$\mathcal{P}_{\text{total}} = \sum_{k=1}^{K}(\xi p_k + P_{\text{UE},k}) + P_{\text{BS}} + NP_n(b), \tag{3.16}$$

where $p_k = |\mathbf{w}_k|^2$ is the transmit power intended for user k; $\xi := \nu^{-1}$ with ν being the efficiency of the transmit power amplifier; P_{BS} is the total hardware static power consumption at BS; $P_{\text{UE},k}$ is the hardware static power dissipated by user k; $P_n(b)$ represents the power consumption of each IRS phase shifter with b-bit resolution. Moreover, the EE

is defined as the ratio between the system achievable sum rate and the total system power consumption, i.e., $\eta_{\mathrm{EE}} \triangleq R/\mathcal{P}_{\mathrm{total}}$, which is written as:

$$\eta_{\mathrm{EE}} = \frac{\sum_{k=1}^{K} \log_2 \left(1 + \frac{\left|\mathbf{h}_{r,k}^H \boldsymbol{\Theta} \mathbf{H}_t \mathbf{w}_k\right|^2}{\sum_{i=1, i\neq k}^{K} \left|\mathbf{h}_{r,k}^H \boldsymbol{\Theta} \mathbf{H}_t \mathbf{w}_i\right|^2 + \sigma^2}\right)}{\sum_{k=1}^{K} (\xi p_k + P_{\mathrm{UE},k}) + P_{\mathrm{BS}} + NP_n(b)}. \tag{3.17}$$

Then the EE optimisation problem can be formulated as:

$$\begin{aligned} (\boldsymbol{\Theta}^{\mathrm{opt}}, \mathbf{W}^{\mathrm{opt}}) = & \arg\max \eta_{\mathrm{EE}}, \\ \text{s.t.} \quad & \theta_n \in \mathcal{F}, \ \forall n = 1, ..., N, \\ & \|\mathbf{W}\|_F^2 \leqslant P_{\max} \\ & p_k \geqslant 0, \ \forall k = 1, ..., K, \end{aligned} \tag{3.18}$$

where $\|\mathbf{W}\|_F^2 = \sum_{k=1}^{K} p_k = \sum_{k=1}^{K} |\mathbf{w}_k|^2$ is the total transmit power.

Because of the non-convex objective function over two variables $(\boldsymbol{\Theta}^{\mathrm{opt}}, \mathbf{W}^{\mathrm{opt}})$, it is challenging to solve the optimisation problem in Eq. (3.18) directly. Thus, we introduce an iterative method which follows a similar procedure as in [23]. Firstly, we select one phase value for each θ_n from $\mathcal{F}$ successively. Then the candidate phase shift matrix $\boldsymbol{\Theta}$ is constructed. After that, the CMA-ES algorithm is applied to the iterative optimisation of the phase-shift matrix $\boldsymbol{\Theta}$. Meanwhile, with a given $\boldsymbol{\Theta}$ in each iteration, we can obtain the optimal $\mathbf{W}$ with the effective channel matrix, i.e., $\mathbf{H}_{\mathrm{eq}} = \mathbf{H}_r \boldsymbol{\Theta} \mathbf{H}_t$, through the Dinkelbach's method and zero-forcing algorithm.

CMA-ES Algorithm

CMA-ES [70] is an iterative evolution strategy algorithm, in which the next generation's population is sampled from a multivariate normal distribution over a covariance matrix. The CMA-ES algorithm continuously chooses the best local individual to enhance the population's fitness. The main advantages of the CMA-ES algorithm are its rotational invariance, fast converging rate and its ability to efficiently solve the optimisation problem by only sampling a few data points.

The kth offspring's coordinates at the $(g+1)$th generation $\mathbf{x}_k^{(g+1)}$ are drawn from Eq. (3.19) where λ denotes the population size; the mean value of the distribution $\mathbf{m}^{(g)} \in \mathbb{R}^n$, the step size $\sigma^{(g)}$, and the covariance matrix $\mathbf{C}^{(g)} \in \mathbb{R}^{n\times n}$ are computed as Eqs. (3.20)-(3.22), where $c_m \leqslant 1$ is the learning rate, usually set to 1; μ is parent population size; c_1 and c_μ are the learning rates of the rank-one and rank-μ updates, respectively; $\delta(h_\sigma) \in 0, 1$ is a parameter of minor correction for unusual conditions; $\mathbf{x}_{i:\lambda}^{(g+1)}$ indicates the ith best individual in the population of size λ and the positive recombination weights ω is subject

$$\mathbf{x}_k^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(0, \mathbf{C}^{(g)}), \qquad k = 1, ..., \lambda, \tag{3.19}$$

$$\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + c_m \sum_{i=1}^{\mu} \omega_i \left(\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)} \right), \tag{3.20}$$

$$\mathbf{C}^{(g+1)} = c_\mu \underbrace{\sum_{i=1}^{\lambda} \omega_i \mathbf{y}_{i:\lambda}^{(g+1)} \mathbf{y}_{i:\lambda}^{(g+1)^\top}}_{\text{rank-}\mu\text{ update}} + (1 \underbrace{- c_1 - c_\mu \sum \omega_j}_{\text{can be close or equal to 0}}) \mathbf{C}^{(g)} + c_1 \underbrace{\mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^\top}}_{\text{rank-one update}}, \tag{3.21}$$

$$\ln \sigma^{(g+1)} = \frac{c_\sigma}{d_\sigma} \left(\frac{\| \mathbf{p}_c^{(g+1)} \|}{\mathrm{E} \| \mathcal{N}(\mathbf{0}, \mathbf{I}) \|} - 1 \right) + \ln \sigma^{(g)}, \tag{3.22}$$

Algorithm 1 CMA-ES with Dinkelbach's method for EE optimisation (outer loop)

1: Define $I_{\max}$ as the maximum number of iterations
2: Initialise $\mathbf{m}^0$, $\mathbf{C}^0$ and σ^0
3: **for** $i = 0, ..., I_{\max}$ **do**
4: Randomly sample λ candidates $\{\mathbf{\Theta}^k\}_{k=1}^{\lambda}$ using $\mathbf{m}^i$, $\mathbf{C}^i$ and σ^i according to Eq. (3.19)
5: Calculate the precoding matrices $\{\mathbf{W}^k\}_{k=1}^{\lambda}$ for all candidates using Dinkelbach's method (middle loop)
6: Compute the EE values $\{\eta_{\mathrm{EE}}(\mathbf{\Theta}^k)\}_{k=1}^{\lambda}$ for all candidates
7: Rank $\{\eta_{\mathrm{EE}}(\mathbf{\Theta}^k)\}_{k=1}^{\lambda}$ in a descending order as $\eta_{\mathrm{EE}}(\mathbf{\Theta}^{(1)}) \geqslant \eta_{\mathrm{EE}}(\mathbf{\Theta}^{(2)}) \geqslant ... \geqslant \eta_{\mathrm{EE}}(\mathbf{\Theta}^{(\lambda)})$
8: Update $\mathbf{m}^{i+1}$, $\mathbf{C}^{i+1}$ and σ^{i+1} according to Eqs. (3.20) to (3.22)
9: **end for**
10: Output $\eta_{\mathrm{EE}}^{\max}$, $\mathbf{W}^{\mathrm{opt}}$ and $\mathbf{\Theta}^{\mathrm{opt}}$

to $\sum \omega_i = 1$; $\mathbf{p}_\sigma$ is the conjugated evolution path used in cumulative step-length adaptation (CSA); $\mathbf{p}_c$ represents the evolution path; the factor c_σ is the learning rate for step size update; $d_\sigma \approx 1$ is a damping parameter; $\mathbf{y}_{i:\lambda} = (\mathbf{x}_{i:\lambda} - \mathbf{m}^{(g)})/\sigma^{(g)}$ is the standardised coordinates; $\mathrm{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|$ means the expectation of the Euclidean norm of the isotropic multivariate normal distribution. It is worth noting that the CMA-ES algorithm updates the covariance matrix by ranking all the sample points (i.e., IRS phase-shift matrix sets) based on the evaluation scores (i.e., the EE values) in each iteration.

The procedure of the proposed EE optimisation algorithm is presented in Algorithm 1. First of all, we successively select one phase value for each IRS element and then construct the candidate phase-shift matrix. The optimal precoding matrix can be calculated by Dinkelbach's method with a given phase-shift matrix. After that, we compare all the EE values calculated by the constructed phase-shift matrices and the corresponding precoding

Algorithm 2 Dinkelbach's method for EE maximisation (middle loop)

Define $I_{\max}$ as the maximum number of iterations
Define $\varepsilon > 0$ as the convergence tolerance
Define $\mathcal{P} = \{p_1, ..., p_K\}$ as the power set
Initialise $q_0 \leftarrow 1$
Initialise $i \leftarrow 0$
for $i = 0, ..., I_{\max}$ **do**
  $i \leftarrow i + 1$
  Solve $\max_{\mathcal{P}} R(\mathcal{P}) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P})$ to obtain the optimal power set $\mathcal{P}^{\text{opt}} = \left\{p_1^{\text{opt}}, ..., p_K^{\text{opt}}\right\}$ (inner loop)
  if $|R(\mathcal{P}^{\text{opt}}) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P}^{\text{opt}})| < \varepsilon$ **then**
    break
  else
    $q_i \leftarrow R(\mathcal{P}^{\text{opt}})/\mathcal{P}_{\text{total}}(\mathcal{P}^{\text{opt}})$
  end if
end for

matrices. Finally, the optimal EE value with the precoding matrix and phase-shift matrix are obtained simultaneously.

Dinkelbach's and Lagrange Multiplier Methods

In this step, we first utilise the zero-forcing method to cancel the co-channel interference from the undesirable signals, such that Eq. (3.15) becomes:

$$\begin{aligned} R &= \sum_{k=1}^{K} \log_2 \left(1 + \frac{\left| \mathbf{h}_{r,k}^H \mathbf{\Theta} \mathbf{H}_t \mathbf{w}_k \right|^2}{\sigma^2} \right) \\ &= \sum_{k=1}^{K} \log_2 \left(1 + \frac{p_k |h_k|^2}{\sigma^2} \right), \end{aligned} \tag{3.23}$$

where $h_k = \mathbf{h}_{r,k}^H \mathbf{\Theta} \mathbf{H}_t \frac{\mathbf{w}_k}{|\mathbf{w}_k|}$, which makes the numerator of the EE function convex in the powers. In this case, the EE optimisation problem becomes a fractional optimisation problem. Obviously, both the denominator and numerator of η_{EE} are convex. Hence, this problem can be solved by jointly applying the Dinkelbach and Lagrangian multiplier methods.

The Dinkelbach's method, proposed in [81], appears to be an efficient iterative algorithm for solving fractional programming problems with a convex numerator and denominator, whose general process is listed in Algorithm 2. In each iteration, Dinkelbach's method introduces a new convex optimisation problem, which is given by:

Algorithm 3 Lagrange multiplier method with binary search (inner loop)

Define $J_{\max}$ as the maximum number of iterations for binary search
Define $\varepsilon > 0$ as the convergence tolerance
Define $\mathcal{P}_1 = \left\{p_{1,1}, ..., p_{K,1}\right\}$ as the power set satisfying the condition in Eq. (3.28)
Define $\mathcal{P}_2 = \left\{p_{1,2}, ..., p_{K,2}\right\}$ as the power set satisfying the condition in Eq. (3.29)
Initialise $j \leftarrow 0$
Initialise $\mu_{\min} \leftarrow 0$
Initialise $\mu_{\max} \leftarrow \frac{P_{\max} + \sum_{k=1}^{K} \frac{\sigma^2}{|h_k|^2}}{K}$
while $\mu_{\max} - \mu_{\min} > \varepsilon$ **and** $j < J_{\max}$ **do**
    $\mu_j \leftarrow \frac{\mu_{\min} + \mu_{\max}}{2}$
    Obtain $\mathcal{P}_1$ and $R(\mathcal{P}_1) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P}_1)$ by solving Eq. (3.27) using μ_j
    if $\sum_{k=1}^{K} p_{k,1} > P_{\max}$ **then**
        $\mu_{\max} \leftarrow \mu_j$
    else
        $\mu_{\min} \leftarrow \mu_j$
    end if
    $j \leftarrow j + 1$
end while
Obtain $\mathcal{P}_2$ and $R(\mathcal{P}_2) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P}_2)$ by solving Eq. (3.27) using $\lambda = 0$
if $R(\mathcal{P}_2) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P}_2) > R(\mathcal{P}_1) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P}_1)$ **and** $\sum_{k=1}^{K} p_{k,2} \leqslant P_{\max}$ **then**
    $\mathcal{P}^{\text{opt}} \leftarrow \mathcal{P}_2$
else
    $\mathcal{P}^{\text{opt}} \leftarrow \mathcal{P}_1$
end if

$$\begin{aligned} &\max_{\mathcal{P}} R(\mathcal{P}) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P}) \\ &\text{s.t.} \sum_{k=1}^{K} p_k \leqslant P_{\max} \\ &\quad p_k \geqslant 0, \ \forall k = 1, ..., K. \end{aligned} \tag{3.24}$$

Then consider the following expression:

$$\mathcal{L}(\lambda, \mathcal{P}) = R(\mathcal{P}) - q_{i-1}\mathcal{P}_{\text{total}}(\mathcal{P}) + \lambda \left(\sum_{k=1}^{K} p_k - P_{\max} \right), \tag{3.25}$$

where $\lambda \geqslant 0$ is the Lagrange multiplier. The Kuhn-Tucker condition for the optimal solution is:

$$\left.\begin{array}{ll} \frac{\partial \mathcal{L}(\lambda, \mathcal{P})}{\partial p_k} = 0 & \text{if } p_k > 0 \\ \frac{\partial \mathcal{L}(\lambda, \mathcal{P})}{\partial p_k} \leqslant 0 & \text{if } p_k = 0 \end{array}\right\}, \ \forall k = 1, ..., K. \tag{3.26}$$

Define $x^+ := \max(x, 0)$ as the ramp function, the optimal power allocation for kth user at ith iteration can then be expressed as:

$$p_k^{\text{opt}} = \left(\frac{1}{\ln 2 \left(q_{i-1} \cdot \xi - \lambda \right)} - \frac{\sigma^2}{|h_k|^2} \right)^+, \forall k = 1, ..., K, \tag{3.27}$$

which is the optimal solution if the Lagrange multiplier λ satisfies either of the following two conditions:

$$\begin{aligned} P_{\max} &= \sum_{k=1}^{K} \left(\frac{1}{\ln 2 \left(q_{i-1} \cdot \xi - \lambda \right)} - \frac{\sigma^2}{|h_k|^2} \right)^+ \\ &\geqslant \sum_{k=1}^{K} \left(\frac{1}{\ln 2 \left(q_{i-1} \cdot \xi - \lambda \right)} - \frac{\sigma^2}{|h_k|^2} \right), \end{aligned} \tag{3.28}$$

or

$$\lambda = 0. \tag{3.29}$$

For the condition in Eq. (3.28), let $\mu = \frac{1}{\ln 2 (q_{i-1} \cdot \xi - \lambda)}$, we can obtain the upper bound of μ, which is given by:

$$\mu \leqslant \frac{P_{\max} + \sum_{k=1}^{K} \frac{\sigma^2}{|h_k|^2}}{K}. \tag{3.30}$$

The Lagrange multiplier λ satisfying the condition in Eq. (3.28) and Eq. (3.30) can then be found using the binary search method.

The optimal power set $\mathcal{P}$ of the ith iteration is the power set that gives a higher value of the objective function with λ satisfying one of the conditions in Eqs. (3.28) and (3.29), which follows the processes in Algorithm 3.

After that, the optimal kth vector $\mathbf{w}_k^{\text{opt}}$ can be computed with the optimal power p_k^{opt}, which is given by:

$$\mathbf{w}_k^{\text{opt}} = \frac{\mathbf{w}_k}{|\mathbf{w}_k|} \cdot \sqrt{p_k^{\text{opt}}}, \ \forall k = 1, ..., K. \tag{3.31}$$

This way, we have found the optimal precoding matrix $\mathbf{W}^{\text{opt}}$ that cancels the co-channel interference and optimises the EE given a specific phase shift matrix $\boldsymbol{\Theta}$ by jointly applying zero-forcing and Dinkelbach's methods.

3.3.3 Complexity Analysis

The complexity analysis for our proposed optimisation scheme can be divided into two parts: deep learning and optimisation algorithm.

For the deep learning complexity analysis, we compare the proposed TE-CIE model with two baseline models, RNN and MLP, respectively. The complexity of transformer models is dominated by that of self-attention operation which has a theoretical complexity of $O(M^2 d_{\text{model}})$ [7], where M is the sliding window size and d_{model} is the embedding dimension. For RNN and MLP, the complexities of predicting a single channel are given by $O(M d_{\text{model}}^2)$ and $O(M d_{\text{model}})$, respectively. Since there are NK channels, the complexities of both RNN and MLP should be multiplied by NK, which results in $O(NKM d_{\text{model}}^2)$ and

$O(NKMd_{\text{model}})$, respectively. However, since our proposed TE-CIE model can inherently predict all channels in parallel by assigning a channel index to each channel, its complexity remains $O(M^2 d_{\text{model}})$ for predicting NK channels. As a result, our proposed TE-CIE model has much lower complexity than RNN and MLP for predicting multiple channels.

For the optimisation algorithm part, we derive the complexity for the outer, middle and inner loops separately. The outer loop is based on the CMA-ES algorithm, which iterates $I_{\max}$ times with sampling λ candidates at each iteration. The ranking operation of EE values for all candidates can be done with a complexity of $O(\lambda \log_2(\lambda))$. Therefore, the complexity for the outer loop is $O(I_{\max}\lambda(1+\log_2(\lambda)))$. It should be noted that we use the same number of iterations and candidates for the CE method in [23] for a fair performance comparison, i.e., the complexity is the same for CMA-ES and CE methods in our simulation. The RS method has a complexity of $O(1)$ since it contains no loop and randomly chooses the angles within the feasible solution space. The LS method has a complexity of $O(N2^b)$ since it loops over all reflecting elements and exhaustively searches for the best angle among all 2^b candidate solutions. The middle loop is based on Dinkelbach's method with a convergence tolerance ε. Since there are K variables, the complexity of the middle loop can be derived as $O(\sqrt{K}\log_2(1/\varepsilon))$. Similarly, the inner loop with the convergence tolerance ε has a complexity of $O(\log_2(1/\varepsilon))$. The overall complexity of the optimisation algorithm is $O(I_{\max}\lambda\sqrt{K}\log_2^2(1/\varepsilon)(1+\log_2(\lambda))$.

As a result, the overall complexity of our proposed algorithm is expressed as:

$$O(M^2 d_{\text{model}} + I_{\max}\lambda\sqrt{K}\log_2^2(1/\varepsilon)(1+\log_2(\lambda)). \tag{3.32}$$

The overall complexity is lower than the existing methods, e.g., RNN or MLP for channel prediction and the CE method for EE optimisation, since the proposed deep learning-based channel prediction method has lower complexity than MLP and RNN baselines and the EE optimisation algorithm maintains the same complexity as the CE method. Moreover, the proposed optimisation algorithm is guaranteed to converge since the CMA-ES algorithm [70], Dinkelbach's method [81] and binary search [82] algorithms are all guaranteed to converge with finite objective function space.

3.4 Simulation Result

The simulation results are illustrated in this section. First, we illustrate the performance comparison between our proposed TE-CIE model and the baseline methods, namely MLP, RNN and the persistence model. Then we compare the EE optimisation performance of our proposed CMA-ES and Dinkelbach's method with three baseline algorithms in the literature: the RS method, LS method [23] and CE method [23]. Finally, we present the

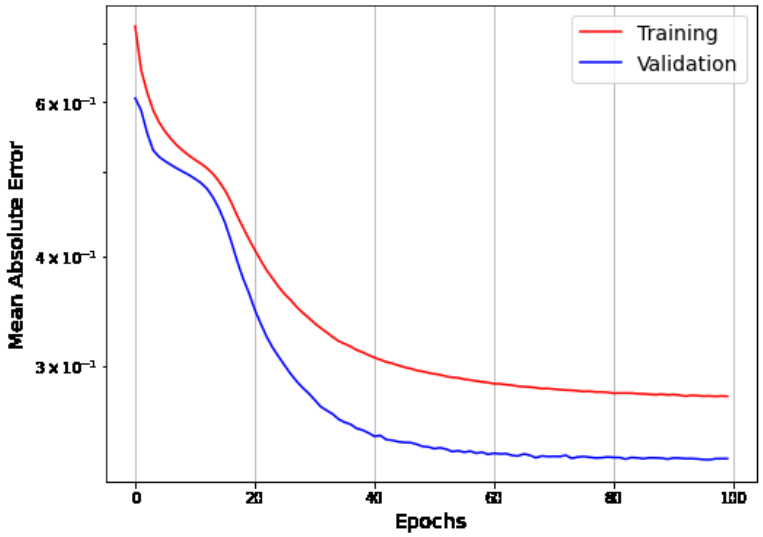

Fig. 3.4 The training curve of the TE-CIE model where the red curve represents the decrease of loss on the training set whilst the blue curve represents the decrease of loss on the validation set.

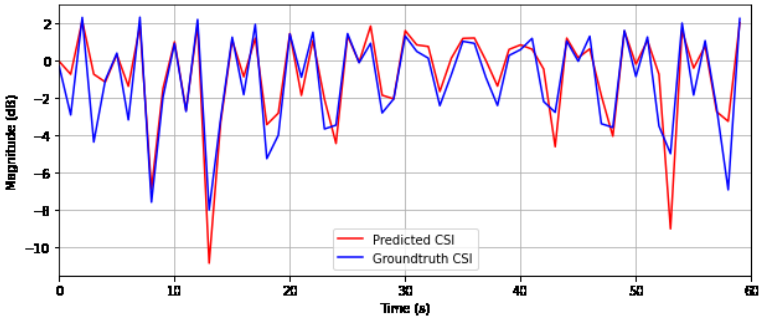

Fig. 3.5 The magnitude comparison in dB between predicted CSI and groundtruth CSI by our proposed TE-CIE model for one channel.

simulation comparison between our deep learning-based prediction-optimisation scheme and the existing methods in the literature.

3.4.1 Channel Prediction

For channel prediction, considering a specific channel estimation method may result in a loss of generality. Hence, we assume the channel estimation error follows complex Gaussian distribution $\mathcal{CN}(0, 0.1)$. All models are trained on CSI sequences with channel estimation error (i.e., the train set) and evaluated on error-free groundtruth CSI (i.e., the test set). We also assume the number of users $K = 4$; the number of IRS elements $N = 64$; the THz carrier frequency $f_{\text{THz}} = 220 \times 10^9$ Hz; Each user's speed is uniformly initialised between 0 to 1 m/s for low-mobility applications.

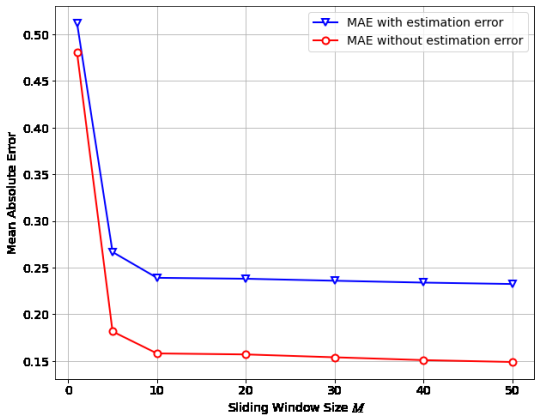

Fig. 3.6 Mean absolute error versus sliding window size M comparison of the TE-CIE model. The blue and red curves show the MAE loss comparison between the model's predicted CSI and the estimated CSI and the groundtruth CSI, respectively.

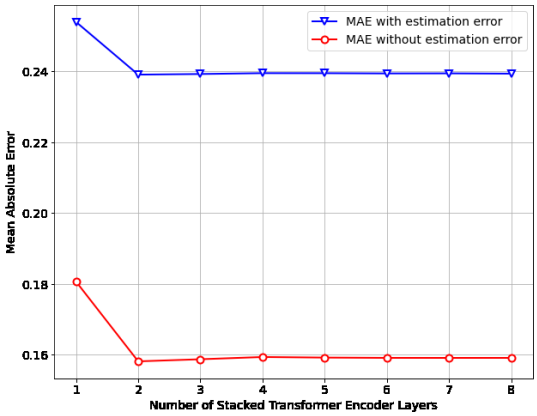

Fig. 3.7 Mean absolute error versus the number of stacked transformer encoder layers comparison of the TE-CIE model. The blue and red curves show the MAE loss comparison between the model's predicted CSI and the estimated CSI and the groundtruth CSI, respectively.

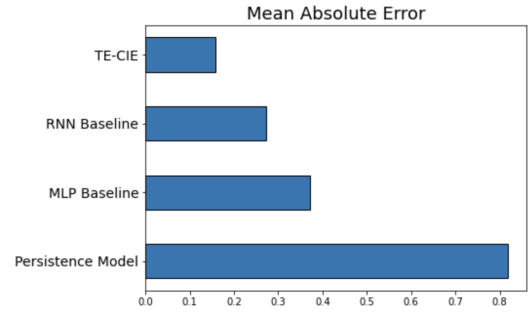

Fig. 3.8 Mean absolute error comparison between TE-CIE, persistence model baseline, MLP baseline and RNN baseline.

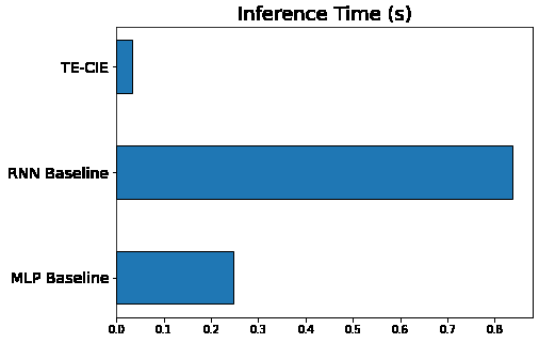

Fig. 3.9 The time comparison of predicting NK channels between TE-CIE, MLP baseline and RNN baseline.

Our training and testing datasets are both simulated using our own system model. Based on Eq. (3.3), we generate a CSI sequence with 2000 time steps for each IRS-user path (i.e., $NK = 256$ paths in total) and use the first 1000 time steps for training and the remaining 1000 time steps for testing. The sliding window size M is set to 10. Therefore, there are $(1000 - M)NK = 253440$ training samples and $1000NK = 256000$ testing samples.

We apply the commonly used MSE loss function for training and MAE loss for evaluation. The MSE loss is differentiable and penalises large errors more than the non-differentiable MAE loss, which makes it a better choice as a loss function. However, the MAE loss measures the same unit as the dependent variable (i.e., the CSI). The MAE loss curves on both training and testing sets are shown in Fig. 3.4.

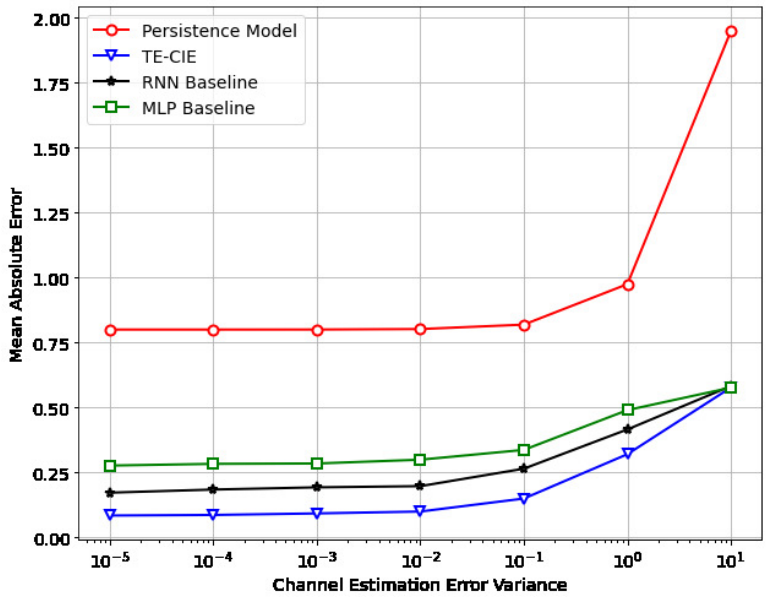

Fig. 3.10 Mean absolute error versus the variance of channel estimation error comparison among TE-CIE, persistence model baseline, MLP baseline and RNN baseline.

For the TE-CIE model, the embedding dimension $d_{\text{model}} = 32$; the number of stacked transformer encoder layers is set to be 2; the number of heads in multi-head attention layers is set to be 8; the dimension of the feed-forward layer $d_{ff} = 64$; the dropout rate equals to 0.1. For training, we use a large batch size of 4096. The maximum number of epochs is set to be 100 for a single-cycle cosine annealing scheduler with the learning rate ranging between 10^{-4} and 10^{-3}. The MLP and RNN models use the same hyperparameters as in [66] and [67]. It is worth noting that both MLP and RNN models also use our proposed CIE method for a fair comparison, which is different from the original implementation in the literature.

Fig. 3.5 shows the magnitude comparison in dB between predicted CSI and groundtruth CSI by our proposed TE-CIE model for one channel. Our proposed TE-CIE model can predict the CSI of the next slot with minor errors. We convert the linear magnitude to decibels here since the decibel form can express magnitude in a more manageable scale over a wide range of values.

Fig. 3.6 shows the MAE loss versus sliding window size M comparison for the proposed TE-CIE model. The blue and red curves show the MAE loss comparison between the model's predicted CSI and the estimated CSI and the groundtruth CSI, respectively. It is evident that the MAE loss decreases dramatically with the increase of sliding window size M at the beginning, and then converges after $M = 10$. This indicates that the past 10 slots of CSI are sufficient to provide a relatively good approximation for the current CSI.

Fig. 3.7 illustrates the MAE loss versus the number of stacked transformer encoder layers for the proposed TE-CIE model. From this figure, we can see that two stacked layers are sufficient. Although adding more layers provides the model with a larger representation capacity, it also leads to overfitting. Moreover, adding more stacked layers may result in the 'gradient vanishing' problem which makes the model harder to train.

Fig. 3.8 shows the MAE loss comparison between TE-CIE, MLP baseline, RNN baseline and the persistence model baseline. The persistence model baseline calculates the future CSI assuming that nothing changes between the current slot and the next slot. It simply uses the current estimated CSI to predict the CSI of the next time slot. The figure indicates that our proposed TE-CIE model has a much lower MAE compared to the other three baseline models. The RNN model shows better performance in terms of lower MAE than MLP while the persistence model baseline has the worst performance.

Fig. 3.9 shows the time comparison of predicting NK channels between TE-CIE, MLP baseline and RNN baseline. It is evident from Fig. 3.9 that our proposed TE-CIE model has the lowest inference time compared with the other two baseline models. The RNN model consumes a longer inference time than MLP and TE-CIE models due to its sequential input nature.

Fig. 3.10 compares the MAE versus the variance of channel estimation error among TE-CIE, persistence model baseline, MLP baseline and RNN baseline. The variance of channel estimation error is measured in a fairly wide range, i.e., 10^{-5} to 10, to ensure that the performances of all possible types of channel estimation methods in the literature are covered. From this figure, we can see that the prediction error increases with the increase of channel estimation error for all models. This indicates that having an accurate channel estimation method can improve channel prediction performance. Among all methods, our proposed TE-CIE model can maintain the best prediction performance in terms of the lowest MAE. When the variance of channel estimation error is extremely large such as 10, all deep learning-based methods tend to have the same prediction performance since the CSI is dominated by the channel estimation error in this case. Nevertheless, all deep learning-based methods still outperform the non-prediction-based persistence model.

3.4.2 Energy Efficiency Optimisation

For EE optimisation, we assume that perfect channel estimation and full channel information are available to all the methods for a fair comparison. The total number of effective rays L_n for each IRS element is assumed to be 3, the antenna space $d_a = 0.68$ mm, the bit quantisation $b = 2$ and the signal wavelength is 1.36 mm. In addition, the circuit power consumption coefficient ξ is assumed to be 1.2, the hardware static power consumption at BS $P_{\mathrm{BS}} = 9$ dBW, the power consumption $P_n(b)$ at the IRS element n is 10 dBm and the

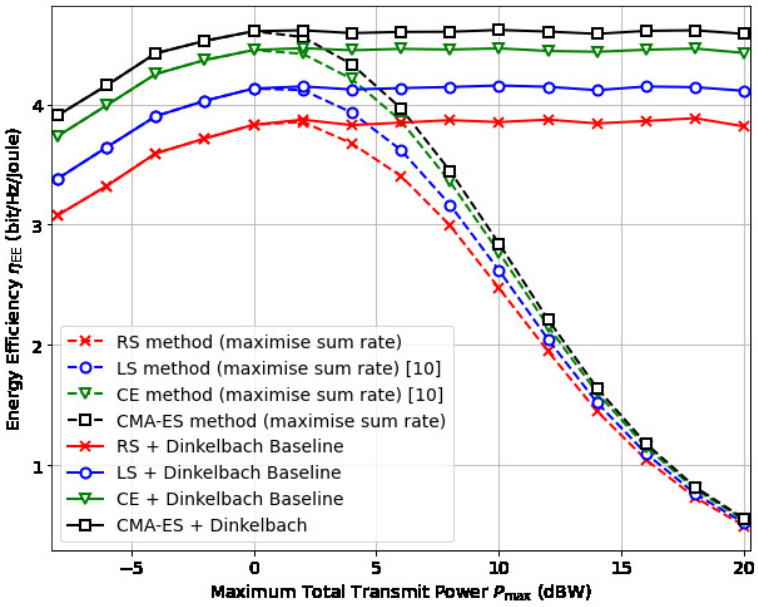

Fig. 3.11 The EE versus maximum total transmit power comparison with 64 IRS reflecting elements, 4 users and 16 transmission antennas.

hardware static power consumption P_{UE} at each user is 10 dBm. Moreover, the number of IRS elements N is assumed to be 64, the number of users $K = 4$, and the number of transmission antennas $N_t = 16$.

We consider two cases for the simulation comparison. The first case is to maximise the system sum rate whilst the second case is about using Dinkelbach's method for EE optimisation. All our results are averaged from 100 randomly realised channels. Since only Case 1 is studied in [23], it is worth noting that we add Dinkelbach's method to the three baselines only to fairly compare our proposed method with baselines and show its effectiveness.

Fig. 3.11 shows the EE versus maximum total transmit power comparisons between our proposed method and the three baseline methods in [23]. From Fig. 3.11, we can see that the EE values of all methods in the first case decrease significantly after reaching the maximum value while the EE values of Case 2 improve with the maximum total transmit power P_{max} for low P_{max} values and reaches the maximum point for high P_{max} values. From such observation, we can see that Dinkelbach's method can effectively improve the EE of the system by selectively using parts of the maximum total transmit power. In addition, the EE values of the proposed CMA-ES-based method are always the highest compared with those of the three baselines. This indicates that the CMA-ES algorithm is a more efficient and effective non-convex optimisation method than the three baseline methods since the correlation between the phase shift of each IRS reflecting element is considered by using a global multivariate normal distribution to model their distributions. Moreover, with the

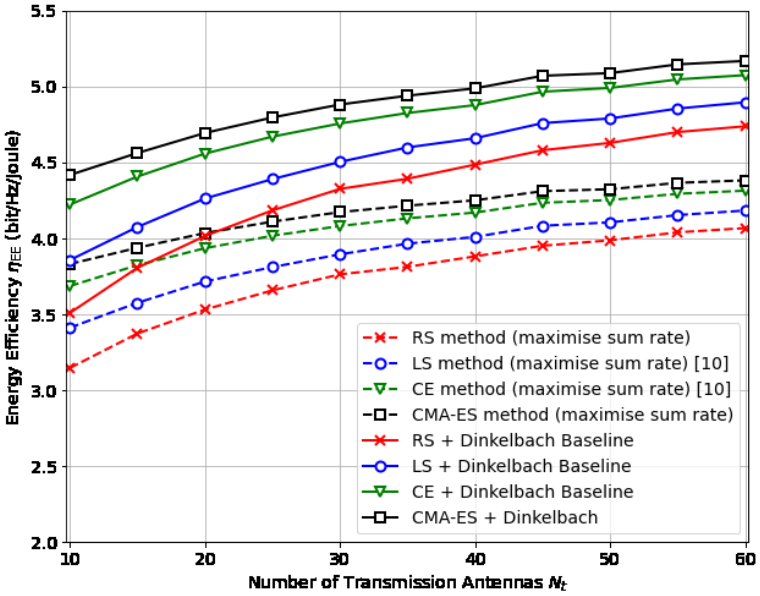

Fig. 3.12 The EE versus the number of transmission antennas comparison with 64 IRS reflecting elements, 4 users and 6 dBW maximum total transmit power.

same number of iterations, the CMA-ES algorithm converges faster than the CE method proposed in [23] by adjusting the step size. To summarise, Fig. 3.11 proves it is effective to combine the CMA-ES algorithm with Dinkelbach's method to jointly optimise the IRS phase-shift matrix and the precoding matrix of the BS.

Figs. 3.12 and 3.13 shows the EE versus N_t and N comparisons. Both figures show that using more reflecting elements helps improve the system beamforming performance. Meanwhile, it is highlighted that the EE values increase with N and N_t due to the increase of transmission antenna gain with the growing number of N_t. We can see from both figures that our proposed EE optimisation algorithm provides the best EE compared to the three baseline methods with any number of N and N_t.

3.4.3 Deep Learning-Based Channel Prediction and Energy Efficiency Optimisation

For the deep learning-based prediction-optimisation part, we assume the same channel estimation error as in the channel prediction part. We compare our proposed method with MLP, RNN and persistence model-based EE optimisation methods.

Fig. 3.14 shows the EE versus maximum total transmit power $P_{\max}$ comparisons between our proposed method, MLP, RNN and persistence model methods. For a fair comparison, all methods use CMA-ES and Dinkelbach's method for EE optimisation. This figure illustrates that our proposed method outperforms the three baselines in terms of EE

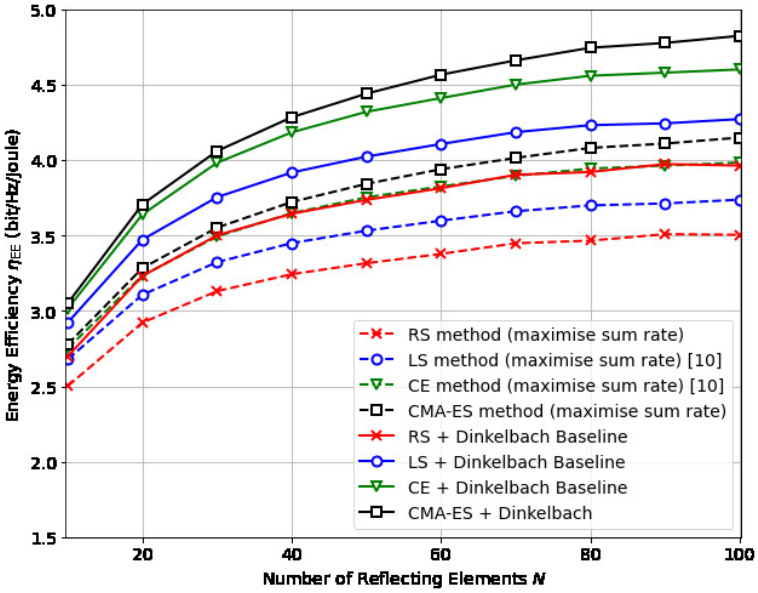

Fig. 3.13 The EE versus the number of reflecting elements comparison with 6 dBW maximum total transmit power, 4 users and 16 transmission antennas.

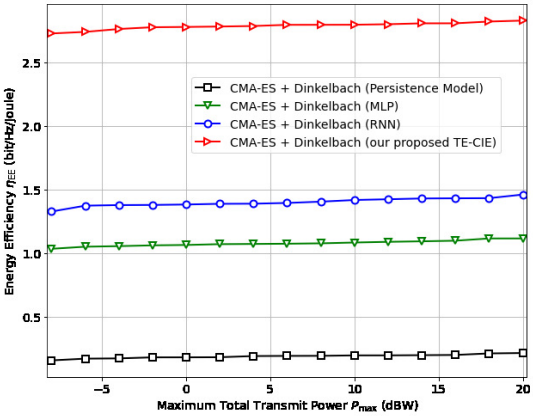

Fig. 3.14 The EE versus maximum total transmit power comparisons between our proposed method, MLP, RNN and persistence model baselines with 64 IRS reflecting elements, 4 users and 16 transmission antennas.

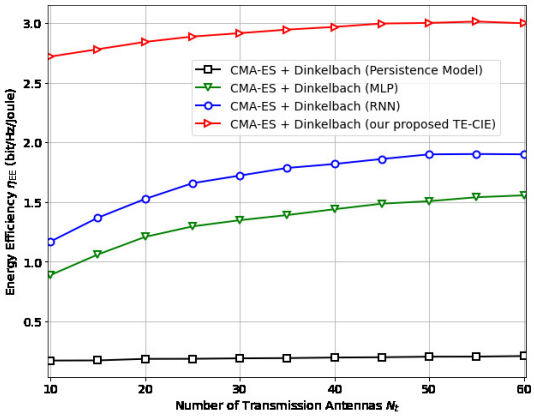

Fig. 3.15 The EE versus the number of transmission antennas comparisons between our proposed method, MLP, RNN and persistence model baselines with 16 IRS reflecting elements, 4 users and 6 dBW maximum total transmit power.

for any value of maximum total transmit power. Specifically, it outperforms RNN, MLP and persistence model-based prediction methods by approximately 2 times, 2.5 times and 15 times higher EE, respectively. By jointly analysing Fig. 3.11 and Fig. 3.14, we can see that using the previous slot's CSI alone for EE optimisation (i.e., the persistence model method) is not suitable for moving user scenarios since the EE drops up to 20 times for

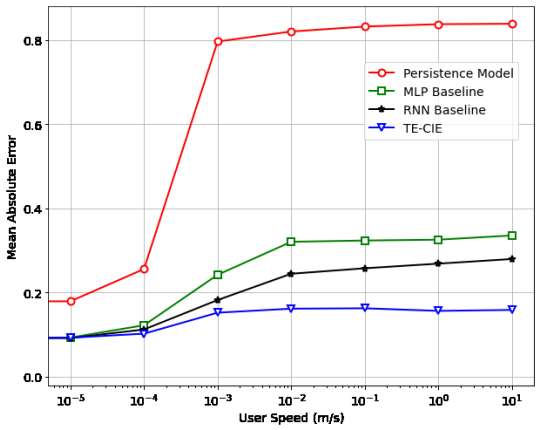

Fig. 3.16 The MAE versus user speed comparisons between our proposed method, MLP, RNN and persistence model baselines.

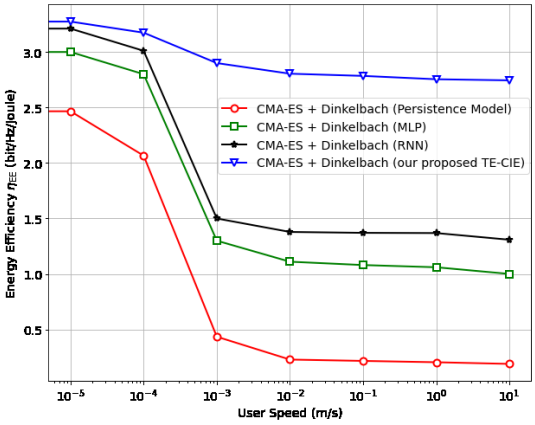

Fig. 3.17 The EE versus user speed comparisons between our proposed method, MLP, RNN and persistence model baselines with 16 IRS reflecting elements, 4 users, 6 dBW maximum total transmit power and 16 transmission antennas.

this non-prediction-based method. This also indicates the necessity of channel prediction for moving users on the THz channel.

Fig. 3.15 illustrates the EE versus the number of transmission antennas N_t comparisons between our proposed method, MLP, RNN and persistence model methods. The same conclusion can be drawn that with the help of accurate channel prediction, the EE can be improved by at least 2 times compared to the baseline methods. Moreover, the EE values of all three deep learning-based methods (i.e., MLP, RNN and TE-CIE methods) increase with N_t, whereas the EE values of the persistence model method remain the same for any values of N_t. This indicates that increasing N_t brings no improvement on EE when the CSI used for EE optimisation differs a lot from the groundtruth CSI.

Figs. 3.16 and 3.17 respectively show the MAE and EE versus user speed comparisons between our proposed method, MLP, RNN and persistence model baselines. Fig. 3.16 indicates that the performance of the persistence model degrades significantly even with a user speed as slow as 10^{-3} m/s. This demonstrates that even a very little movement of the user could result in a significant CSI change due to the high operating frequency of the THz channel. Although the RNN and MLP models can provide much lower MAE, they are still not as robust as our proposed TE-CIE model in terms of a much smaller difference in MAE when increasing user speeds. Fig. 3.17 indicates the success of using TE-CIE together with the proposed EE optimisation algorithm. The comparison suggests that our proposed deep channel prediction-based EE optimisation method is robust for both low and high user speeds.

The optimised EE is closely related to the prediction error whose relationship with the user speed is shown in Fig. 3.16, though there is no closed-form mathematical relationship between the optimised EE and user speed. To elaborate, the procedure for calculating the EE under different user speeds (as depicted in Fig. 3.17), is given as follows:

- We first train different deep learning models based on CSI sequences generated by various user speeds and predict the next CSI matrix $\hat{\mathbf{H}}_r$ of the next slot. This results in different prediction errors by comparing the predicted CSI matrix $\hat{\mathbf{H}}_r$ with the groundtruth $\mathbf{H}_r$.
- We then use the predicted CSI matrix of the next slot, $\hat{\mathbf{H}}_r$, to optimise the EE function and obtain the IRS phase shift matrix $\hat{\blacksquare}$ and the hybrid precoding matrix $\hat{\mathbf{W}}$.
- Finally, we calculate the groundtruth EE using $\hat{\blacksquare}$ and $\hat{\mathbf{W}}$ and the groundtruth CSI of the next slot.

The above procedure shows that the groundtruth EE depends on the accuracy of the optimised IRS phase shift matrix $\hat{\blacksquare}$ and hybrid precoding matrix $\hat{\mathbf{W}}$ which in turn is dependent on the accuracy of the model's prediction, i.e., a higher prediction error results in a worse optimised EE value.

3.5 Conclusion

In this chapter, our exploration delved into a pioneering approach centred around employing deep learning techniques to tackle the intricate challenges of channel prediction and EE optimisation within an IRS-assisted THz communication framework. Our primary focus revolved around devising a deep learning-driven method tailored for forecasting time-varying fading channels. We introduced the TE-CIE model, adeptly constructed to capture the nuanced temporal correlations between past CSI and the subsequent CSI. This model serves as a robust predictor, crucial for anticipating channel behaviour accurately.

Simultaneously, we scrutinised the EE optimisation quandary within an IRS-assisted MU-MISO system operating in the realm of THz communications. Our examination honed in on optimising the precoding matrix alongside the IRS phase shift matrix, strategically aligning them to maximise system EE while adhering to the confines of maximum transmit power. The amalgamation of the TE-CIE channel prediction method with our devised EE optimisation algorithm culminated in a groundbreaking deep learning-infused prediction optimisation framework. This fusion, tailored for EE maximisation in the IRS-assisted THz MU-MISO communication system, represents a notable stride in enhancing performance metrics.

The simulations conducted to validate our proposed scheme unequivocally underscore its prowess. Our framework showcased a substantial enhancement, exhibiting an EE improvement of at least twice the magnitude when juxtaposed against prevailing methodologies documented in the literature. Such tangible results reaffirm the efficacy and potency of integrating deep learning methodologies into the realm of THz communication systems, particularly in optimising EE through accurate channel prediction and strategic resource allocation.

Moreover, the implications extend beyond the scope of THz communication systems. The utilisation of deep learning-based time-series forecasting algorithms harbours the potential not only to fortify IRS-based THz systems but also to fortify the security of UAV swarm-based communication networks. The subsequent chapter is poised to pivot towards the creation of a resilient UAV swarm position optimisation system. This pursuit aims to mitigate security threats stemming from malicious GNSS spoofing attacks, ensuring the robustness and reliability of these communication networks in the face of adversarial interference.

Chapter 4

Deep Learning for Secure UAV Swarm Communication Under Malicious Attacks

4.1 Introduction

UAVs [41, 14, 42], which are commonly known as drones, have attracted great attention in a diversity of applications including aerial photography and videography, disaster zone mapping, product delivery, etc., due to their low acquisition, maintenance costs, ease of deployment, and high-manoeuvrability and ability to hover. UAVs are also widely used in modern wireless communication systems such as playing a role as relays or aerial base stations for public safety communications and network provisioning in emergencies. Due to the high possibilities of line-of-sight (LOS) air-to-ground communication links, UAVs have become a promising solution to enhance conventional wireless networks. To overcome the limited coverage area and capacity problems of a single UAV, UAV swarms [15–17], where multiple UAVs operate cooperatively, are employed to adapt to complicated wireless communication scenarios. While UAV swarms offer tremendous benefits for modern wireless communication, their effectiveness can be compromised by real-world malicious attacks, especially when malicious GNSS spoofing attackers come into play. These attackers can disrupt the precision and accuracy of UAV operations, impacting the performance of prediction and optimisation methods utilised within UAV swarm-enabled wireless communication systems. Therefore, the severe security problem motivates us to develop innovative solutions that can fortify UAV swarm-enabled wireless communication systems against the disruptive presence of malicious GNSS spoofing attackers.

4.1.1 Prior Works

One of the potential applications of UAV swarms is to cooperatively serve multiple ground users [15–17, 83, 84]. Considering the UAV's high and controllable mobility, the communication quality between UAVs and ground users can be significantly improved by proper user assignment and drone trajectory scheduling. The authors in [26] proposed a robust trajectory and communication design for a multi-UAV-enabled wireless communication system in the presence of jammers with imperfect location information. An efficient convex approximation method was proposed to jointly optimise UAV trajectory and transmission power. In [27], a joint trajectory and power control scheme was proposed to minimise the cross-link interference caused by the LOS-dominated propagation conditions between UAVs. Recently, the authors in [28] proposed a novel multi-agent deep reinforcement learning-based UAV swarm trajectory design and resource allocation scheme that can jointly optimise the user association, UAV power allocation, and trajectory design.

Most of the existing works of such systems in the literature (i.e., in [26–28]) assume static user locations while only a few consider user mobility [83–85]. The authors in [83] proposed a joint trajectory and communication design for multi-UAV-enabled wireless networks where multiple moving UAVs serve multiple ground users. Although the user assignment and UAV position are jointly optimised, the users are assumed to have fixed locations. Such design cannot be adapted to mobile networks where users are supposed to occasionally move. Besides, a joint optimisation scheme for multi-UAV scenarios of access and backhaul links was proposed in [84]. Although user mobility pattern was considered, the algorithm proposed in [84] assumed the user locations are known to each drone. Such an assumption is hard to realise in practice since the user location information reported to the drones may be outdated due to the fast movement of users.

In the traditional approaches, the UAV can be assisted with a radar, range measuring devices, or assisted methodology to track the user, e.g., the target tracking frameworks proposed in [86, 87], to estimate the future locations. However, those traditional approaches have a few drawbacks including LOS dependency, limited resolution, and high power consumption. Specifically, traditional radar systems usually require a clear LOS to the target. Obstacles, buildings, or terrain can obstruct radar signals, limiting their effectiveness in urban environments or areas with significant obstructions. Radar systems typically provide limited resolution, making it challenging to distinguish between multiple closely spaced targets or to track targets with fine-grained precision. Moreover, radars often require substantial power to operate, which can limit their operational duration in battery-powered or remote scenarios. All these drawbacks lead to a demand for developing a high-precision and cost-effective method, e.g., a data-driven approach such as a machine-learning-based location forecasting model. A well-trained data-driven forecasting model is

valuable because it considers the user's past movements, habits, and patterns, which can provide more accurate predictions. Moreover, equipping a forecasting model can be more energy-efficient than using a high-power radar system. Recently, a deep learning-based joint resource allocation and trajectory design was proposed in [85]. The authors used joint deep reinforcement learning and deep unfolding networks to solve the non-convex optimisation problems. A Gauss Markov (GM) model-based user mobility pattern was applied to describe the random user movement patterns.

The ground user mobility model that is widely considered in the literature (i.e. in [84, 85]) is either a simple RWP model in which each user randomly chooses a point as its destination for each move or a classic GM model where the speed and direction of a mobile terminal are updated according to their past values at earlier time intervals [88]. However, according to the latest study on human mobility [89], real-world mobile users follow much more complicated physical laws than these two classic models. To accurately forecast intricate mobility patterns, robust and sophisticated forecasting algorithms, such as deep learning methods, are required.

In recent years, the availability of massive data and rapid computation power has led to deep learning algorithms becoming a crucial aspect of next-generation time-series forecasting techniques. Unlike traditional parametric models that require domain expertise, deep learning-based forecasting methods solely rely on data-driven approaches and do not require any prior knowledge. With their ability to perform non-linear transformations and universal approximation, deep learning algorithms can capture complicated implicit patterns that traditional parametric models cannot. In instances where the underlying data pattern is intricate, using a data-driven deep learning algorithm can significantly decrease the amount of human supervision required for designing a dependable expertise-informed parametric model. Some recent works [90, 91] focus on investigating deep learning-based user mobility prediction methods for UAVs to enhance the flying trajectory design by recognising the complicated underlying user movement patterns.

However, the performance of prediction-based methods is highly dependent on the 'actual' location of the user and can be easily degraded due to malicious attacks. The existence of malicious attackers has become a severe threat to wireless networks due to the vulnerable nature of wireless propagation [92]. One common location-targeted malicious attack type is called a GNSS spoofing attack where the attacker opportunistically spoofs the victims by modifying or sending incorrect and fake location information [93, 44]. Such attacks can significantly degrade the prediction performance of deep learning-based user mobility forecasting which urges the use of robust training methods that are not heavily influenced by malicious attacks. The formulation of multiple spoofers in the scene to make a stealthy GNSS spoofing for generalised cases (including UAV-based systems) has been studied in [94, 95]. Specifically, the authors in [94] analysed the requirements for

successful GNSS spoofing attacks on individuals and groups of victims with civilian or military GNSS receivers. It is also proven in [95] that the constrained power is so powerful, and the target is unaware of the spoofing activity. The above works have not studied the specific techniques and methods for achieving the optimal sum rate in UAV systems under GNSS spoofing attacks. In this chapter, we mainly focus on the robust optimisation of the overall sum rate of the UAV system. Therefore, we simplify the success hit ratio of GNSS spoofing to a spoofing probability in general and do not consider a specific spoofing system design. Recent works proposed in [18–20] only focused on detecting and classifying the GNSS spoofing attacks. However, robust methods of reconstructing the information from corrupted data and combining it with user mobility prediction have not been studied in the literature yet. More specifically, the primary challenge of the combination problem lies in ensuring the integrity and reliability of data that has been corrupted by GNSS spoofing attacks. For example, when attackers manipulate GPS signals to mislead UAVs, the data collected from these compromised sources becomes untrustworthy. Combining such data with user mobility predictions can introduce errors and inaccuracies into the system. In this case, using this corrupted data for solving underlying optimisation problems, e.g., the methods proposed in [83–85], can significantly compromise the accuracy of the solution, potentially leading to suboptimal or incorrect decisions in UAV swarm operations. On the other hand, traditional deep learning-based user mobility prediction models or training methods, e.g., those proposed in [90, 91], cannot ensure the robustness of the prediction precision when the input user location information is corrupted by malicious attacks. Therefore, it becomes critical to design a reliable deep learning architecture suitable for user mobility prediction and to develop a dedicated training method that can yield a robust model. In the following section, we are going to explain our novel approach to solving the problem, i.e., we propose a novel deep learning model architecture and training pipeline that can learn to retrieve clean data from the corrupted data and accordingly make an accurate prediction that can be used by the underlying optimisation algorithm.

Although modern deep learning models can offer tremendous benefits for wireless communication, they usually suffer from high computational complexity and energy consumption, making it challenging to deploy them on energy-limited devices such as UAVs. Specifically, the state-of-the-art (SOTA) time-series model, i.e., Transformer [7], has a quadratic time complexity for the sequence length. Therefore, it is critical to propose an effective method to reduce the complexity while maintaining the prediction power. Recent advancements in knowledge distillation [21] showcase its ability to reduce a model's time complexity while maintaining prediction power by transferring learned knowledge from a complex model to a smaller, more efficient one. This technique enables smaller models to emulate the performance of larger counterparts, achieving decreased computational demands without compromising predictive accuracy.

4.1.2 Contributions

In this chapter, we propose a deep learning-based user mobility prediction, user assignment, and drone position optimisation scheme for UAV swarm-enabled wireless communication systems in the presence of malicious GNSS spoofing attackers. Specifically, multiple drones are employed to cooperatively serve a group of moving users on the ground in a given 2D area whose most recent location data are periodically reported to the UAV swarm. A malicious attacker continuously tries attacking and spoofing the location data to degrade the performance of UAVs' user mobility predictor. It is worth noting that we do not consider a specific spoofing activity such as jamming, meaconing, repeater-based spoofing, etc. [96, 97]. Instead, we measure the spoofing success hit ratio as a spoofing probability. In general, our proposed algorithm can work in all cases of spoofing activities. Based on these assumptions, we first propose a DART model for the user mobility prediction model which is robust to such GNSS spoofing attacks. With the predicted user location information, we then propose two efficient user assignment and drone position optimisation algorithms, namely SCA and SDP, respectively. Simulation results show that our proposed DART model outperforms LSTM baselines in terms of much lower mean squared error (MSE) loss on the test set. Meanwhile, the proposed SDP method is demonstrated to be superior to the conventional SCA method by offering a much higher overall sum rate. Finally, the deep learning-based prediction-optimisation scheme is proven to achieve a near-optimal overall sum rate compared with using the ground-truth user location information for optimisation. The pre-trained and fine-tuned DART model with the SDP method can provide up to 30% higher overall sum rate compared with the adversarial trained LSTM baseline and almost double the overall sum rate compared with the vanilla LSTM baseline. It is worth clarifying that our proposed specific user location forecasting model works on an arbitrary number of users because each sequence it processes is independent of others. Since the model does not require user identification information and is fitted to general user movement patterns (i.e., the EPR user mobility model), it can also adapt to any new-coming user that follows the movement pattern. The test set used for our simulation results is assumed to be fully out-of-sample, which means the users are not seen by the model in the training set. Moreover, we further distil the Transformer model into a smaller GRU model based on the knowledge distillation method to reduce the time complexity of the model while maintaining its prediction power. Simulation results demonstrate that the optimised sum rate using the distilled GRU student model's predicted user locations can achieve almost 99% compared to the Transformer teacher model. Meanwhile, the inference time of the student model is only 4% compared to the teacher model.

The remainder of this chapter is organised as follows. Section 4.2 introduces the UAV swarm-enabled wireless communication system model. In Section 4.3, we formulate

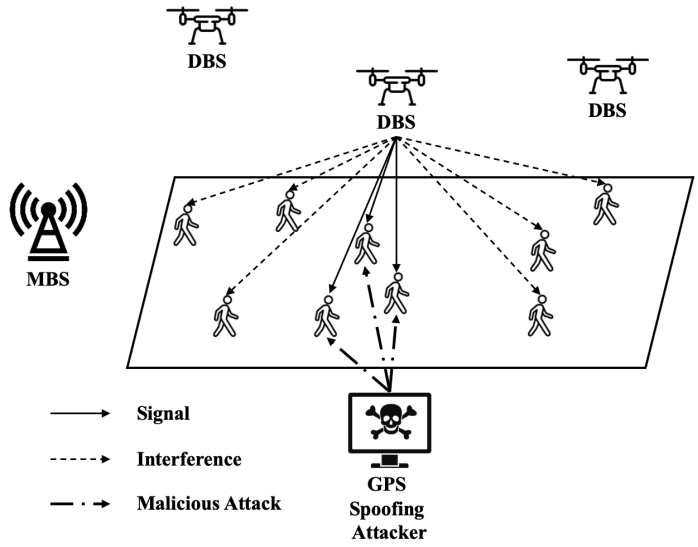

Fig. 4.1 System model of the communication links between the UAV swarm and users.

the overall optimisation problem. Then in Section 4.4, we explain our proposed deep learning-based prediction-optimisation scheme in detail and the SCA and SDP-based user assignment and drone position optimisation methods are studied. Finally, the simulation results are illustrated in Section 4.5, and the conclusions are drawn in Section 4.6.

4.2 System Model

Consider a UAV swarm-assisted wireless communication system, as shown in Fig. 4.1, where a swarm of N single antenna UAVs acting as drone base stations (DBSs) is serving M moving users in the downlink. The access link established between UAVs and users is connected by a backhaul link to a terrestrial macro base station (MBS). Due to the high transmission power of MBS, the backhaul link usually has a high capacity. Thus, the bottleneck of the achievable rate is usually at the access link. In this chapter, we mainly focus on optimising the achievable rate for the access link.

The UAV and user sets are denoted by $\mathcal{N} = \{1,2,...,N\}$ and $\mathcal{M} = \{1,2,...,M\}$, respectively. All UAVs are assumed to fly at a fixed height H, which is the minimum height to avoid obstacles. The UAVs and users are all initially uniformly distributed over the square service area with a width of r. The three-dimensional (3D) Cartesian coordinate system is considered for the locations of UAVs and users. Therefore, the locations of UAVs $n \in \mathcal{N}$ and users $m \in \mathcal{M}$ are denoted by $\mathbf{q}_n = [x_n, y_n, H]^T \in \mathbb{R}^{3\times 1}$ and $\mathbf{p}_m = [x_m, y_m, 0]^T \in \mathbb{R}^{3\times 1}$, respectively, where x_n, y_n, x_m, y_m all follow a uniform distribution $\mathcal{U}(0,r)$. To facilitate UAV position optimisation, we consider the total service time T which is divided into K

equal time slots d_t so that $T = Kd_t$. Thus, the locations of UAV n and user m at time slot k are denoted by $\mathbf{q}_n[k] = [x_n[k], y_n[k], H]^T \in \mathbb{R}^{3\times 1}$ and $\mathbf{p}_m[k] = [x_m[k], y_m[k], 0]^T \in \mathbb{R}^{3\times 1}$ where $k = 1, ..., K$, respectively.

Without loss of generality, we consider dynamic user assignment between UAVs and users in each time slot by introducing a binary variable $S_{n,m}[k] \in \{0,1\}, \forall n, m, k$ where $S_{n,m}[k] = 0$ indicates that there is no communication between UAV n and user m at time slot k, whilst $S_{n,m}[k] = 1$ represents that UAV n transmits data to user m at time slot k.

According to the field trials in [98, 99], the channels between UAVs and users are mainly dominated by LOS transmission. Thus, the channel power gain between UAV n and user m considering the free space propagation path loss can be formulated as $g_{n,m}[k] = \beta_0 ||\mathbf{q}_n[k] - \mathbf{p}_m[k]||^{-\alpha}$ where α is the path loss exponent and β_0 is the reference channel power gain. Then the achievable rate from UAV n to user m at time slot k is given by:

$$R_{n,m}[k] = \log_2 \left(1 + \frac{P_n g_{n,m}[k]}{\sum_{i\in\mathcal{N}, i\neq n} P_i g_{i,m}[k] + \sigma^2} \right), \forall n, m, \tag{4.1}$$

where $P_i, i \in \mathcal{N}$ is the transmission power of UAV i and σ^2 denotes the power of AWGN. We assume that the UAVs are transmitting with a relatively conservative power which is sufficient to support long-term flight and communication missions.

Recent research works [89] have indicated that the RWP model, which is widely considered in many works in the literature, is not realistic and cannot well reflect the real-world human mobility pattern. Therefore, we consider a family of more realistic user mobility models, namely the EPR model [89], which can better represent the real-world human moving trajectory pattern. The specific EPR model of individual human mobility considered In this chapter consists of the following mechanisms:

- **Waiting Time Choice.** The waiting time δ_t between two movements of the user is chosen randomly from the power-law probability distribution $P(\delta_t) = \delta_t^{-1-\beta} e^{-\frac{\delta_t}{\tau}}$, where parameters β and τ are two arguments of the user movement constructor.

- **Action Selection.** With probability $P_{\text{new}} = \rho S^{-\gamma}$, the user visits a new location (i.e., Exploration phase); Otherwise, it returns to a previously visited location (i.e., Return phase). S is the number of distinct locations previously visited by the user, and parameters ρ and γ are two arguments of the constructor.

- **Exploration Phase.** The user randomly selects a different location within the serving area in which case the number of distinct locations visited, S, is increased by 1.

- **Return Phase.** The user returns to a randomly selected visited location with equal probability.

The moving speed of each user is randomly selected within a specific range for each travel in the exploration and return phases.

The users report their locations to the UAVs in each time slot. Due to the vulnerable nature of wireless communication systems, some real-world malicious attackers (i.e., GNSS spoofing attackers) can modify the location information transmitted by the users [43]. We assume there is a GNSS spoofing attacker that opportunistically modifies the user location information with probability p_s by a random location within the service area. The corrupted user location considering the GNSS spoofing attack is denoted by $\overline{\mathbf{p}}_m, \forall m \in \mathcal{M}$. We assume that the MBS is responsible for offline model training, online user mobility forecasting and online optimisation. Once the optimisation results are obtained, the decisions for updating the UAVs' positions are broadcast to each DBS through a control channel in each time slot.

4.3 Problem Formulation

Considering the mentioned UAV swarm system with the EPR user mobility model, we aim to maximise the overall sum rate of all UAVs in each time slot. It is worth noting that the security threat in this work is based on user location modification rather than communication disruption, which is addressed by our proposed robust deep learning model. Therefore, we choose the general sum rate as the key performance indicator for optimisation. Mathematically, the optimisation problem (P1) is formulated as:

$$
\begin{aligned}
(\mathrm{P1}): \max_{\mathbf{q}_n[k], S_{n,m}[k]} \quad & \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}} S_{n,m}[k] R_{n,m}[k], \\
\text{s.t.} \quad & \sum_{n\in\mathcal{N}} S_{n,m}[k] = 1, \forall m, && \text{(4.2a)} \\
& ||\mathbf{q}_n[k] - \mathbf{q}_n[k-1]|| \leqslant V_{\max} d_t, \forall n, && \text{(4.2b)} \\
& ||\mathbf{q}_n[k] - \mathbf{q}_n[k-1]|| \geqslant V_{\min} d_t, \forall n, && \text{(4.2c)} \\
& ||\mathbf{q}_n[k] - \mathbf{q}_i[k]|| \geqslant d_{\min}, \forall i, n, i \neq n, && \text{(4.2d)} \\
& \cos\phi_n[k] \geqslant \cos\phi_{\max}, \forall n, && \text{(4.2e)} \\
& 0 \leqslant x_n \leqslant r, \forall n, && \text{(4.2f)} \\
& 0 \leqslant y_n \leqslant r, \forall n, && \text{(4.2g)}
\end{aligned}
$$

where $V_{\min}$, $V_{\max}$ and $d_{\min}$ are the UAV's minimum flying speed, the UAV's maximum flying speed and the minimum distance required between any two UAVs, respectively; ϕ_n and $\phi_{\max}$ are the turning angle (i.e., the yaw angle) of UAV n and the UAV's maximum turning angle, respectively. In general, the number of UAVs N should be independent of

the number of users M. In this case, a UAV in a swarm can choose to serve one or more users if they are within its service area, or it can also choose to switch off if there is no user nearby.

The first constraint (4.2a) guarantees that each user is guaranteed to be assigned to only one UAV. The next two constraints (4.2b) and (4.2c) guarantee that UAVs are restricted to flying within a certain range of speed. The fifth constraint (4.2d) ensures there is no collision between UAVs whilst the sixth constraint (4.2e) ensures that UAVs should turn their angles within a restricted range. The last two constraints (4.2f) and (4.2g) ensure the drones are always flying within the service area.

The optimisation problem (P1) is a dynamic control problem since the location of each user changes over time and the UAVs should adjust their positions accordingly. Since the users are continuously moving, their latest reported location information can be outdated. Moreover, the user's reported location may be corrupted by the GNSS spoofing attack, which makes the optimisation problem even more challenging to solve. Therefore, the UAVs should be equipped with a forecasting model that can predict users' future locations based on the previously reported locations considering the GNSS spoofing attack to optimise their positions in advance. Mathematically, we are aiming to find a mapping function which inputs the previous W corrupted locations of a user and outputs the estimated next location as follows:

$$(\text{P2}) : \widetilde{\mathbf{p}}_m[k] = f\left(\overline{\mathbf{p}}_m[k-W], ..., \overline{\mathbf{p}}_m[k-1]\right), \forall m. \tag{4.3}$$

In this case, the optimisation problem (P1) can be decoupled and re-formulated as two sub-problems (P2) and (P3) in each time slot considering the estimated user locations where the problem (P3) can be expressed as:

$$\begin{aligned}(\text{P3}) : \max_{\mathbf{q}_n[k], S_{n,m}[k]} \quad & \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}} S_{n,m}[k]\widetilde{R}_{n,m}[k] \\ \text{s.t.} \quad & (4.2a), (4.2b), (4.2c), (4.2d), (4.2e), (4.2f), (4.2g),\end{aligned}$$

where $\widetilde{R}_{n,m}[k]$ is the estimated achievable rate from UAV n to user m at time slot k considering the estimated user location $\widetilde{\mathbf{p}}_m[k]$ obtained by solving the problem (P2). Specifically, the optimisation problem (P3) is constructed on the estimated user location $\widetilde{R}_{n,m}[k]$ predicted by the deep learning method in (P2).

Since the solution of problem (P3) is obtained from the estimated user locations $\widetilde{\mathbf{p}}_m[k]$, the overall sum rate is optimised if the location estimation error is guaranteed under a robust design scenario, i.e., the Euclidean distances between the estimated user locations and the groundtruth user locations should remain below a given threshold for all $\widetilde{\mathbf{p}}_m[k]$ in

an uncertain region. Mathematically, the robust problem (P3) is formulated as:

$$\begin{aligned}\text{Robust Problem (P3)}: \max_{\mathbf{q}_n[k],S_{n,m}[k]} \min_{||\widetilde{\mathbf{p}}_m[k]-\mathbf{p}_m[k]||\leqslant\varepsilon_{\text{error}}} & \sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}} S_{n,m}[k]\widetilde{R}_{n,m}[k] \\ \text{s.t.} \quad & (4.2a),(4.2b),(4.2c),(4.2d), \\ & (4.2e),(4.2f),(4.2g),\end{aligned} \tag{4.4}$$

where $\varepsilon_{\text{error}}$ characterises the uncertainty threshold of the estimated user locations. Thus, the robust version of the problem (P3) aims at maximising the overall sum rate while considering the worst case of the user location estimation error. According to the definition of the uncertainty in [100], the robust problem (P3) belongs to a type of ellipsoid uncertainty problem, i.e., the uncertain parameter $\widetilde{\mathbf{p}}_m[k]$ is confined in a range of an ellipsoid $\mathcal{H}(\varepsilon_{\text{error}})$, where $\mathcal{H}(\varepsilon_{\text{error}}) := \{\widetilde{\mathbf{p}}_m[k] \,|\, ||\widetilde{\mathbf{p}}_m[k]-\mathbf{p}_m[k]|| \leqslant \varepsilon_{\text{error}}\}$. Therefore, the optimal solution of the robust problem (P3) can guarantee the user location estimation error constraint for all $\widetilde{\mathbf{p}}_m[k] \in \mathcal{H}(\varepsilon_{\text{error}})$, and so the robustness of problem (P3) is in the *worst case* sense [101], i.e., in the case of incurring largest error in location estimation, the estimation constraint should also be satisfied. The robust optimisation problem (P3) can be solved using a similar method in [102] and the details are omitted here due to the page limit.

It can be proven that minimising the user location prediction error $\varepsilon_{\text{error}}$ can maximise the objective function of the optimisation problem (P3). Let $\widetilde{R}$, $\widetilde{\mathbf{q}}$ and $\widetilde{S}$ denote the overall sum rate, the UAV locations and the user cluster optimised using the predicted user locations, respectively. We can have the overall sum rate measured on the groundtruth user locations $\widetilde{R}_{\text{real}} = f(\widetilde{\mathbf{q}},\widetilde{S},\mathbf{p})$ and the overall sum rate measured on the predicted user locations $\widetilde{R} = f(\widetilde{\mathbf{q}},\widetilde{S},\widetilde{\mathbf{p}})$. The error in the objective function due to the difference between $\widetilde{\mathbf{p}}$ and $\mathbf{p}$ is given by:

$$\varepsilon_{\text{objective}} = |\widetilde{R}_{\text{real}} - \widetilde{R}| = |f(\widetilde{\mathbf{q}},\widetilde{S},\mathbf{p}) - f(\widetilde{\mathbf{q}},\widetilde{S},\widetilde{\mathbf{p}})| = |f(\widetilde{\mathbf{q}},\widetilde{S},\widetilde{\mathbf{p}}+\Delta\mathbf{p}) - f(\widetilde{\mathbf{q}},\widetilde{S},\widetilde{\mathbf{p}})|, \tag{4.5}$$

where $\Delta\mathbf{p} = \varepsilon_{\text{error}}$ refers to the signed difference between $\mathbf{p}$ and $\widetilde{\mathbf{p}}$. We can then use the Taylor series expansion to approximate $\widetilde{R}$ around $\widetilde{\mathbf{p}}$, which is given by:

$$f(\widetilde{\mathbf{q}},\widetilde{S},\widetilde{\mathbf{p}}+\Delta\mathbf{p}) \approx \widetilde{R} + \nabla\widetilde{R}\cdot\Delta\mathbf{p}, \tag{4.6}$$

where $\nabla\widetilde{R}$ is the gradient of $\widetilde{R}$ with respect to $\mathbf{p}$ evaluated at $\widetilde{\mathbf{p}}$. The error in the objective function can then be approximated by:

$$\varepsilon_{\text{objective}} = |\widetilde{R}_{\text{real}} - \widetilde{R}| = |\nabla\widetilde{R}\cdot\Delta\mathbf{p}| = |\nabla\widetilde{R}\cdot(\widetilde{\mathbf{p}}-\mathbf{p})|. \tag{4.7}$$

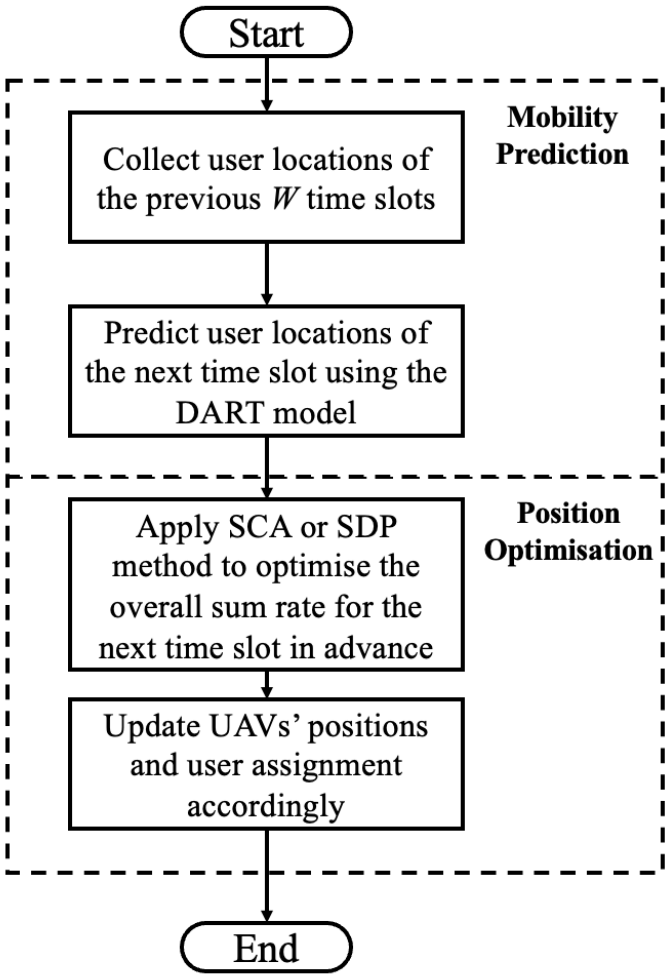

Fig. 4.2 The flow chart of the proposed user mobility prediction and UAV position optimisation scheme.

Therefore, minimising the user location prediction error can maximise the objective function of the optimisation problem.

In the following sections, we introduce the DART model for user mobility prediction in detail and then propose two efficient optimisation methods, namely SCA and SDP, to solve the non-convex optimisation problem (P3).

4.4 Proposed Algorithms

In the following, the DART model is proposed for user mobility prediction whilst SCA and SDP algorithms are introduced to optimise the user assignment and UAV positions, whose flow chart is illustrated in Fig. 4.2. Specifically, we aim to find an effective

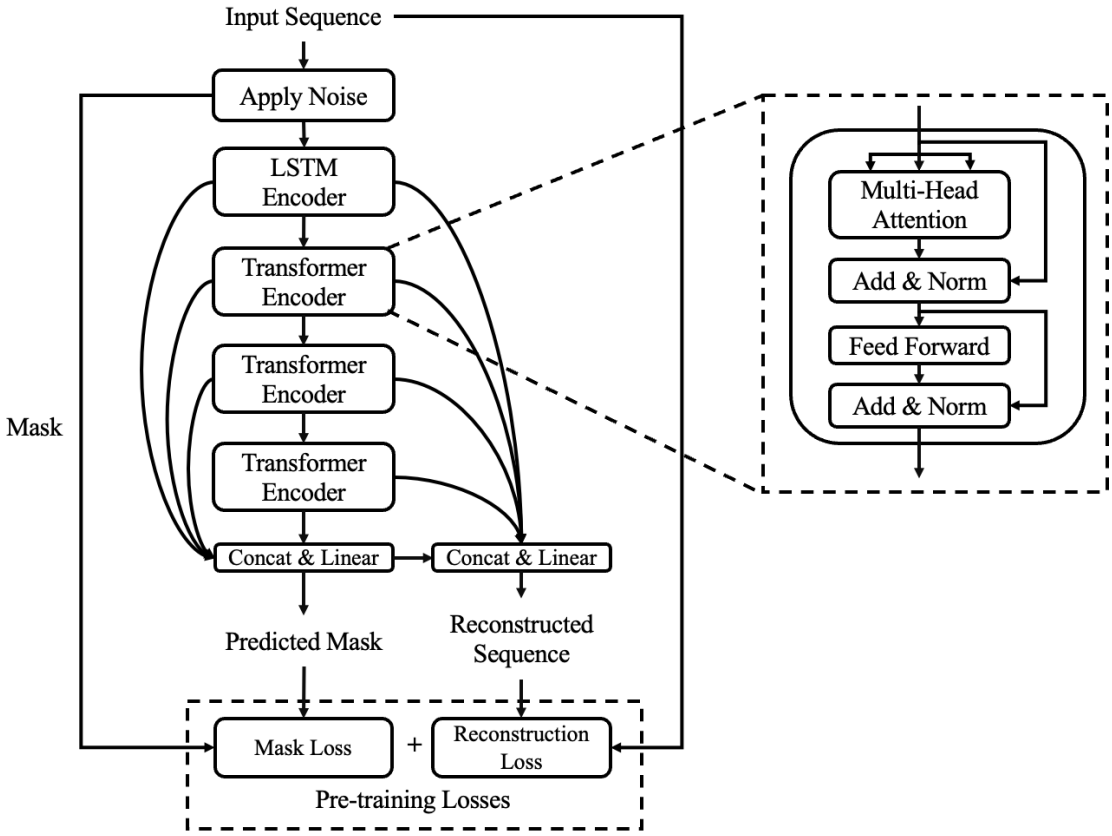

Fig. 4.3 The model architecture of our proposed DART model for adversarial pre-training.

mapping function to solve the problem (P2) and then optimise the user assignment and UAV positions in advance to solve the problem (P3).

4.4.1 Mobility Prediction: Denoising Autoencoder Recurrent Transformer

In this section, a deep learning-based user mobility prediction model, namely DART, is developed to predict the user's location in the next time slot given its previous W location vectors. We assume that the offline training data (i.e., historical user location sequences) is clean and the GNSS spoofing attacker corrupts the location information in the online inference stage. In this case, a model trained on clean data has a high probability of failing in the online inference stage. We also assume the detection probability of the user is unity but the number of users can vary in each time slot.

There are two major reasons why assuming clean offline training data and susceptibility to GNSS spoofing during online inference is realistic in real-world applications. First, during the offline training phase, data is typically collected in controlled environments or

historical datasets where the data can be carefully curated and validated to ensure accuracy. However, acquiring clean and reliable data in dynamic and uncontrolled environments can be challenging in real-world applications. Factors such as signal obstructions, interference, and the existence of spoofing attackers can introduce inaccuracies. Second, acquiring high-fidelity real-world location data for training can be expensive and resource-intensive. Therefore, in practice, organisations might rely on historical datasets, publicly available data, or simulated data, which may not fully reflect the complexities and nuances of real-world scenarios where spoofing attacks should be taken into account.

One common solution to this kind of problem is to apply 'adversarial training' [50–52], i.e., introducing a similar type of noise to the model during the offline training to prevent failure during the online inference. With such kind of training method, the model can learn the spoofing pattern in advance. However, only applying adversarial training is not enough. To further improve the prediction performance, we propose an adversarial pre-training and fine-tuning scheme. In the adversarial pre-training stage, the model learns to predict which location vector is corrupted in the sequence and tries to reconstruct or denoise it. Then in the adversarial fine-tuning stage, the model learns to predict the future user location.

Adversarial Pre-training

The architecture of our proposed DART model for adversarial pre-training is shown in Fig. 4.3. The general model architecture follows four layers, namely the noise layer, the recurrent layer (i.e., the LSTM encoder layer), the stacked transformer layers, and the concatenation and linear layers. The purpose of adversarial pre-training is to exploit the model's capability of distinguishing clean and corrupted location information and then trying to denoise the sequence. The input of the model is the historical user mobility sequence $\mathbf{X} \in \mathbb{R}^{W\times 2}$ containing the last W 2D locations of the user. The output of the model consists of two parts, namely probability of predicted mask $\widetilde{\boldsymbol{\psi}} \in \mathbb{R}^{W}$ and reconstructed location sequence $\widetilde{\mathbf{X}} \in \mathbb{R}^{W\times 2}$, respectively.

The noise layer applies synthetic spoofing noise to the input to imitate the GNSS spoofing attack. First, a binary mask $\boldsymbol{\psi} \in \mathbb{R}^{W}$ which follows the Bernoulli distribution of probability p_{mask} is created. Then a random sequence $\mathbf{U} \in \mathbb{R}^{W\times 2}$ is sampled from a 2D uniform distribution which represents random points within the service area. The output of the noise layer, i.e., the corrupted location sequence, $\hat{\mathbf{X}} \in \mathbb{R}^{W\times 2}$ is given by:

$$\hat{\mathbf{X}} = \begin{cases} \mathbf{X}, & \psi_i = 0, \\ \mathbf{U}, & \psi_i = 1, \forall i, \end{cases} \tag{4.8}$$

where ψ_i is the ith element in $\boldsymbol{\psi}$. It is worth noting that Eq. (4.8) refers to the pretraining noise added to the model training, which is used to teach the model how to distinguish

clean user location information from corrupted information in the historical sequences. With such a pertaining task, the model can learn to classify and fix the spoofed location information from the spoofed data. The reason that we use a random noise model here is to address a more general spoofing attack case rather than studying a specific type of spoofing attack. On the other hand, we also aim at making the pretraining task more difficult for the model by aggressively adding random noise instead of directional patterned noise such as persistent false target, persistent walking target, persistent pull-off target, and persistent walking pull-off target models in [97], though the model can also learn how to distinguish from these types of noise theoretically by additionally incorporating them into the pretraining task.

After the noise layer, the corrupted sequence $\hat{\mathbf{X}}$ is passed through an LSTM encoder layer where the sequential dependence and positional relationship are learned. The LSTM encoder consists of three key components, namely the input gate, forget gate and output gate, respectively. These gates control the ratio of the amount of information to be stored or forgotten. The input gate decides how much new information flows into the memory cell whilst the forget gate determines the amount of information that should be dropped. Meanwhile, the output gate imposes controls on the amount of information passed into the rest of the network. The mapping function of the LSTM layer from the input $\hat{\mathbf{X}}$ to the output $\mathbf{H} \in \mathbb{R}^{W \times d_{\text{model}}}$ is precisely specified by:

$$\mathbf{i}_w = \text{sigmoid}(\mathbf{W}_i^x \cdot \hat{\mathbf{x}}_w + \mathbf{W}_i^h \cdot \mathbf{h}_{w-1} + \mathbf{b}_i), \tag{4.9a}$$

$$\mathbf{f}_w = \text{sigmoid}(\mathbf{W}_f^x \cdot \hat{\mathbf{x}}_w + \mathbf{W}_f^h \cdot \mathbf{h}_{w-1} + \mathbf{b}_f), \tag{4.9b}$$

$$\mathbf{o}_w = \text{sigmoid}(\mathbf{W}_o^x \cdot \hat{\mathbf{x}}_w + \mathbf{W}_o^h \cdot \mathbf{h}_{w-1} + \mathbf{b}_o), \tag{4.9c}$$

$$\tilde{\mathbf{c}}_w = \tanh(\mathbf{W}_c^x \cdot \hat{\mathbf{x}}_w + \mathbf{W}_c^h \cdot \mathbf{h}_{w-1} + \mathbf{b}_c), \tag{4.9d}$$

$$\mathbf{c}_w = \mathbf{i}_w \odot \tilde{\mathbf{c}}_w + \mathbf{f}_w \odot \mathbf{c}_{w-1}, \tag{4.9e}$$

$$\mathbf{h}_w = \mathbf{o}_w \odot \tanh(\mathbf{c}_w), \tag{4.9f}$$

where $\mathbf{h}_w$ is the w-th row vector in $\mathbf{H}$; the operator $\odot$ refers to the Hadamard product; $\mathbf{W}_i^x \in \mathbb{R}^{d_{\text{model}} \times 2}$, $\mathbf{W}_i^h \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, $\mathbf{W}_f^x \in \mathbb{R}^{d_{\text{model}} \times 2}$, $\mathbf{W}_f^h \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, $\mathbf{W}_o^x \in \mathbb{R}^{d_{\text{model}} \times 2}$, $\mathbf{W}_o^h \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, $\mathbf{W}_c^x \in \mathbb{R}^{d_{\text{model}} \times 2}$, $\mathbf{W}_c^h \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are the corresponding weight matrices; $\mathbf{b}_i \in \mathbb{R}^{d_{\text{model}}}$, $\mathbf{b}_f \in \mathbb{R}^{d_{\text{model}}}$, $\mathbf{b}_o \in \mathbb{R}^{d_{\text{model}}}$, $\mathbf{b}_c \in \mathbb{R}^{d_{\text{model}}}$ are bias vectors; d_{model} is the model's hidden dimension.

The output of the LSTM layer is then fed into a stacked transformer encoder network. The Transformer architecture is an effective solution for sequential modelling problems due to its attention mechanism, allowing it to capture long-range dependencies and relationships within the input sequence. The stacked transformer encoder network contains three transformer encoder layers, whose structures are originally proposed in [7]. As shown

in Fig. 4.3, each layer consists of two sub-layers, namely the multi-head self-attention mechanism layer and the position-wise fully connected feed-forward network, respectively. Moreover, a residual connection [74] with layer normalisation [103] is applied after each of the two sub-layers. It is worth noting that layer normalisation [75] is used in the original transformer architecture as a compromise on the dynamic sequence length issue. However, since the input sequence length is fixed as W, the batch normalisation technique is a better choice.

In general, let $\mathbf{H}_o \in \mathbb{R}^{W \times d_{\text{model}}}$ denote the output of the last layer. The output of the multi-head attention $\mathbf{Y}_{\text{MHA}} \in \mathbb{R}^{W \times d_{\text{model}}}$ of each transformer encoder layer is given by:

$$\mathbf{Y}_{\text{head},i} = \text{softmax}\left(\frac{\mathbf{H}_o\mathbf{W}_i^Q(\mathbf{H}_o\mathbf{W}_i^K)^T}{\sqrt{d_k}}\right)\mathbf{H}_o\mathbf{W}_i^V, \tag{4.10}$$

$$\mathbf{Y}_{\text{MHA}} = \text{Concat}(\mathbf{Y}_{\text{head},1}, ..., \mathbf{Y}_{\text{head},h})\mathbf{W}^O, \tag{4.11}$$

where $d_k = d_{\text{model}}/h$ is the depth of each head and h is the number of heads; $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $\mathbf{W}^O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are the projection parameter matrices; Concat(.) is the concatenation operation which concatenates the output of each head $\mathbf{Y}_{\text{head},i} \in \mathbb{R}^{W \times d_k}$ into a new matrix with dimension $\mathbb{R}^{W \times d_{\text{model}}}$.

Besides the multi-head attention, each transformer encoder layer also contains a fully connected feed-forward network to each position separately and identically. It applies two linear transformations with a GELU activation function in between. The GELU activation function has been proven to be more empirically effective than the traditional Rectified Linear Unit (ReLU) function with advantages such as smoothness around zero, having a continuous derivative and being robust to variations in input data distribution. However, it also has some drawbacks. For example, it is more computationally intensive than ReLU. Moreover, since GELU is a non-monotonic function, it can have both increasing and decreasing regions. This can make training more difficult, especially when compared to simpler activation functions like ReLU, which are monotonic and generally lead to more stable training.

Let $\mathbf{H}_{\text{FFN}} \in \mathbb{R}^{W \times d_{\text{model}}}$ denote the input of the fully-connected feed-forward network, whose output $\mathbf{Y}_{\text{FFN}} \in \mathbb{R}^{W \times d_{\text{model}}}$ is given by:

$$\mathbf{Y}_{\text{FFN}} = \text{GELU}(\mathbf{H}_{\text{FFN}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{4.12}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{model}} \times d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d_{\text{model}}}$, $\mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$ and $\mathbf{b}_2 \in \mathbb{R}^{d_{\text{model}}}$ are projection parameter matrices and vectors, respectively; d_{ff} is the inner-layer hidden dimension.

After that, the outputs of all previous layers including the LSTM encoder layer are concatenated into a new input matrix $\mathbf{H}_{\text{concat}} \in \mathbb{R}^{W \times 4d_{\text{model}}}$ as the input of the linear trans-

formation layer for the mask prediction task. Such a kind of network connection is inspired by the DenseNet architecture [104]. The output of this layer $\mathbf{y}_{\text{mask}} \in \mathbb{R}^W$ can be expressed as:

$$y_{w,\text{mask}} = \text{Sigmoid}\left(\mathbf{H}_{\text{concat}}\mathbf{w}_{\text{mask}}^T + b_{\text{mask}}\right), \tag{4.13}$$

where $y_{w,\text{mask}}$ is the w-th element of the output, denoting the probability of having mask at step w; $\mathbf{w}_{\text{mask}} \in \mathbb{R}^{4d_{\text{model}}}$ and $b_{\text{mask}} \in \mathbb{R}$ are the projection parameter vector and bias, respectively.

Then the output of the mask prediction layer, along with the previously concatenated outputs, are provided as the input of the final reconstruction layer. Let $\mathbf{H}_{\text{rec}} \in \mathbb{R}^{W\times(4d_{\text{model}}+1)}$ denote the input. The final reconstructed sequence $\mathbf{Y}_{\text{rec}} \in \mathbb{R}^{W\times 2}$ for adversarial pre-training can be calculated as:

$$\mathbf{Y}_{\text{rec}} = \mathbf{H}_{\text{rec}}\mathbf{W}_{\text{rec}} + \mathbf{b}_{\text{rec}}, \tag{4.14}$$

where $\mathbf{W}_{\text{rec}} \in \mathbb{R}^{(4d_{\text{model}}+1)\times 2}$ and $\mathbf{b}_{\text{rec}} \in \mathbb{R}^2$ are the projection parameter matrix and bias vector, respectively.

The binary cross-entropy (BCE) loss is applied to the mask prediction task whilst the weighted MSE loss is applied to the sequence reconstruction task where the corrupted steps are given a higher weight. In comparison, the clean steps are given a lower weight. The overall loss function is also task-wise weighted, which is given by:

$$L_{\text{pre}} = w_{\text{mask}} \cdot L_{\text{mask}} + L_{\text{rec}}, \tag{4.15}$$

where w_{mask} is the loss weight set to the mask prediction task; L_{mask} and L_{rec} are the loss of mask prediction task and sequence reconstruction task, respectively. It is worth noting that pre-training enhances underlying task performance by leveraging large-scale data and learning informative representations, enabling the model to generalise better and capture intricate patterns, thus improving its ability to perform specific tasks effectively. Designing two separate pre-training tasks, i.e., mask prediction and feature reconstruction, is beneficial for better representation learning as it encourages the model to capture both contextual relationships through masked sequence modelling and high-level features through reconstruction, leading to a more robust and versatile pre-trained model.

Adversarial Fine-tuning

After the adversarial pre-training, the pre-trained model weights are re-used for the adversarial fine-tuning of the next-location prediction task. The architecture of our proposed DART model for adversarial fine-tuning is shown in Fig. 4.4. Most of the model parts are the same as in Fig. 4.3 except for the last linear layer.

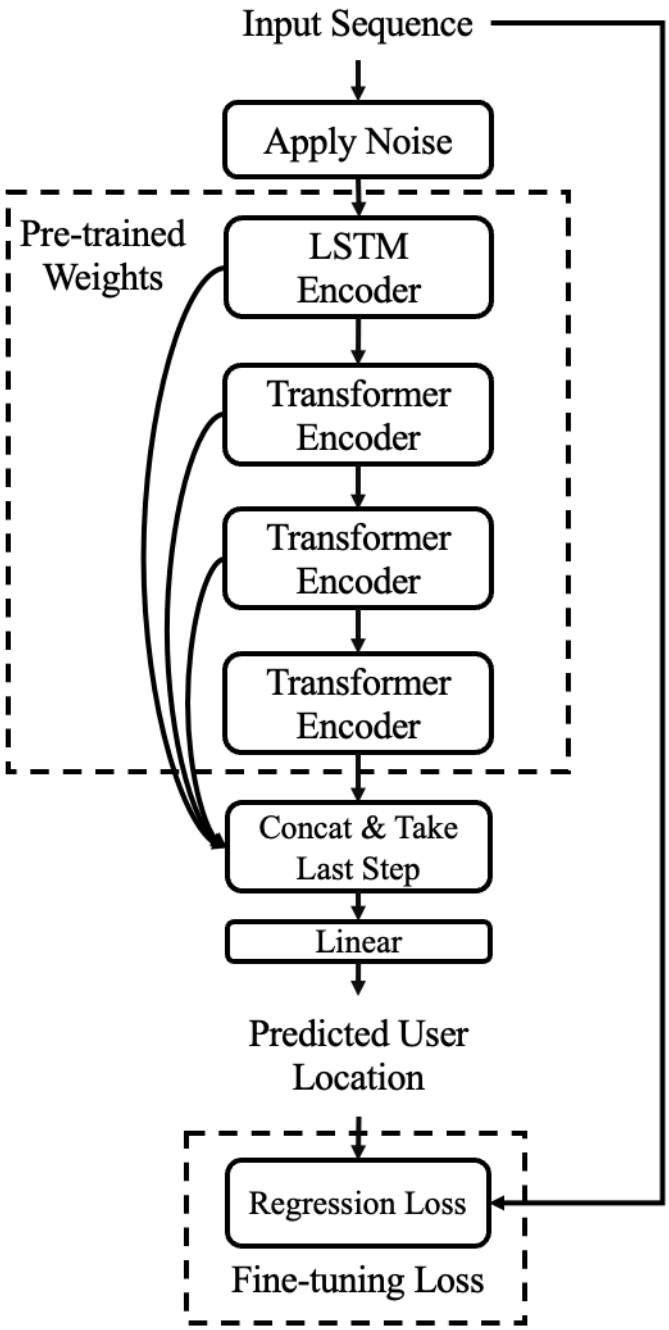

Fig. 4.4 Model architecture of our proposed DART model for adversarial fine-tuning.

The output of all pre-trained encoder layers is first concatenated and only the last step is taken as the input to predict the next user location. These skip connections in the proposed DART model facilitate efficient information flow and gradient propagation across layers, aiding in the effective learning of both local and global features. Let $\mathbf{h}_W \in \mathbb{R}^{1\times 4d_{\text{model}}}$ denote the concatenated hidden vector of the last step. The final predicted user location

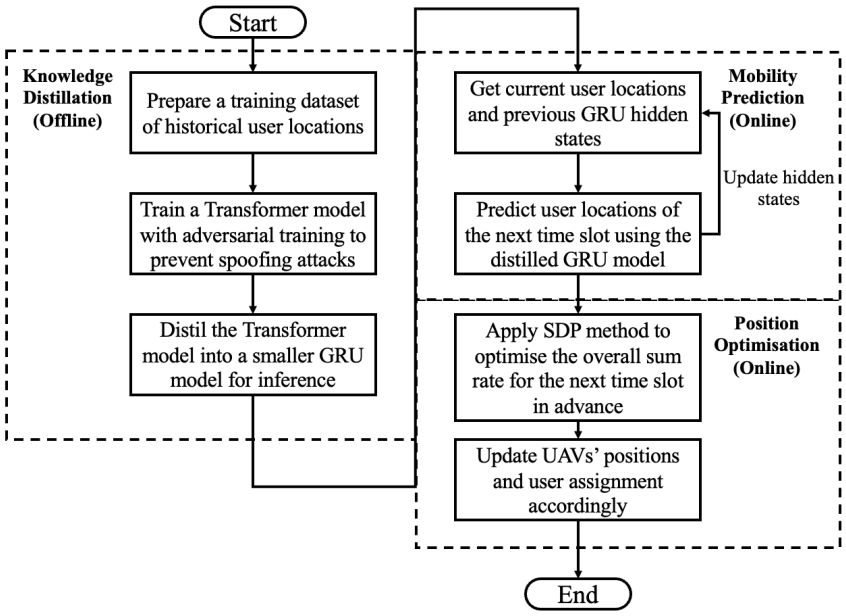

Fig. 4.5 The flow chart of the proposed user mobility prediction and UAV position optimisation scheme with knowledge distillation.

vector $\widetilde{\mathbf{y}} \in \mathbb{R}^{1\times 2}$ is given by:

$$\widetilde{\mathbf{y}} = \mathbf{h}_W \mathbf{W}_{\text{pred}} + \mathbf{b}_{\text{pred}}, \tag{4.16}$$

where $\mathbf{W}_{\text{pred}} \in \mathbb{R}^{4d_{\text{model}}\times 2}$ and $\mathbf{b}_{\text{pred}} \in \mathbb{R}^{1\times 2}$ are the projection parameter matrix and vector, respectively. The MSE loss function is applied since it is a regression task. It is worth noting that the pre-trained network part should apply a much lower learning rate than the last linear layer (or even zero learning rate) to achieve the full benefit from the adversarial pre-training task.

Knowledge Distillation

Although the proposed DART model has emerged as a remarkable innovation with its strong learning capacity, its high inference complexity makes it less feasible for deployment in resource-constrained environments. An ingenious solution is to distil the knowledge learnt by a vast Transformer into a compact low-complexity model that can do fast inference such as GRU. While the GRU model has inherent limitations in learning complicated patterns on its own, its recurrent inference capability is well-suited for certain tasks. The overall flow chart is illustrated in Fig. 4.5 and the knowledge distillation training pipeline is illustrated in Fig. 4.6, where we train the GRU student model using a combination loss L_c, i.e., a

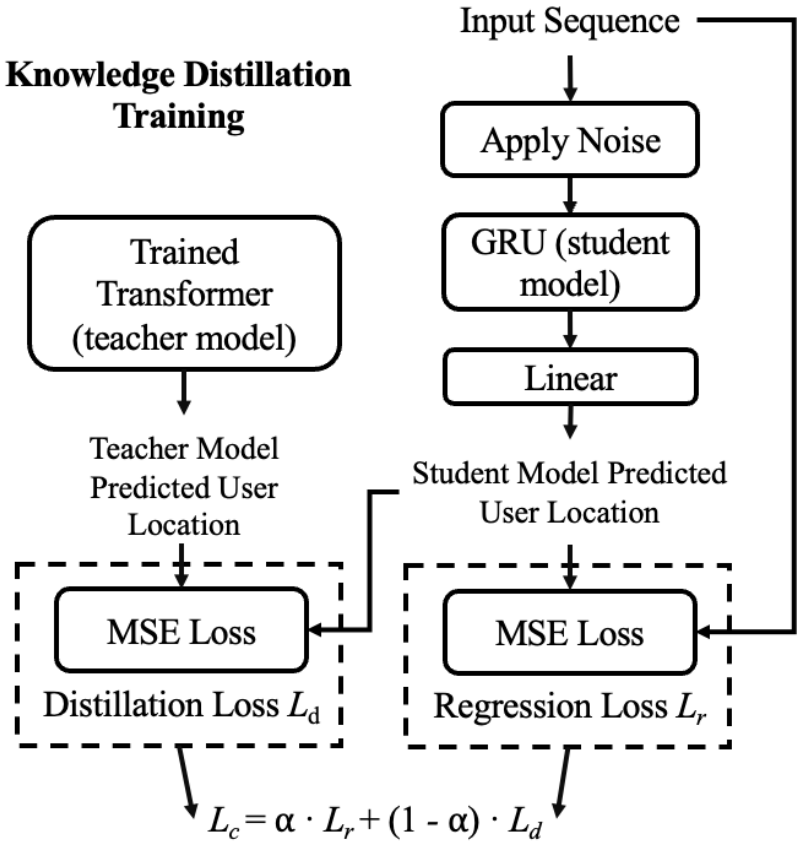

Fig. 4.6 The knowledge distillation and training pipeline for the GRU student model.

weighted average of distillation loss L_d and regression loss L_r, which is expressed as:

$$L_c = \alpha \cdot L_r + (1 - \alpha) \cdot L_d, \tag{4.17}$$

where α is a tunable hyperparameter to control the weight between the model's learning process on knowledge distillation and groundtruth labels.

After training with the proposed knowledge distillation pipeline, the GRU model can predict users' next locations by recurrent inference, i.e., taking the past hidden states and current locations as input and returning the updated hidden states and the predicted locations. It is worth noting that the hidden states for GRU are initialised to all zeros in the first time slot.

4.4.2 Clustering and Position Optimisation: Successive Convex Approximation and Successive Differential Programming

As described in Section 4.3, solving the sub-problem (P3) requires the estimated user location $\widetilde{\mathbf{p}}_m[k]$ predicted by the deep learning model obtained in Eq. (4.16). Since problem (P3) is a non-convex mixed-integer optimisation problem, one commonly used approach is to apply the block coordinate descent (BCD) [105] for successive optimisation. Specifically, the clustering variables and UAV positions are iteratively fixed in turn. In this case, the clustering optimisation and position optimisation problems are solved separately as two sub-problems. In general, the position optimisation problem can be solved by approximating and re-formulating it into a convex optimisation problem using the SCA method. Although solving the re-formulated convex optimisation problem is mathematically efficient, there are still some drawbacks in practice. We then propose an alternative SDP method that outperforms the SCA method in terms of a much higher sum rate.

User Assignment Optimisation

In each iteration of successive optimisation, the UAV positions are first fixed and the clustering variables are optimised by solving an integer linear programming (ILP) problem:

$$\begin{aligned} (\text{P4}): \max_{S_{n,m}[k]} \quad & \sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{M}} S_{n,m}[k]\widetilde{R}_{n,m}[k] \\ \text{s.t.} \quad & (4.2a). \end{aligned} \tag{4.18}$$

The above problem can be efficiently solved by any standard optimisation software library that supports ILP such as CVXPY [106, 107] by using the branch-and-bound method [108].

With the solved clustering variables $S_{n,m}[k]$, the set of users assigned to UAV n can be denoted by $\mathcal{J}_n$. Then problem (P3) can be rewritten as follows:

$$\begin{aligned} (\text{P5}): \max_{\mathbf{q}_n[k]} \quad & \sum_{n\in\mathcal{N}} \widetilde{R}_n[k] \\ \text{s.t.} \quad & (4.2b), (4.2c), (4.2d), (4.2e), (4.2f), (4.2g), \end{aligned} \tag{4.19}$$

where

$$\widetilde{R}_n[k] = \sum_{j\in\mathcal{J}_n} \log_2\left(1+\frac{P_n g_{n,j}[k]}{\sum_{i\in\mathcal{N},i\neq n} P_i g_{i,j}[k]+\sigma^2}\right) \tag{4.20}$$

is the estimated sum rate of UAV n at time slot k considering the estimated user locations. Such a problem can be solved by SCA or our proposed SDP methods which will be explained in detail in the following sections.

$$R_n^{lb}[k] = \log_2\left(1+\frac{1}{A_{n,j}^f[k]B_{n,j}^f[k]}\right) - \log_2 e\cdot\left(\frac{A_{n,j}[k]-A_{n,j}^f[k]}{A_{n,j}^f[k]+\left(A_{n,j}^f[k]\right)^2 B_{n,j}^f[k]}+\frac{B_{n,j}[k]-B_{n,j}^f[k]}{B_{n,j}^f[k]+\left(B_{n,j}^f[k]\right)^2 A_{n,j}^f[k]}\right),\forall n,j, \tag{4.23}$$

Convex Approximation

The optimisation problem (P5) has a non-convex objective function with three convex constraints, i.e., (4.2b), (4.2f) and (4.2g), and three non-convex constraints, i.e., (4.2c), (4.2d) and (4.2e). Thus, the overall problem is a non-convex optimisation problem. In the sequel, we focus on approximating and transforming the non-convex objective function and constraints into convex forms.

By introducing two slack variables $A_{n,j}[k]$ and $B_{n,j}[k]$, problem (P5) can be rewritten as:

$$\begin{aligned}
(\text{P6}): \quad & \max_{\left\{\mathbf{q}_n[k],A_{n,j}[k],B_{n,j}[k]\right\}} \quad \sum_{n\in\mathcal{N}}\hat{R}_n[k] \\
\text{s.t.} \quad & (4.2b),(4.2c),(4.2d),(4.2e),(4.2f),(4.2g), \\
& \frac{1}{A_{n,j}[k]} \leqslant P_n\widetilde{g}_{n,j}[k],\forall n,j, && \text{(4.21a)} \\
& \sum_{i\in\mathcal{N},i\neq n} P_i\widetilde{g}_{i,j}[k]+\sigma^2 \leqslant B_{n,j}[k],\forall n,j, && \text{(4.21b)}
\end{aligned}$$

where

$$\hat{R}_n[k] = \sum_{j\in\mathcal{J}_n}\log_2\left(1+\frac{1}{A_{n,j}[k]B_{n,j}[k]}\right) \tag{4.22}$$

is the transformed sum rate of UAV n at time slot k; $\widetilde{g}_{n,m}[k]=\beta_0||\mathbf{q}_n[k]-\widetilde{\mathbf{p}}_m[k]||^{-\alpha}$ is the estimated channel power gain between UAV n and user m at time slot k considering the estimated user location.

Introducing slack variables $A_{n,j}[k]$ and $B_{n,j}[k]$ potentially enlarges the feasible region of the problem. Specifically, we can always improve the objective value by decreasing the values of $A_{n,j}[k]$ and $B_{n,j}[k]$ if constraint (4.21a) and (4.21b) hold with inequalities in the optimal solution.

After the transformation, the objective function of problem (P6), i.e., Eq. (4.22), becomes convex rather than concave with respect to $A_{n,j}[k]$ and $B_{n,j}[k]$. Therefore, we can apply the first-order Taylor expansion to Eq. (4.22) to obtain a concave global under-

estimation for a given feasible point $\left\{A^f_{n,j}[k], B^f_{n,j}[k]\right\}$, i.e., $R^{lb}_n[k]$, which can be expressed as Eq. (4.23) where $R^{lb}_n[k] \leqslant \hat{R}_n[k]$ provides a tight lower bound. It is worth noting that the feasible point $\left\{A^f_{n,j}[k], B^f_{n,j}[k]\right\}$ is equivalent to the point in the previous time slot $\left\{A_{n,j}[k-1], B_{n,j}[k-1]\right\}$ with respect to UAV's and user's previous locations.

The new constraint (4.21a) is convex whilst constraint (4.21b) is still non-convex. With $\alpha > 1$, constraint (4.21a) can be rewritten as the following convex form:

$$||\mathbf{q}_n[k] - \widetilde{\mathbf{p}}_j[k]|| - \left(P_n \beta_0 A_{n,j}[k]\right)^{\frac{1}{\alpha}} \leqslant 0, \forall n, j. \tag{4.24}$$

Since constraint (4.21b) is non-convex, we can also introduce a new slack variable $C_{n,j}[k]$. In this case, the constraint (4.21b) can be replaced by:

$$\sum_{i \in \mathcal{N}, i \neq n} P_i \beta_0 C_{i,j}[k]^{-\alpha} + \sigma^2 \leqslant B_{n,j}[k], \forall n, j, \tag{4.25a}$$

$$C_{i,j}[k] \geqslant 0, \forall j, \tag{4.25b}$$

$$C_{i,j}[k] \leqslant ||\mathbf{q}_i[k] - \mathbf{p}_j[k]||, \forall i, n, j, i \neq n, \tag{4.25c}$$

where constraints (4.25a) and (4.25b) are convex and (4.25c) is non-convex.

To deal with the non-convexity in constraint (4.25c), the Cauchy-Schwarz inequality $||\mathbf{a}||||\mathbf{b}|| \geqslant |\langle \mathbf{a}, \mathbf{b} \rangle|$ is applied. Then we have:

$$\begin{aligned} ||\mathbf{q}_i[k] - \mathbf{p}_j[k]||\, ||\mathbf{q}^f_i[k] - \mathbf{p}^f_j[k]|| &\geqslant |(\mathbf{q}_i[k] - \mathbf{p}_j[k])^T (\mathbf{q}^f_i[k] - \mathbf{p}^f_j[k])| \\ &\geqslant (\mathbf{q}_i[k] - \mathbf{p}_j[k])^T (\mathbf{q}^f_i[k] - \mathbf{p}^f_j[k]). \end{aligned} \tag{4.26}$$

Hence, we can derive the tight lower bound of $||\mathbf{q}_i[k] - \mathbf{p}_j[k]||$ as:

$$||\mathbf{q}_i[k] - \mathbf{p}_j[k]|| \geqslant \frac{(\mathbf{q}_i[k] - \mathbf{p}_j[k])^T (\mathbf{q}^f_i[k] - \mathbf{p}^f_j[k])}{||\mathbf{q}^f_i[k] - \mathbf{p}^f_j[k]||} \geqslant C_{i,j}[k], \forall i, n, j, i \neq n. \tag{4.27}$$

Similarly, constraints (4.2c) and (4.2d) can also be tightly lower bounded as:

$$||\mathbf{q}_n[k] - \mathbf{q}_n[k-1]|| \geqslant \frac{(\mathbf{q}_n[k] - \mathbf{q}_n[k-1])^T (\mathbf{q}^f_n[k] - \mathbf{q}^f_n[k-1])}{||\mathbf{q}^f_n[k] - \mathbf{q}^f_n[k-1]||} \geqslant V_{\min} d_t, \forall n, \tag{4.28}$$

and

$$||\mathbf{q}_n[k] - \mathbf{q}_i[k]|| \geqslant \frac{(\mathbf{q}_n[k] - \mathbf{q}_i[k])^T (\mathbf{q}^f_n[k] - \mathbf{q}^f_i[k])}{||\mathbf{q}^f_n[k] - \mathbf{q}^f_i[k]||} \geqslant d_{\min}, \forall i, n, i \neq n. \tag{4.29}$$

For the angle constraint (4.2e), the dot product formula of the angle between two vectors $\mathbf{a}$ and $\mathbf{b}$, i.e., $\cos\theta = \frac{\mathbf{a}^T\mathbf{b}}{||\mathbf{a}||||\mathbf{b}||}$, is first applied to transform it into:

$$\cos\phi_n[k] = \frac{\mathbf{x}^T\mathbf{y}}{||\mathbf{x}||\ ||\mathbf{y}||} \geqslant \cos\phi_{\max}, \forall n, \tag{4.30}$$

or into a more tractable form:

$$\mathbf{x}^T\mathbf{y} - ||\mathbf{x}||\ ||\mathbf{y}||\cos\phi_{\max} \geqslant 0, \forall n, \tag{4.31}$$

where $\mathbf{x} = \mathbf{q}_n[k] - \mathbf{q}_n[k-1]$ and $\mathbf{y} = \mathbf{q}_n[k-1] - \mathbf{q}_n[k-2]$, respectively.

Although the new constraint (4.31) is non-convex, Proposition 3 in [109] gives an insightful solution to address the concave lower bound of the first term $\mathbf{x}^T\mathbf{y}$ in (4.31), which is given by:

$$\mathbf{x}^T\mathbf{y} \geqslant (\mathbf{x}^f)^T\mathbf{y} + \mathbf{x}^T\mathbf{y}^f - (\mathbf{x}^f)^T\mathbf{y}^f - \frac{1}{2}||\mathbf{x}-\mathbf{x}^f||^2 - \frac{1}{2}||\mathbf{y}-\mathbf{y}^f||^2, \tag{4.32}$$

where the proof is given in Appendix C in [109].

For the second term $-||\mathbf{x}||\ ||\mathbf{y}||\cos\phi_{\max}$ in constraint (4.31), the Young inequality $ab \geqslant \frac{1}{2}(\varepsilon a^2 + \varepsilon^{-1}b^2)$ can be applied. Then we have:

$$-||\mathbf{x}||\ ||\mathbf{y}||\cos\phi_{\max} \geqslant -\frac{1}{2}\cos\phi_{\max}\left(\varepsilon||\mathbf{x}||^2 + \varepsilon^{-1}||\mathbf{y}||^2\right), \tag{4.33}$$

where $\varepsilon = \frac{||\mathbf{y}^f||}{||\mathbf{x}^f||}$[110]. By combining (4.32) and (4.33), the constraint (4.31) is lower bounded by:

$$\begin{aligned}&(\mathbf{x}^f)^T\mathbf{y} + \mathbf{x}^T\mathbf{y}^f - (\mathbf{x}^f)^T\mathbf{y}^f - \frac{1}{2}||\mathbf{x}-\mathbf{x}^f||^2 - \frac{1}{2}||\mathbf{y}-\mathbf{y}^f||^2 \\ &- \frac{1}{2}\cos\phi_{\max}\left(\varepsilon||\mathbf{x}||^2 + \varepsilon^{-1}||\mathbf{y}||^2\right) \geqslant 0, \forall n,\end{aligned} \tag{4.34}$$

It is worth noting that the convexity in (4.34) is valid if and only if $0 \leqslant \phi_{\max} \leqslant \frac{\pi}{2}$.

Finally, we obtain the convex transformation of the problem (P5) as:

$$\begin{aligned}&(\text{P7}): \max_{\left\{\mathbf{q}_n[k], A_{n,j}[k], B_{n,j}[k], C_{n,j}[k]\right\}} \quad \sum_{n\in\mathcal{N}} R_n^{lb}[k] \\ &\text{s.t.} \quad (4.2b), (4.2f), (4.2g), (4.24), (4.25a), (4.25b), (4.27), (4.28), (4.29), (4.34).\end{aligned} \tag{4.35}$$

The above optimisation problem can be solved using standard convex optimisation methods including existing software libraries such as CVXPY [106, 107]. However, there

$$(\text{P8}): \min_{\mathbf{q}_n[k]} L[k] = H[k] + P_{\text{penalty}}[k], \tag{4.36}$$

$$H[k] = -\sum_{n\in\mathcal{N}} \widetilde{R}_n[k], \tag{4.37}$$

$$\begin{aligned} P_{\text{penalty}}[k] =& D\Bigg[\sum_{n\in\mathcal{N}} \left(||\mathbf{q}_n[k]-\mathbf{q}_n[k-1]|| - V_{\max}d_t\right)^+ + \sum_{n\in\mathcal{N}} \left(V_{\min}d_t - ||\mathbf{q}_n[k]-\mathbf{q}_n[k-1]||\right)^+ \\ &+ \sum_{n\in\mathcal{N},i\in\mathcal{N},i\neq n} \left(d_{\min} - ||\mathbf{q}_n[k]-\mathbf{q}_i[k]||\right)^+ + \sum_{n\in\mathcal{N}} \left(\cos\phi_{\max} - \frac{\mathbf{x}\mathbf{y}^T}{||\mathbf{x}||\,||\mathbf{y}||}\right)^+ \\ &+ \sum_{n\in\mathcal{N}} (-x_n[k])^+ + \sum_{n\in\mathcal{N}} (-y_n[k])^+ + \sum_{n\in\mathcal{N}} (x_n[k]-r)^+ + \sum_{n\in\mathcal{N}} (y_n[k]-r)^+\Bigg], \end{aligned} \tag{4.38}$$

are still some drawbacks in practice. First, the order of magnitudes of $A_{n,j}$ and $B_{n,j}$ is either extremely large or small since the order of magnitudes of the channel power gain is extremely small in practice (e.g., 10^{-13}). This may cause an issue called floating-point overflow (FPO) [111] which prevents the optimisation software from successfully providing an optimal or even valid solution. Secondly, since the constraint (4.34) is non-convex for the maximum angle that is larger than $\frac{\pi}{2}$, the problem cannot be solved by common convex optimisation methods for the cases that the maximum angle is larger than $\frac{\pi}{2}$. Therefore, we also propose an effective alternative optimisation method in the next subsection, namely SDP. It does not require complicated approximation and transformation derivation. Moreover, it is valid in the case that the maximum angle is larger than $\frac{\pi}{2}$.

Differential Programming

To overcome the aforementioned drawbacks in the SCA method, we propose an alternative SDP algorithm. The proposed SDP algorithm relies on automatic differentiation which is widely used in training modern deep learning algorithms such as Pytorch [112] and Tensorflow [113]. Unlike conventional symbolic differentiation methods, automatic differentiation does not require a closed-form mathematical expression of derivatives. Instead, no matter how complicated the function is, automatic differentiation computes the derivatives by applying the chain rules repeatedly to the differentiable operations in the function. Moreover, the widely used SOTA gradient descent solvers such as Adam and root mean square propagation (RMSprop) are proven to have a strong and stable capability of escaping from local optimal or saddle points [114]. Thus, SDP is an effective and efficient method to solve differentiable optimisation problems.

Algorithm 4 Pseudocode for UAV position optimisation using Differential Programming

Define $I_{\max}$ as the maximum number of iterations.
Define $\mathbf{q}_n[k]_{\text{best}}$ as the optimal value of $\mathbf{q}_n[k]$.
Define $H[k]_{\text{best}}$ as the optimal value of $H[k]$.
Initialise $\mathbf{q}_n[k] \leftarrow \mathbf{q}_n[k-1], \forall n$.
Initialise $H[k]_{\text{best}} \leftarrow \infty$.
for $i = 0, ..., I_{\max}$ **do**
    Calculate $L[k]$, $H[k]$ and $P_{\text{penalty}}[k]$ from Eq. (4.36), (4.37) and (4.38).
    Compute the gradient of $\mathbf{q}_n[k]$ on $L[k]$ using automatic differentiation.
    Update $\mathbf{q}_n[k]$ using any popular gradient descent algorithms [114] such as Adam and RMSprop, etc.
    if $H[k] < H[k]_{\text{best}}$ **and** $P_{\text{penalty}}[k] \leqslant 0$ **then**
        $\mathbf{q}_n[k]_{\text{best}} \leftarrow \mathbf{q}_n[k]$.
        $H[k]_{\text{best}} \leftarrow H[k]$.
    end if
end for
Output $\mathbf{q}_n[k]_{\text{best}}$ as the optimal value of $\mathbf{q}_n[k]$.
Output $-H[k]_{\text{best}}$ as the optimal value of estimated sum rate at time slot k.

Although problem (P5) is differentiable with respect to $\mathbf{q}_n[k]$, the four constraints make it a constrained differentiable optimisation problem which cannot be directly solved by applying the SDP algorithm. Therefore, we introduce a penalty term to the objective and transfer the problem (P5) into an unconstrained differentiable optimisation problem (P8) which is given by (4.36), (4.37) and (4.38), where $x^+ = \max(x, 0)$ is the ramp function, factor D is a penalty coefficient with a large positive value which controls the penalty intensity for all the constraints. The ramp function in the penalty terms in problem (P8) ensures the penalty terms only contribute to the objective function if and only if they are positive, i.e., do not satisfy the original constraints.

The general process of the proposed SDP algorithm is presented in Algorithm 4. The proposed algorithm does not have the FPO problem as in the SCA method because there are no introduced slack variables with extreme order of magnitudes. Moreover, since the turning angle constraint is directly addressed as one of the penalty terms, there is no limitation on the maximum turning angle, i.e., it is always valid for any value of $\phi_{\max}$.

Convergence Analysis

The details of the proposed successive optimisation algorithm are provided in Algorithm 5. The overall algorithm splits the original optimisation problem (P3) into two sub-problems (P4) and (P5) and solves them iteratively until the improvement of the objective function is smaller than a small tolerance threshold η.

Algorithm 5 Proposed Successive Optimisation Algorithm

1: Define $\eta > 0$ as the convergence tolerance.
2: Initialise iteration indicator $i \leftarrow 0$.
3: Denote $S_{n,m}[k]$, $\mathbf{q}_n[k]$, $A_{n,j}[k]$, $B_{n,j}[k]$ and $C_{n,j}[k]$ in the ith iteration as $S_{n,m}[k]^{i+1}$, $\mathbf{q}_n[k]^{i+1}$, $A_{n,j}[k]^{i+1}$, $B_{n,j}[k]^{i+1}$ and $C_{n,j}[k]^{i+1}$.
4: Initialise $\mathbf{q}_n[k]^0 \leftarrow \mathbf{q}_n[k-1]$.
5: **repeat**
6: With given $\mathbf{q}_n[k]^i$, obtain $S_{n,m}[k]^{i+1}$ via solving problem (P4).
7: **if** Use SCA method **then**
8: With obtained $S_{n,m}[k]^{i+1}$, compute $\mathbf{q}_n[k]^{i+1}$, $A_{n,j}[k]^{i+1}$, $B_{n,j}[k]^{i+1}$ and $C_{n,j}[k]^{i+1}$ via solving problem (P7).
9: **else if** Use SDP method **then**
10: With obtained $S_{n,m}[k]^{i+1}$, compute $\mathbf{q}_n[k]^{i+1}$ via solving problem (P8).
11: **end if**
12: Update $i \leftarrow i+1$.
13: **until** The convergence condition is satisfied, i.e., the improvement of the objective value is smaller than η.

The proposed successive optimisation algorithm is guaranteed to converge since the objective function in each sub-problem is non-decreasing in each iteration if the SCA method is used to solve the sub-problem (P5) and the maximum objective value of the problem (P3) is finite. If the SDP method is used to solve the sub-problem (P5), the convergence is also guaranteed since the convergence of popular gradient descent algorithms such as the Adam and RMSprop methods on smooth non-convex objective functions has been proven in [115].

Complexity Analysis

The problem (P3) is solved by the BCD method for successive optimisation. The overall complexity depends on the underlying optimisation algorithms for each sub-problem. Therefore, we calculate the complexity of SCA and SDP methods separately.

If the underlying position optimisation method is SCA, then there are $2N+3NM$ variables in each approximated convex subproblem (P7). With convergence tolerance η, the number of iterations required is $\mathcal{O}(\sqrt{2N+3NM}\log_2(1/\eta))$. At each iteration, the complexity of solving the clustering problem (P4) depends on the complexity of the branch-and-bound method, which has a complexity of $\mathcal{O}((NM)^{2.5})$ for the best case and $\mathcal{O}(2^{NM})$ for the worst case [116]. At each iteration, the complexity of solving the position optimisation problem (P7) is $\mathcal{O}(\Phi_1^2\Phi_2)$, where $\Phi_1 = 2N+2NM$ is the total number of variables and $\Phi_2 = 3.5N+3NM+0.5N^2$ is the total number of constraints [117]. As a result, the total complexity of using SCA as the position optimisation method is $\mathcal{O}((NM)^{3.5}\log_2(1/\eta))$ for the best case and $\mathcal{O}(2^{NM}(NM)^{0.5}\log_2(1/\eta))$ for the worst

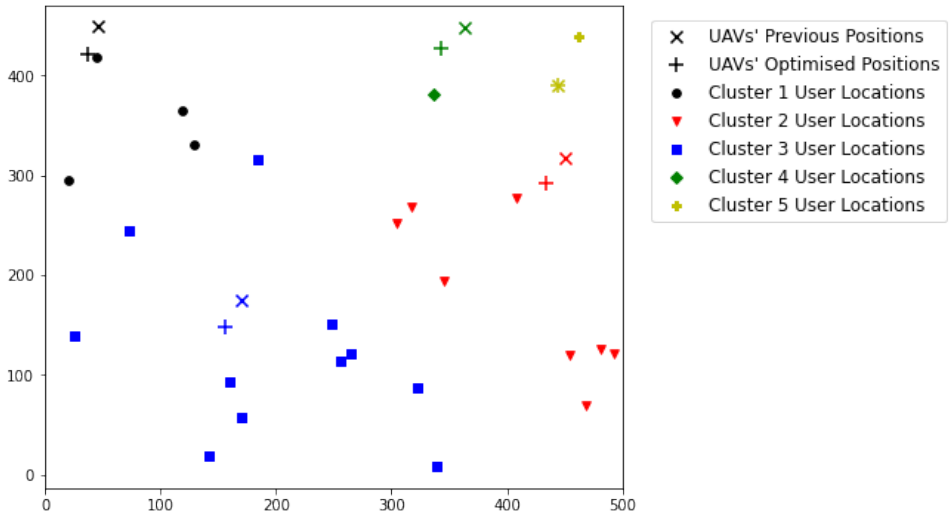

Fig. 4.7 An example scenario generation plot for 25 users' locations and 5 UAVs' previous and optimised positions.

case, given that the number of drones N is usually much smaller than the number of users M they are serving.

If the underlying position optimisation method is SDP, then the main complexity lies in calculating the objective function of the problem (P8). Since there are no extra slack variables for SDP, the complexity of the objective function depends on the number of variables which is $2N$. Then with I denoting the number of iterations, the complexity of the SDP method can be calculated as $\mathcal{O}(IN)$. As a result, the total complexity of using SDP as the position optimisation method is $\mathcal{O}((NM)^3 \log_2(1/\eta))$ for the best case and $\mathcal{O}(2^{NM}(NM)^{0.5} \log_2(1/\eta))$ for the worst case, given that IN is usually an order of magnitude smaller than $(NM)^{2.5}$. It can be seen that using SDP as the underlying position optimisation method has lower overall complexity than using SCA for the best case and the same complexity for the worst case.

4.5 Simulation Results

In this section, we first demonstrate the simulation results for the prediction performance of our proposed DART model and the comparison results between DART and LSTM baseline models. Then we compare the performance of the proposed SCA and SDP algorithms for user clustering and UAV position optimisation. Finally, we illustrate the performance of the deep learning-based prediction-optimisation scheme in terms of the overall sum rate. The main environment parameters for simulation are listed in Table 4.1. An example scenario generation plot for 25 users' locations and 5 UAVs' previous and optimised positions is provided in Fig. 4.7.

Table 4.1 The main system parameters for simulation

Parameter	Value	Description
r	500 m	Service area width
N	10	Number of UAVs
M	50	Number of users
H	20 m	Height
d_t	1 sec	Time slot duration
P_n	20 dBm	Transmit power for each UAV
σ^2	-95 dBm	Noise power
β_0	1e-6	Reference channel power gain
α	2	Path loss exponent
$V_{\min}$	5 m/s	UAV's minimum flying speed
$V_{\max}$	30 m/s	UAV's maximum flying speed
$d_{\min}$	10 m	Minimum distance among UAVs
$\phi_{\max}$	$\pi/3$	Maximum turning angle of UAVs
$V_{\min}^{\text{user}}$	0.2 m/s	User's minimum speed
$V_{\max}^{\text{user}}$	5 m/s	User's maximum speed
ρ	0.6	Rho argument for EPR model
γ	0.21	Gamma argument for EPR model

For our proposed DART model, we use a window size W of 50, hidden dimension d_{model} of 128, number of heads h of 4, inner-layer hidden dimension d_{ff} of 512 and dropout rate of 0.1. For the adversarial pre-training, we use a mask weight w_{mask} of 2, batch size of 512, a learning rate of 10^{-3} and weight decay of 10^{-6} for L2 regularisation. For adversarial fine-tuning, we use a learning rate of 10^{-4} for pre-trained layers and a learning rate of 10^{-3} for the extra linear output layer. The baseline LSTM model uses a

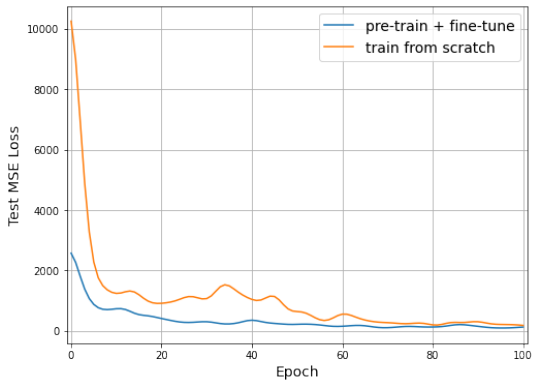

Fig. 4.8 Mean squared error losses smoothed by Gaussian filter with sigma=1.0 on the test set comparison of DART between the adversarial pre-training/fine-tuning scheme and training from scratch (i.e., no adversarial pre-training).

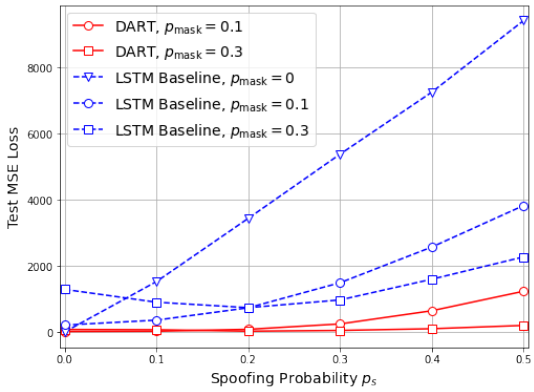

Fig. 4.9 Mean squared error losses on the test set versus spoofing probability comparison between DART and LSTM baseline with different masking probabilities for adversarial pre-training.

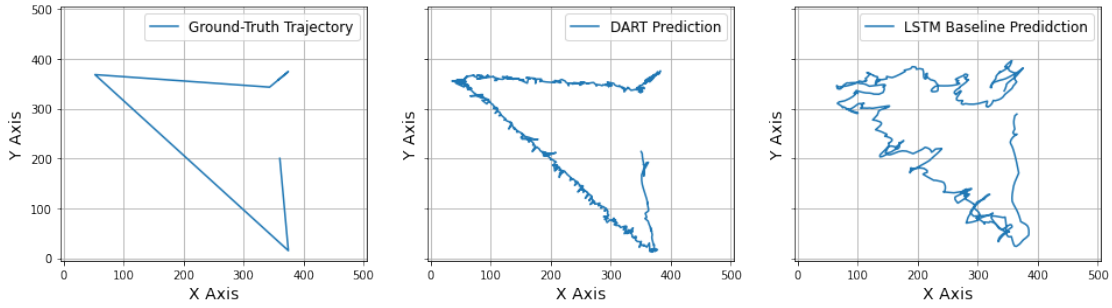

Fig. 4.10 Predicted 2D user trajectories comparison between DART and LSTM baseline.

hidden dimension of 256 and 3 hidden layers. For knowledge distillation, we use a 1-layer GRU, a noise probability of 0.3 for adversarial training and a distillation alpha of 0.5.

For the proposed optimisation algorithms, we consider a convergence tolerance η of 10^{-3}, a scaling factor of 10^{13} for the numerical stability of the SCA method, a penalty coefficient D of 100 and a learning rate of 0.1 for the SDP method.

4.5.1 User Mobility Prediction Performance Analysis

Figure. 4.8 shows MSE losses smoothed by Gaussian filter with sigma=1.0 on the test set comparison of DART between the adversarial pre-training/fine-tuning scheme and training from scratch (i.e., no adversarial pre-training). It can be seen that the pre-training helps stabilise the loss at the fine-tuning stage and makes the model converge much faster and better with fewer epochs than training from scratch, which demonstrates the effectiveness of the adversarial pre-training.

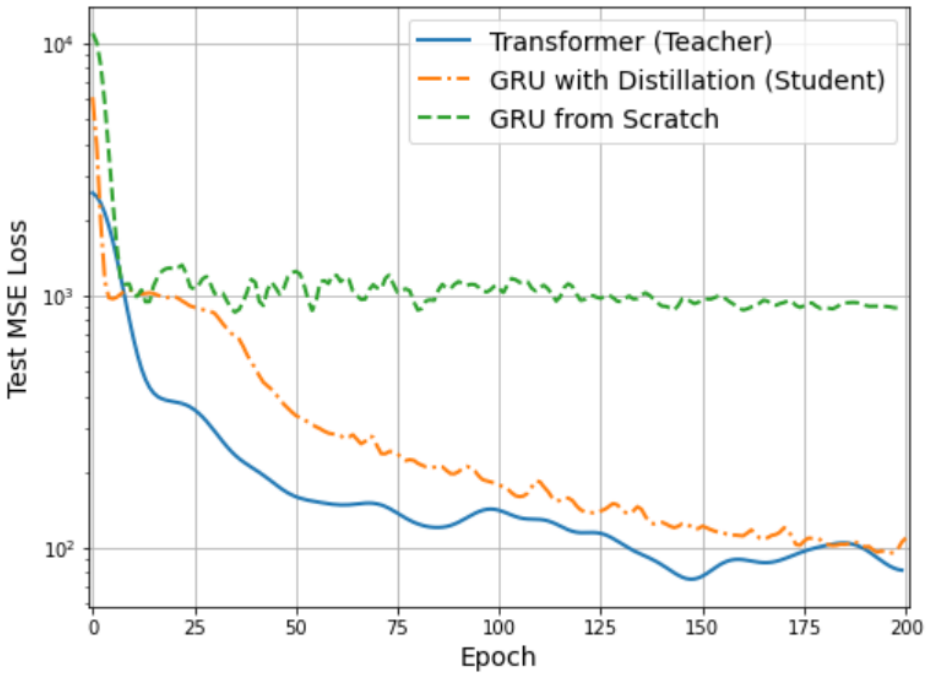

Fig. 4.11 Test loss comparison during training between DART (teacher), GRU with distillation (student) and GRU from scratch.

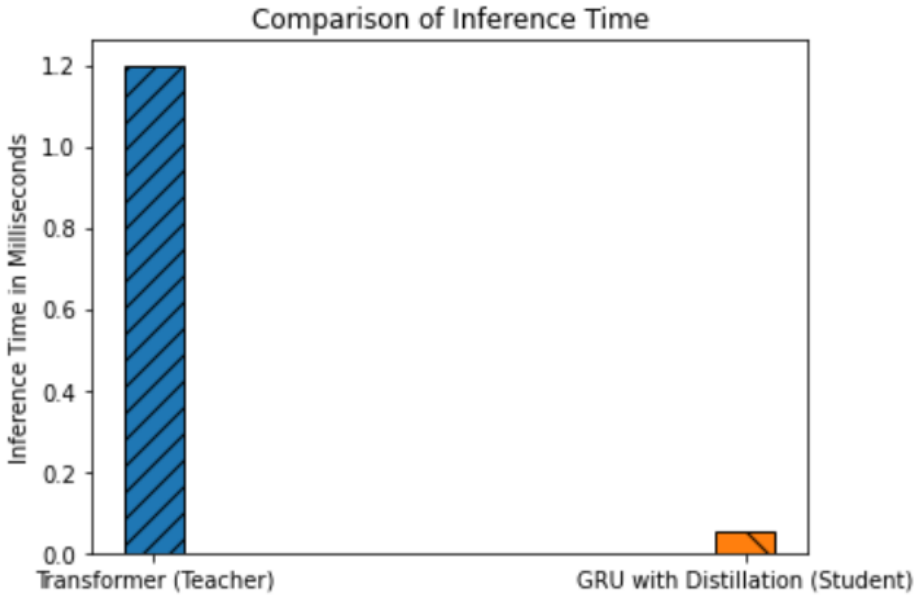

Fig. 4.12 Inference time comparison between DART (teacher) and GRU with distillation (student).

Figure. 4.9 illustrates MSE losses on the test set versus spoofing probability comparison between DART and LSTM baseline with different masking probability for adversarial pre-training. It can be seen that our proposed DART model outperforms the LSTM baseline in

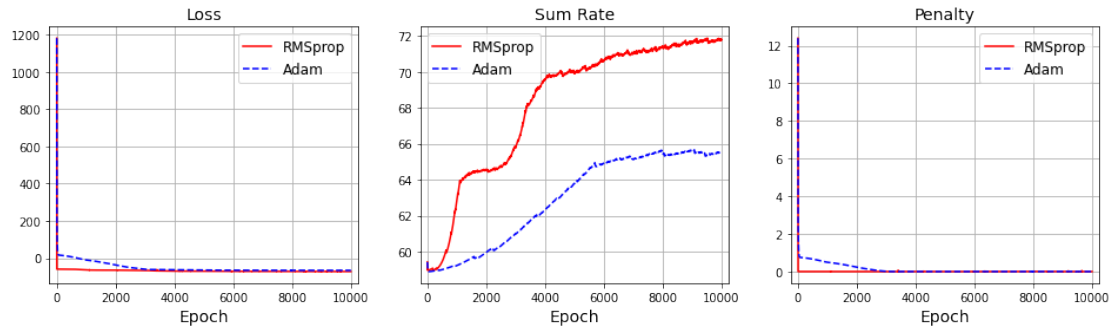

Fig. 4.13 Loss, sum rate and penalty comparisons between different gradient descent algorithms for the proposed SDP method, i.e., Adam and RMSprop algorithms. Loss and penalty terms do not have units, whereas the sum rate is measured in bits per second per Hertz.

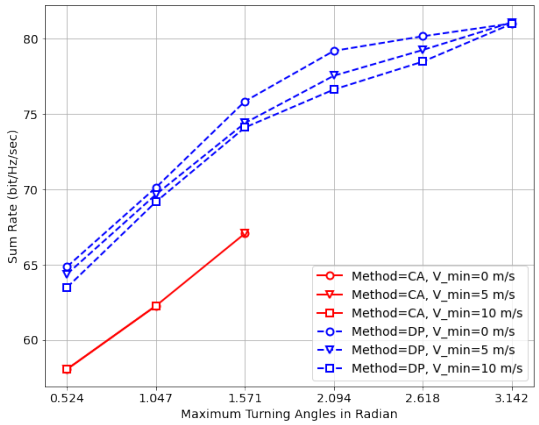

Fig. 4.14 Overall sum rate versus maximum turning angle in radian with different minimum speeds for SCA and SDP methods.

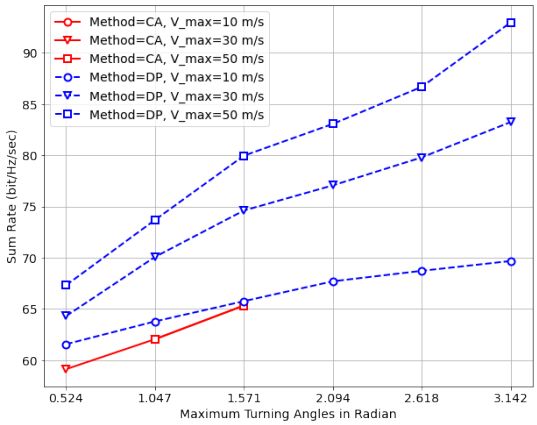

Fig. 4.15 Overall sum rate versus maximum turning angle in radian with different maximum speeds for SCA and SDP methods.

terms of much lower test MSE loss. For the LSTM baseline, a higher masking probability does not always help reduce the MSE loss on spoofing probability cases. That is, using a higher masking probability worsens the MSE loss on smaller spoofing probability cases. However, for our proposed DART model, a higher masking probability always helps reduce the MSE loss significantly which makes the model more robust to the GNSS spoofing attack.

Figure. 4.10 depicts the predicted 2D user trajectories comparison between DART and LSTM baseline. We assume the GNSS spoofing attacker opportunistically modifies the user location information by uniformly sampling a location within the service area with probability $p_s = 0.3$. It can be seen from the figure that the prediction of DART is closer to the ground-truth trajectory than the prediction of the LSTM baseline. This demonstrates that the DART model is more robust to GNSS spoofing attacks and can generate reliable user trajectory predictions.

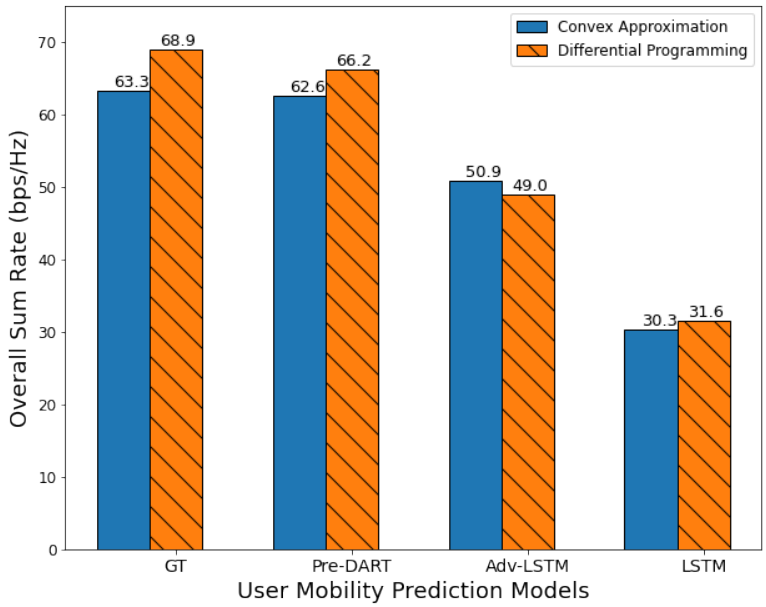

Fig. 4.16 Overall sum rate comparison between pre-trained DART, adversarial trained LSTM baseline, LSTM baseline from scratch with underlying SCA and SDP optimisation methods, where GT refers to ground-truth, Pre-DART refers to pre-trained DART model, Adv-LSTM refers to adversarial trained LSTM baseline model and LSTM is the LSTM baseline trained from scratch, respectively.

Figure. 4.11 shows the Gaussian filter smoothed MSE losses on the test set comparison between the DART teacher model, the distilled GRU student model and the GRU trained from scratch. It can be seen that the GRU trained by knowledge distillation can predict the next user locations as accurately as the teacher model under GNSS spoofing attacks whilst the GRU trained from scratch fails to converge. This demonstrates the effectiveness of our proposed knowledge distillation-based training method.

Figure. 4.12 compares the inference time between the DART teacher model and the distilled GRU student model. The inference time is calculated by converting the Pytorch models into Open Neural Network Exchange (ONNX) which is a widely used framework to optimise model speed for real-time inference. It can be seen that the GRU model is almost 25 times faster than the teacher model by maintaining a very close predictive power. The reason is that the teacher model requires storing all historical time slots for inference and the attention mechanism inside the model has a computational complexity of $O(W^2)$ since it cannot do recurrent inference like GRU, where W is the window size of the stored sequence. Thus, the overall computational complexity is $O(W^2)$ for the Transformer and $O(1)$ for GRU for a single-step inference.

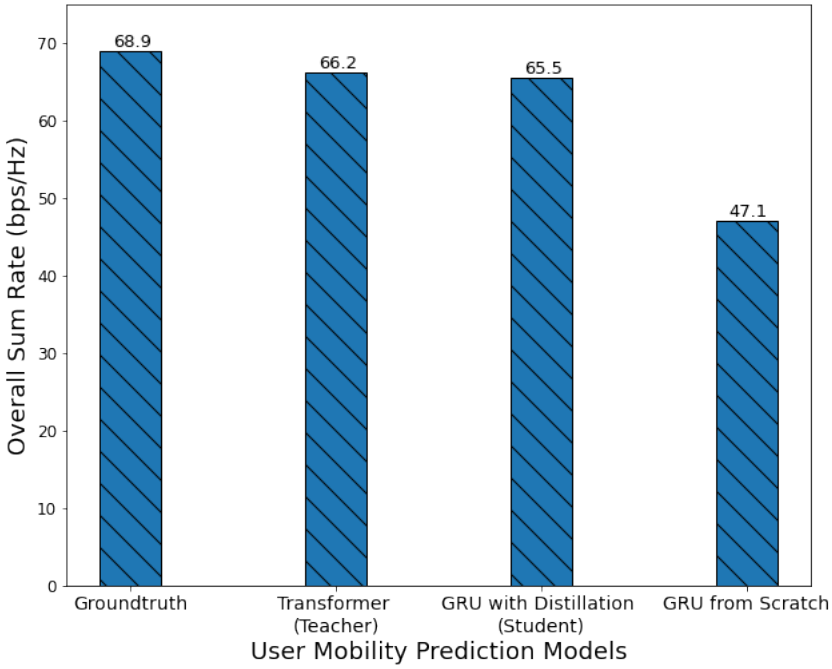

Fig. 4.17 Differential programming optimised sum rate comparison between Transformer (teacher), GRU with distillation (student) and GRU from scratch.

4.5.2 User Clustering and UAV Position Optimisation Performance Analysis

Figure. 4.13 shows the loss, sum rate and penalty comparison between different gradient descent algorithms for the proposed SDP method, i.e., Adam and RMSprop algorithms. In the simulation, we have found the standard gradient descent method fails to reduce the penalty to zero and is very sensitive to the learning rate. Therefore, we don't include it in the comparison. It can be seen that the RMSprop algorithm shows superior convergence speed and performance than the Adam algorithm with the same learning rate. From the figure of the sum rate, we can see that the derivative of the sum rate curve first decreases and then increases at around 1000 epochs. The flat plane at around 1000 epoch is a local optimum, however, our proposed SDP method can easily escape from the local optimum and lead the optimisation toward a better direction. Equation (4.36) shows that the loss consists of two terms, i.e., the opposite sum rate term $H[k]$ and the penalty term $P_{\text{penalty}}[k]$. From Eq. (4.38), we can see that the penalty term is non-negative due to the ramp function. Therefore, the negative loss function is caused by the opposite sum rate term in Eq. (4.37), which is reasonable since we want to maximise the sum rate (i.e., minimise the opposite number of sum rate). It is worth noting that a negative loss does not indicate that the

moving average of the squared gradients, i.e., the RMS value, is negative. The RMS value in the RMSprop algorithm remains positive and the algorithm points in the direction of decreasing loss by looking at the direction of the gradient.

Figure. 4.14 and Fig. 4.15 illustrate the overall sum rate versus maximum turning angle with different minimum and maximum speeds for SCA and SDP methods, respectively. The curves of the SCA method in both figures are all partial since the SCA method can only generate a valid solution with a maximum turning angle smaller than $\pi/2$. However, our proposed SDP method is valid for all values of maximum turning angles. It can be seen from both figures that the sum rates of the SDP method in both figures are much higher than the sum rates of the SCA method where the sum rate of the maximum turning angle of π is the maximum.

4.5.3 Deep Learning-Based Prediction-Optimisation Scheme Performance Analysis

Figure. 4.16 shows the overall sum rate comparison between pre-trained DART, adversarial trained LSTM baseline, and LSTM baseline from scratch with underlying SCA and SDP optimisation methods, where GT refers to ground-truth, Pre-DART refers to pre-trained DART model, Adv-LSTM refers to adversarial trained LSTM baseline model and LSTM is the LSTM baseline trained from scratch, respectively. Specifically, we apply each prediction model to the user mobility prediction. The predicted user locations in the next time slot are then used for user clustering and UAV position optimisation. Once the users for each UAV are arranged and the UAV position is optimised, we measure the overall sum rate of the system based on the ground-truth user location in the next time slot. It can be seen that the overall sum rate of our proposed Pre-DART model is very close to the optimal value, i.e., the overall sum rate measured by using the ground-truth user locations. Meanwhile, the SDP method outperforms the SCA method in GT and Pre-DART model-based optimisation results but gives a similar performance on Adv-LSTM and LSTM baseline models. This demonstrates that the SDP method can give more accurate UAV position optimisation results with more accurate predicted user locations. To conclude, our proposed deep learning-based prediction-optimisation scheme with the Pre-DART model and SDP method can provide up to 30% higher overall sum rate compared with the Adv-LSTM baseline and almost double the overall sum rate compared with the LSTM baseline.

Figure. 4.17 shows the overall sum rate comparison between the groundtruth user locations, the DART Transformer teacher model, the distilled GRU model and the GRU model trained from scratch with underlying SDP optimisation methods. Specifically, we apply each prediction model to the user mobility prediction. The predicted user locations

in the next time slot are then used for user clustering and UAV position optimisation. Once the users for each UAV are arranged and the UAV position is optimised, we measure the overall sum rate of the system based on the ground-truth user location in the next time slot. It can be seen that the overall sum rate of the distilled GRU student model is almost 99% close to the teacher model and both of them are close to the optimal value, i.e., the overall sum rate measured by using the ground-truth user locations. However, the sum rate optimised from the predictions of GRU trained from scratch is much worse than the other three.

4.6 Conclusion

In this chapter, our focus has been on innovating a pioneering deep learning-centric approach tailored to predict user mobility, optimise user assignments, and strategically position drones within a UAV swarm-enabled wireless communication system, all while confronting the challenges posed by malicious GNSS spoofing attackers. A cornerstone of our proposal lies in deploying a robust deep learning model, instrumental in forecasting future user locations. Leveraging this prediction capability, we devise a method to optimise user assignments and determine UAV positions preemptively using efficient optimisation techniques.

The simulation outcomes serve as a testament to the efficacy of our proposed DART model. It exhibits near-optimal predictive performance even under diverse GNSS spoofing attack scenarios. Furthermore, our introduced SDP method emerges as a game-changer, surpassing the commonly employed SCA method. It not only yields superior objective values but also boasts faster convergence rates, establishing its superiority in the optimisation landscape.

Demonstrating the robustness of our framework, the deep learning-driven prediction-optimisation scheme showcases near-optimal overall sum rates compared to scenarios leveraging ground-truth user location information solely for optimisation purposes. The pre-trained and fine-tuned DART model, coupled with the SDP method, attains remarkable performance heights, showcasing up to a 30% increase in overall sum rates in contrast to an adversarially trained LSTM baseline. Moreover, this approach nearly doubles the overall sum rates when compared to the vanilla LSTM baseline, underlining its substantial leap in performance metrics.

While the Transformer-based user location forecasting algorithm introduced exhibits superior predictive prowess, its inference speed might pose constraints for real-world applicability. Addressing this concern, we delve into the introduction of a knowledge distillation methodology. This strategic approach aims to streamline the complexity of

the Transformer-based forecasting algorithm while retaining its predictive power. The overarching goal is to distil the extensive DART model into a more manageable GRU model, maintaining predictive efficacy while significantly reducing computational overhead.

The simulation outcomes serve as compelling evidence for the efficacy of our proposed Knowledge Distillation-based scheme. We witness that the optimised sum rates achieved using the distilled GRU student model's predicted user locations nearly approach an impressive 99% parity with the Transformer teacher model. Moreover, a significant highlight is the striking difference in inference time between the two models, where the student model operates at an incredibly swift pace, consuming a mere 4% of the time required by the teacher model for inference.

This remarkable achievement underscores the potential of knowledge distillation in streamlining complex models without compromising predictive accuracy. Not only does our distilled GRU student model approximate the performance of the Transformer teacher model, but its expeditious inference time renders it a more viable and practical choice for real-time applications within UAV swarm-enabled communication systems.

The amalgamation of prediction accuracy and computational efficiency achieved through knowledge distillation presents a pivotal advancement in the realm of user mobility prediction and resource allocation. By distilling the extensive knowledge encapsulated within the Transformer-based teacher model into the streamlined GRU-based student model, we have successfully struck a balance between predictive capability and computational expediency.

Chapter 5

Conclusions and Potential Research Directions

5.1 Conclusions

The thesis has delved into the pressing challenges faced by modern wireless networks, highlighting the critical need for sustainable and secure communication protocols due to escalating demands for connectivity and spectral efficiency, alongside the growing threat of malicious attacks. Focusing on two vital domains within wireless communication systems - IRS-aided THz communication and UAV swarm-enabled networks - the research has aimed to address energy inefficiencies and fortify against security threats. Firstly, it has introduced a cutting-edge deep learning-based algorithm, the TE-CIE, designed to predict channel behaviour and optimise energy efficiency in IRS-aided THz communication systems, showcasing substantial improvements in accuracy and computational efficiency. Secondly, it has presented a robust UAV swarm position optimisation system utilising a novel deep learning framework, the DART, which significantly mitigates the impact of malicious GNSS spoofing attacks. Additionally, the thesis has explored knowledge distillation techniques to streamline computational complexities, resulting in a smaller yet efficient model, the GRU, facilitating real-time deployment in UAV swarm-based networks while maintaining considerable performance. Overall, the research has contributed innovative solutions for sustainable and secure wireless communication paradigms, emphasising the significance of these advancements in enhancing energy efficiency, resilience against attacks, and overall performance compared to established benchmarks, while paving the way for future research in diverse wireless communication scenarios.

Specifically, in Chapter 3, we have investigated a novel deep learning-based channel prediction and EE optimisation problem for an IRS-assisted THz communication system. We have designed a deep learning-based channel prediction method for time-varying

fading channel prediction. A TE-CIE model has been developed to accurately capture the temporal correlation between past CSI and the next CSI. Meanwhile, the EE optimisation problem has been studied in an IRS-assisted MU-MISO system with THz communications. In the considered system, the precoding matrix and IRS phase shift matrix have jointly been optimised for maximising the system EE when meeting the constraint of maximum transmit power. Finally, combining the TE-CIE channel prediction method with the EE optimisation algorithm leads to our proposed deep learning-based prediction-optimisation scheme for EE maximisation in the IRS-assisted THz MU-MISO communication system. We have shown in the simulation that our proposed scheme can achieve at least twice the EE improvement compared to baseline methods in the literature.

In Chapter 4, we have proposed a novel deep learning-based user mobility prediction, user assignment and drone position optimisation scheme for a UAV swarm-enabled wireless communication system in the presence of malicious GNSS spoofing attackers. A robust deep learning model is deployed to predict the future user locations and the user assignment and UAV positions for the next time slot can be optimised in advance by efficient optimisation methods. Simulation results demonstrate that our proposed DART model can achieve near-optimal prediction performance under various GNSS spoofing attack settings. Meanwhile, the proposed SDP method significantly outperforms the commonly used SCA method with better objective values and faster convergence. Finally, the deep learning-based prediction-optimisation scheme is proven to achieve a near-optimal overall sum rate compared with using the ground-truth user location information for optimisation. The pre-trained and fine-tuned DART model with the SDP method can provide up to 30% higher overall sum rate compared with the adversarial trained LSTM baseline and almost double the overall sum rate compared with the vanilla LSTM baseline.

Finally, in Chapter 5, we have proposed a novel knowledge distillation-based user mobility prediction, user assignment and drone position optimisation scheme for a UAV swarm-enabled wireless communication system in the presence of malicious GNSS spoofing attackers. A robust and efficient deep learning model is deployed to predict the future user locations and the user assignment and UAV positions for the next time slot can be optimised in advance by efficient optimisation methods. Simulation results demonstrate that the optimised sum rate using the distilled GRU student model's predicted user locations can achieve almost 99% compared to the Transformer teacher model. Meanwhile, the inference time of the student model is only 4% compared to the teacher model.

5.2 Comprehensive Future Research Directions

This thesis has covered topics including the imperative need for sustainable and secure communication protocols in modern wireless networks, advancements in IRS-aided THz communication systems, strategies to mitigate energy inefficiencies, and fortification against malicious attacks in wireless networks, particularly focusing on UAV swarm-enabled communication networks. However, there are still several challenges that need to be addressed to extend and improve the scope of the current research. Some of the points are listed as follows:

- **Adaptive Channel Prediction:** Enhancing adaptive channel prediction in IRS-aided THz systems might involve exploring reinforcement learning or other adaptive techniques that allow the system to learn and adapt to changing channel conditions in real time, thereby improving the accuracy of channel predictions.
- **Real-Time Implementation:** To validate the practicality of deep learning models like DART and TE-CIE, future research could focus on optimising these models for deployment in real-time scenarios. This could involve hardware acceleration, model compression techniques, and efficient model architectures to ensure their effectiveness in dynamic environments.
- **Energy Harvesting Techniques:** Exploring and integrating novel energy harvesting techniques (solar, kinetic, RF, etc.) could be pivotal for sustaining UAV swarms and IRS-enabled systems. Future research might focus on optimising energy harvesting systems and energy storage solutions to maximise autonomy and operational lifetime, which can also be enhanced by deep learning algorithms.
- **Integration with 6G Networks:** Investigating integration with emerging 6G networks involves assessing compatibility, scalability, and efficiency. Future studies might explore how the proposed solutions align with the architectural principles and technological advancements envisioned for 6G networks.
- **Edge Computing and Decentralisation:** The system models studied in this thesis are all centralised. However, exploring the role of edge computing and decentralised architectures in wireless communication systems can contribute to increased security and efficiency. Future research could investigate how decentralised decision-making and edge processing can improve latency, privacy, and resilience in UAV swarm networks.
- **Multi-Objective Optimisation:** Investigating multi-objective optimisation techniques would involve finding optimal solutions that balance conflicting objectives

such as energy efficiency, spectral efficiency, and security. Future research could explore sophisticated algorithms, such as multi-objective evolutionary algorithms or game-theoretic approaches, to handle trade-offs effectively.

- **Dynamic Security Measures:** Future research could focus on developing adaptive security measures that dynamically evolve to counter emerging threats. This might involve Artificial Intelligence (AI)-driven threat detection systems that continuously learn from new attack patterns, adapting network configurations in real-time to mitigate potential vulnerabilities in UAV swarm networks.

- **Resource Allocation Optimisation:** Advanced resource allocation algorithms for UAV swarm communication systems could account for dynamic user mobility patterns and varying environmental conditions. Future research might delve into deep learning-based predictive algorithms that anticipate user movement or adaptive algorithms that dynamically allocate resources based on changing network conditions.

- **Standardisation and Deployment Strategies:** Addressing standardisation challenges involves defining protocols and interfaces that ensure interoperability among diverse systems. Future research could focus on developing deployment strategies that consider regulatory frameworks, scalability, and ease of integration to facilitate the practical implementation of these advanced technologies in real-world scenarios.

- **Privacy-Preserving Communication Protocols:** Security issues include not only malicious attacks but also unauthorised access or surveillance. Therefore, future research could focus on investigating privacy-preserving communication protocols for wireless networks, focusing on developing robust encryption techniques and anonymisation methods to safeguard sensitive data exchanged within UAV swarm-enabled networks, ensuring confidentiality and integrity while minimising the risk of unauthorised access or surveillance.

References

[1] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[2] ——, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[3] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning," *Ieee Network*, vol. 33, no. 5, pp. 156–165, 2019.

[4] L. R. Medsker, L. Jain *et al.*, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[5] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[8] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1546–1577, 2021.

[9] S. Gong, X. Lu, D. T. Hoang, D. Niyato, L. Shu, D. I. Kim, and Y. Liang, "Towards smart wireless communications via intelligent reflecting surfaces: A contemporary survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2020.

[10] Z. Chen, X. Ma, B. Zhang, Y. Zhang, Z. Niu, N. Kuang, W. Chen, L. Li, and S. Li, "A survey on terahertz communications," *China Communications*, vol. 16, no. 2, pp. 1–35, 2019.

[11] V. Petrov, A. Pyattaev, D. Moltchanov, and Y. Koucheryavy, "Terahertz band communications: Applications, research challenges, and standardization activities," in *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2016, pp. 183–190.

[12] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[13] L. Bariah, L. Mohjazi, S. Muhaidat, P. C. Sofotasios, G. K. Kurt, H. Yanikomeroglu, and O. A. Dobre, "A prospective look: Key enabling technologies, applications and open research topics in 6G networks," *IEEE Access*, vol. 8, pp. 174 792–174 820, 2020.

[14] Z. Xiao, L. Zhu, Y. Liu, P. Yi, R. Zhang, X.-G. Xia, and R. Schober, "A survey on millimeter-wave beamforming enabled uav communications and networking," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 557–610, 2022.

[15] Y. Zhou, B. Rao, and W. Wang, "Uav swarm intelligence: Recent advances and future trends," *IEEE Access*, vol. 8, pp. 183 856–183 878, 2020.

[16] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[17] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019.

[18] E. Schmidt, N. Gatsis, and D. Akopian, "A gps spoofing detection and classification correlator-based technique using the lasso," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4224–4237, 2020.

[19] A. Shafique, A. Mehmood, and M. Elhadef, "Detecting signal spoofing attack in uavs using machine learning models," *IEEE Access*, vol. 9, pp. 93 803–93 815, 2021.

[20] B. Pardhasaradhi and L. R. Cenkeramaddi, "Gps spoofing detection and mitigation for drones using distributed radar tracking and fusion," *IEEE Sensors Journal*, vol. 22, no. 11, pp. 11 122–11 134, 2022.

[21] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[22] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[23] W. Chen, X. Ma, Z. Li, and N. Kuang, "Sum-rate maximization for intelligent reflecting surface based terahertz communication systems," in *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, 2019, pp. 153–157.

[24] Y. Pan, K. Wang, C. Pan, H. Zhu, and J. Wang, "Sum-rate maximization for intelligent reflecting surface assisted terahertz communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3320–3325, 2022.

[25] Q. Wu, Y. Zhang, C. Huang, Y. Chau, Z. Yang, and M. Shikh-Bahaei, "Energy efficient intelligent reflecting surface assisted terahertz communications," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[26] Y. Wu, W. Fan, W. Yang, X. Sun, and X. Guan, "Robust trajectory and communication design for multi-uav enabled wireless networks in the presence of jammers," *IEEE Access*, vol. 8, pp. 2893–2905, 2020.

[27] C. Shen, T.-H. Chang, J. Gong, Y. Zeng, and R. Zhang, "Multi-uav interference coordination via joint trajectory and power control," *IEEE Transactions on Signal Processing*, vol. 68, pp. 843–858, 2020.

[28] Z. Chang, H. Deng, L. You, G. Min, S. Garg, and G. Kaddoum, "Trajectory design and resource allocation for multi-uav networks: Deep reinforcement learning approaches," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2022.

[29] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[30] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4157–4170, 2019.

[31] C. Huang, A. Zappone, M. Debbah, and C. Yuen, "Achievable rate maximization by passive intelligent mirrors," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 3714–3718.

[32] X. Lei, M. Wu, F. Zhou, X. Tang, R. Q. Hu, and P. Fan, "Reconfigurable intelligent surface-based symbiotic radio for 6G: Design, challenges, and opportunities," *IEEE Wireless Communications*, 2021.

[33] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar *et al.*, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," *arXiv preprint arXiv:2206.00422*, 2022.

[34] A. Bhowal and S. Aissa, "RIS-aided communications in indoor and outdoor environments: Performance analysis with a realistic channel model," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2022.

[35] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4157–4170, 2019.

[36] Z. Yang, M. Chen, W. Saad, W. Xu, M. Shikh-Bahaei, H. V. Poor, and S. Cui, "Energy-efficient wireless communications with distributed reconfigurable intelligent surfaces," *IEEE Transactions on Wireless Communications*, vol. 21, no. 1, pp. 665–679, 2021.

[37] W. Ni, Y. Liu, Z. Yang, H. Tian, and X. Shen, "Federated learning in multi-RIS aided systems," *IEEE Internet of Things Journal*, 2021.

[38] K. M. S. Huq, S. A. Busari, J. Rodriguez, V. Frascolla, W. Bazzi, and D. C. Sicker, "Terahertz-enabled wireless system for beyond-5g ultra-fast networks: A brief survey," *IEEE Network*, vol. 33, no. 4, pp. 89–95, 2019.

[39] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang, and D. Zhang, "A survey on green 6G network: Architecture and technologies," *IEEE Access*, vol. 7, pp. 175 758–175 768, 2019.

[40] C. Lin and G. Y. Li, "Indoor terahertz communications: How many antenna arrays are needed?" *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3097–3107, 2015.

[41] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in uav communication networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016.

[42] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for uav communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018.

[43] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 75–86.

[44] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "Gps vulnerability to spoofing threats and a review of antispoofing techniques," *International Journal of Navigation and Observation*, vol. 2012, 2012.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[46] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[48] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[49] D. Zhang, Y. Yu, C. Li, J. Dong, D. Su, C. Chu, and D. Yu, "Mm-llms: Recent advances in multimodal large language models," *arXiv preprint arXiv:2401.13601*, 2024.

[50] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[51] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[52] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*, 2020.

[53] S. Zhang, S. Xu, G. Y. Li, and E. Ayanoglu, "First 20 years of green radios," *arXiv preprint arXiv:1908.07696*, 2019.

[54] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[55] Q. Wu, G. Y. Li, W. Chen, D. W. K. Ng, and R. Schober, "An overview of sustainable green 5G networks," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 72–80, 2017.

[56] G. Y. Li, Z. Xu, C. Xiong, C. Yang, S. Zhang, Y. Chen, and S. Xu, "Energy-efficient wireless communications: Tutorial, survey, and open issues," *IEEE Wireless Commun.*, vol. 18, no. 6, pp. 28–35, Dec. 2011.

[57] S. Buzzi, I. Chih-Lin, T. E. Klein, H. V. Poor, C. Yang, and A. Zappone, "A survey of energy-efficient techniques for 5G networks and challenges ahead," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 697–709, Apr. 2016.

[58] R. Q. Hu and Y. Qian, "An energy efficient and spectrum efficient wireless heterogeneous network framework for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 94–101, 2014.

[59] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network: Joint active and passive beamforming design," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[60] L. Wei, C. Huang, G. C. Alexandropoulos, W. E. I. Sha, Z. Zhang, M. Debbah, and C. Yuen, "Multi-user holographic MIMO surfaces: Channel modeling and spectral efficiency analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 5, pp. 1112–1124, 2022.

[61] Y. Huo, X. Dong, and N. Ferdinand, "Distributed reconfigurable intelligent surfaces for energy efficient indoor terahertz wireless communications," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[62] L. Fan, Q. Wang, Y. Huang, and L. Yang, "Performance analysis of low-complexity channel prediction for uplink massive MIMO," *IET Communications*, vol. 10, no. 14, pp. 1744–1751, 2016.

[63] A. Duel-Hallen, "Fading channel prediction for mobile radio adaptive transmission systems," *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2299–2313, 2007.

[64] K. Baddour and N. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1650–1662, 2005.

[65] C. Min, N. Chang, J. Cha, and J. Kang, "MIMO-OFDM downlink channel prediction for IEEE802.16e systems using kalman filter," in *2007 IEEE Wireless Communications and Networking Conference*, 2007, pp. 942–946.

[66] R.-F. Liao, H. Wen, J. Wu, H. Song, F. Pan, and L. Dong, "The rayleigh fading channel prediction via deep learning," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[67] W. Jiang and H. D. Schotten, "Recurrent neural network-based frequency-domain channel prediction for wideband communications," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, 2019, pp. 1–6.

[68] W. Xu, J. An, Y. Xu, C. Huang, L. Gan, and C. Yuen, "Time-varying channel prediction for RIS-assisted MU-MISO networks via deep learning," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2022.

[69] P. Dent, G. E. Bottomley, and T. Croft, "Jakes fading model revisited," *Electronics letters*, vol. 29, no. 13, pp. 1162–1163, 1993.

[70] N. Hansen, "The CMA evolution strategy: A tutorial," *CoRR*, vol. abs/1604.00772, 2016. [Online]. Available: http://arxiv.org/abs/1604.00772

[71] A. A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE Journal on selected areas in communications*, vol. 5, no. 2, pp. 128–137, 1987.

[72] Y. Fan and J. Thompson, "MIMO configurations for relay channels: Theory and practice," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1774–1786, 2007.

[73] C. Xiao, Y. R. Zheng, and N. C. Beaulieu, "Novel sum-of-sinusoids simulation models for rayleigh and rician fading channels," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3667–3679, 2006.

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[75] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[76] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[77] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[78] C.-S. J. Chu, "Time series segmentation: A sliding window approach," *Information Sciences*, vol. 85, no. 1-3, pp. 147–173, 1995.

[79] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[80] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[81] S. Schaible, "Fractional programming. II, on dinkelbach's algorithm," *Management science*, vol. 22, no. 8, pp. 868–873, 1976.

[82] A. Lin, "Binary search algorithm," *WikiJournal of Science*, vol. 2, no. 1, pp. 1–13, 2019.

[83] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-uav enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[84] A. Fotouhi, M. Ding, L. Galati Giordano, M. Hassan, J. Li, and Z. Lin, "Joint optimization of access and backhaul links for uavs based on reinforcement learning," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[85] Q. Hou, Y. Cai, Q. Hu, M. Lee, and G. Yu, "Joint resource allocation and trajectory design for multi-uav systems with moving users: Pointer network and unfolding," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022.

[86] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of field robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[87] P. Bethi, S. Pathipati, and P. Aparna, "Impact of target tracking module in gps spoofer design for stealthy gps spoofing," in *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE, 2020, pp. 1–6.

[88] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[89] L. Pappalardo, F. Simini, S. Rinzivillo, D. Pedreschi, F. Giannotti, and A.-L. Barabási, "Returners and explorers dichotomy in human mobility," *Nature communications*, vol. 6, no. 1, pp. 1–8, 2015.

[90] E. Chaalal, L. Reynaud, and S. M. Senouci, "Mobility prediction for aerial base stations for a coverage extension in 5g networks," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 2021, pp. 2163–2168.

[91] A. B. Adege, H.-P. Lin, and L.-C. Wang, "Mobility predictions for iot devices using gated recurrent unit network," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 505–517, 2020.

[92] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[93] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 75–86.

[94] ——, "On the requirements for successful gps spoofing attacks," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 75–86.

[95] B. Pardhasaradhi, P. Srihari, and P. Aparna, "Spoofer-to-target association in multi-spoofer multi-target scenario for stealthy gps spoofing," *IEEE Access*, vol. 9, pp. 108 675–108 688, 2021.

[96] S. Z. Khan, M. Mohsin, and W. Iqbal, "On gps spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions," *PeerJ Computer Science*, vol. 7, p. e507, 2021.

[97] P. Bethi, S. Pathipati, and P. Aparna, "Stealthy gps spoofing: Spoofer systems, spoofing techniques and strategies," in *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE, 2020, pp. 1–7.

[98] X. Lin, V. Yajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H.-L. Maattanen, M. Bergstrom, S. Euler, and Y.-P. E. Wang, "The sky is not the limit: Lte for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 204–210, 2018.

[99] U. Challita and W. Saad, "Network formation in the sky: Unmanned aerial vehicles for multi-hop wireless backhauling," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[100] A. Ben-Tal and A. Nemirovski, "Selected topics in robust convex optimization," *Mathematical Programming*, vol. 112, pp. 125–158, 2008.

[101] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[102] L. Zhang, Y.-C. Liang, Y. Xin, and H. V. Poor, "Robust cognitive beamforming with partial channel state information," *IEEE Transactions on Wireless Communications*, vol. 8, no. 8, pp. 4143–4153, 2009.

[103] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" *Advances in neural information processing systems*, vol. 31, 2018.

[104] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[105] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 1, pp. 57–77, 2016.

[106] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[107] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[108] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[109] E. Tzoreff and A. J. Weiss, "Path design for best emitter location using two mobile sensors," *IEEE Transactions on Signal Processing*, vol. 65, no. 19, pp. 5249–5261, 2017.

[110] H. Wang, J. Wang, G. Ding, J. Chen, and J. Yang, "Completion time minimization for turning angle-constrained uav-to-uav communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4569–4574, 2020.

[111] A. Feldstein and P. Turner, "Overflow, underflow, and severe loss of significance in floating-point addition and subtraction," *IMA journal of numerical analysis*, vol. 6, no. 2, pp. 241–251, 1986.

[112] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[113] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 2016, pp. 265–283.

[114] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[115] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[116] W. Zhang, "Branch-and-bound search algorithms and their computational complexity." University of Southern California Marina Del Rey Information Sciences INST, Tech. Rep., 1996.

[117] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear algebra and its applications*, vol. 284, no. 1-3, pp. 193–228, 1998.