2020·广州
世界5G大会
WORLD 5G CONVENTION

DATA DRIVEN AND INTENT AWARE SMART WIRELESS NETWORK

WHITE PAPER V7.0 A
2020.11

Data Driven and Intent Aware Smart Wireless Network 2020

Executive Summary

To accommodate a wide range of scenarios and service requirements, the future networks (i.e. 5G+ and 6G) are expected to become more agile, and inevitably, more complicated, thus posing great challenges to network management and radio resource optimization. Intelligence is expected to be introduced in wireless network for all the domains and all levels, from terminal, local, edge to the cloud. Data analytics, machine learning, artificial intelligence and intent expression are identified as the key drivers for the intelligence evolution and revolution in the wireless network. Data driven and intent aware smart wireless network usage scenarios and network architecture are being heatedly discussed and investigated recently both in the wireless network industry and academia. “Smart and Simplicity” become the industry consensus for 5G and future network. “Smart” requires the network to dynamically adapt to the diversified scenarios and services to efficiently boost both spectrum efficiency, energy efficiency and offer the consistency the user experience according to the network intents of network operators or the business intents of vertical users. “Simplicity” poses the requirements of network automation and autonomy to significantly reduce the network operation and maintenance human labor and cost by applying a concise intent expression model.

This white paper investigates the data driven and intent aware smart wireless network. It is an update and evolution of the previous whitepapers “Wireless Big Data for Smart 5G” and “Wireless Big Data and AI for Smart 5G & Beyond ”, “Data Driven and Intent Aware Smart Wireless Network 2019” where the reference architecture and use case are comprehensively studied. In this white paper, Some new use case have been introduced, for example, terminal intelligence, with increasing capabilities of mobile terminals, for example, integration of GPU or NPU, part of AI processing can be offloaded from network to terminals not only to reduce cloud processing cost, but also to achieve more accurate or efficient processing since user terminals have better understanding user specific radio channel. Furthermore, extensive testing and experimental results of network management optimization, MEC optimization, have been revealed by industry companies to prove the benefits of data driven network. In addition, more in-depth AI algorithms and more comprehensive scenarios of network slicing optimization and radio transmission technologies have been studied.

Furthermore, intent aware wireless network is proposed and investigated to achieve the network “smart” and “simplicity” leveraging the data driven network intelligence. A use case of intent-driven mobility load balancing is taken as an example to illustrate the integration of intent aware wireless network and data driven wireless network. Based on the previous logical architecture, the

implementation architecture of intent driven network is investigated with discussion of two enabling technologies, intent knowledge base and cross-layer intent collaboration. Besides, technical key issues, challenges and the intent service evolution roadmap are presented. Last but not the least, the standardization progress on intent driven network are discussed.

We wish this white paper can provide readers with some valuable insights for industry on the data driven and intent aware smart wireless network and help to accelerate the process of smart wireless network.

Table of Contents

1 Smart Wireless Network

1.1 Data Driven Wireless Network

5G commercialization is becoming a hot topic all over the world and data driven network optimization is gradually mature and widely applied in every aspect of network rollout, maintenance and optimization.

Currently diverse multiple networks have co-existed, e.g. 2G, 3G, 4G, 5G, and network management and optimization is a huge burden for operators, for example, exponentially increasing number of parameters need to be optimized and adjusted timely not only for new base station planning, coverage optimization, but also for anomaly detection and root cause analysis. The introduction of AI related technologies have demonstrated great benefits to improve efficiency and relieve human efforts. In this white paper, extensive testing and experimental results have been revealed by industry companies to prove the benefits of data driven network.

With increasing capabilities of mobile terminals, for example, integration of GPU or NPU in Terminal, mobile terminals is evolved from a smart phone to an intelligent phone. Part of AI processing can be offloaded from network to terminals not only to reduce cloud processing cost, but also to achieve more accurate or efficient processing since user terminals have better understanding user specific radio channel. With the assistance of user terminal, network can better configure and control resource allocation.

Mobile edge computing is a vitally important use case for 5G and even 6G. By utilizing powerful AI processing capabilities on edge, the user plane content can be analyzed in a timely manner to help achieve accurate edge caching policy. Some specific examples and algorithms have been proposed in this white paper.

AI applications have also been experimented in PHY layer optimization and will play an important role in 6G. AI enabled radio transmission technologies, e.g. deep learning aided channel estimation, MIMO detection, have been explored in this white paper. A number of AI algorithms and experiments have been conducted by academics and positive results have been achieved.

In addition, SDOs continue to standardize data driven smart wireless network, for example, ITU-T ML5G, 3GPP SA1/2/5/RAN3 and etc. Great progress had been made in both standardization and test/trials/application in the commercial network. The main updates and progress are described in detail in the section 2 and section 4.

1.2 Intent Aware Wireless Network

In intent aware wireless network, the intent users (e.g. BSS/OSS) uses a controlled language (e.g. domain specific language, DSL) to express the business/network goals, without describing how to accomplish the goals. Such an intent aware wireless network could help the network operators to simply the network management and improve the network automatic capability (e.g. close loop). In case of 5G vertical industry, intent aware wireless network could be introduced to meet with the business intents of the vertical industry operators without deeply understanding the wireless network details. Such vertical industry system or user submit a business intent which describes the specific business goals using the dedicated vertical industry business terms, without having to use network-related language. Then the intent system of the intent aware wireless network translates the business intent to one or more network commands and delivers them to the wireless network entities ensuring the business intent goal to be achieved.

Use cases, logic architecture and expression model for intent aware wireless network has been discussed in V6.0 of this white paper. In this paper, we will further discuss on the following topics:

- Reference implementation architecture. An enhanced ENI architecture for intent aware wireless network is proposed. The functional blocks, external and internal reference points, as well as exchanged data and information will be provided.
- Intent API. The concept of intent has been proposed for several years, but it has not been commercially applied in network management interfaces as far as we know. A brief overview of the composition of intent API is described, which includes operations for intent lifecycle management and intent schema management.
- Use case. An intent-driven MLB (IDMLB) method is proposed to reduce handover times of UEs during load-balancing. It take both network intent (e.g., offloading) and UE intent (e.g., bandwidth) into consideration to design a more fine-grained handover scheme and avoid the deterioration of the QoE (Quality of Experience) after handover.
- Key technologies. Two enabling technologies are introduced. One is intent knowledge base which is used for experimental and automated intent translation. The other is cross-layer intent collaboration which is used to improve the efficiency of intent implementation.
- Technical key issues and challenges. A lot of unsolved problems for intent aware wireless network need to be further studied. Several examples are given in this paper, which include the challenges for intent

API & language evolution, AI integration and business intent alignment.

2 Data Driven Wireless Network

2.1 Overall Introduction

Three white papers have been published in 2017, 2018 and 2019.[1-4] on data driven smart wireless network. These white papers introduced the use cases, potential solutions, reference architecture, procedure and platform designs on the wireless big data and AI driven network. Based on the previous versions of the white paper, this white paper mainly focus on the newly revealed results of previous use cases and new use cases.

In this section, ten use cases are described to update the progress and further demonstrate promising research topics of data driven smart wireless network from four aspects: network management optimization, MEC optimization, network resource optimization, radio transmission technologies and terminal intelligence.

2.2 Network Management Optimization

2.2.1 Use case 1: Anomaly Detection with KPIs and Root Cause Analysis

(1) Introduction

The emerging fifth generation (5G) network has the potential to satisfy the rapidly growing traffic demand and promote the transformation of smartphone-centric networks into an Internet of Things (IoT) ecosystem. Due to the introduction of new communication technologies and the increased density of 5G cells, the complexity of operation and operational expenditure (OPEX) will become very challenging in 5G.

The traditional static threshold method cannot meet the needs for anomaly detection in wireless networks, the major system faults were detected by customer complaints. The solution of the problem depends on manual analysis, the professional requirements of technical personnel are high, difficult to solve difficult problems quickly.

In this paper, we focus on the intelligent operation of wireless network through ML algorithms. One use cases are also studied to use ML algorithms to automate the anomaly detection and fault diagnosis of key performance indicators (KPIs) in wireless networks. The effectiveness of the proposed ML algorithms is demonstrated by the real data experiments, thus encouraging the further research for intelligent wireless network operation.

The paper is organized as follows. In Section 1, we illustrate the scheme design of our system, which implements ML algorithms to automate the anomaly detection and fault diagnosis of key performance indicators (KPIs) in wireless networks. One typical use case for on-site data analysis is shown in Section 2. Finally, we draw the

conclusions in Section 3.

(2) Design

In this section, we describe the details of the two components of our system, demonstrating the ML algorithms developed for anomaly detection and anomaly diagnosis with KPIs. They are the flow charts of the KPI anomaly detection and TopN cell root cause analysis in Fig. 1.

In Figure 2.1 we show the scheme of our system which consists of two main parts. The offline model training sub-process performs subnetwork KPI anomaly detection, health score calculation, and TopN cell root cause analysis. The online detecting sub-process outputs real-time detection results generated from offline modeling, and exact root causes diagnosed by predefined expert rules. User feedback will be used to automatically optimize the parameter settings of our training models.

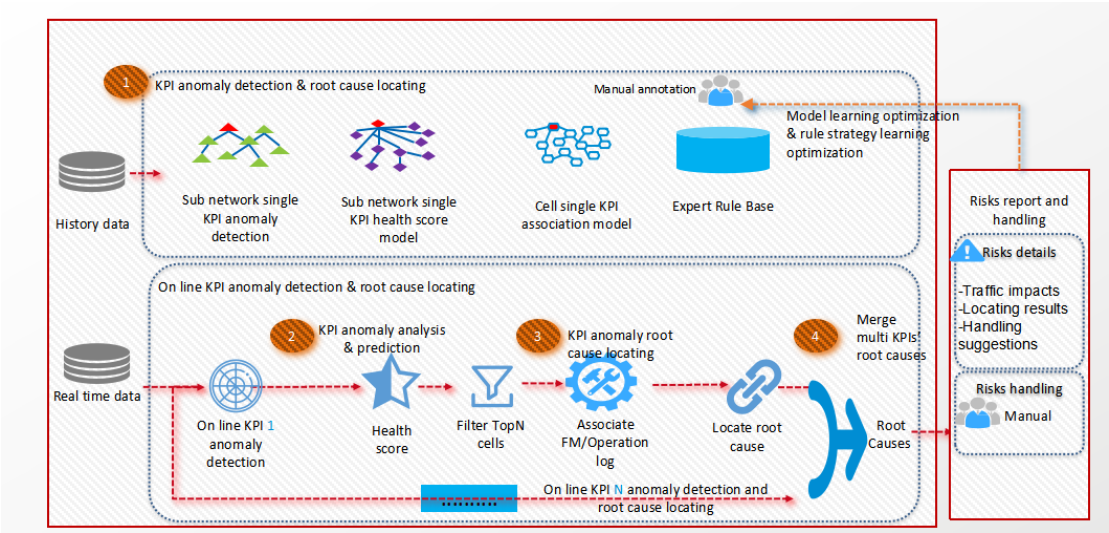


Figure 2.1 KPI anomaly detection &TopN cell Root Cause Analysis

The table 2.1 below is a list of subnet-level anomaly detection KPIs with a detection granularity of 15min.

Table 2.1 subnet-level anomaly detection KPIs.

P310500	RRC Establishment Success Rate
P310501	E-RAB Setup Success Rate
P311502	E-RAB Setup Success Rate in Cell,QCI=1
P310560	Paging Congestion Rate
P311506	E-RAB Setup Success Rate in Cell,QCI=5
P340785	Cell DL PDCP Traffic Average Rate

P340775	Cell UL PDCP Traffic Average Rate
P311475	DL Cell PDCP SDU Volume
P311476	UL Cell PDCP SDU Volume

In Figure 2.2 we show the detection of anomalies in periodic data, while the results of non-periodic data are shown in Figure 2.3 The overall precision and recall of the anomaly detection algorithm are validated to be 95.3% and 96.2%.

The KPIs represent varied characteristics because of the diverse characteristics of network modules. For example, some KPIs show periodicity while others do not; some KPIs have trend, while the other KPIs are stable. A two-stage modeling method is proposed to deal with the huge challenge for comprehensive modeling of all kinds of KPIs. The first stage is the classification stage, where a time series clustering algorithm is formulated to classify the KPIs based on their structure characteristics. In the second stage, the module selects an appropriate time series model for each KPI category, predicting the normal baseline at each time point for a KPI. A value would be denoted as anomaly if it exceeds the baseline of the online detection.

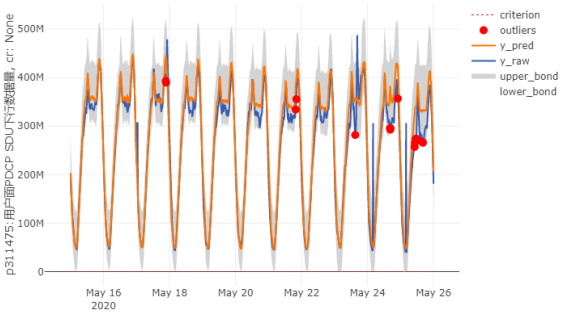


Figure 2.2 anomaly detection in periodic data

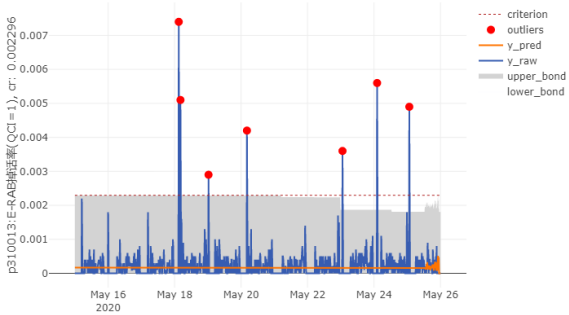


Figure 2.3 anomaly detection in non-periodic data

In Figure 2.4 we show the details of health degree calculation algorithm for filtering false alarms from real anomalies. The health degree value is calculated by the duration and deviation of anomalies revised through self-adaptive criteria. As setting criterion as a critical parameter, the algorithm discriminates between general and abnormal fluctuation with health degree results less affected by absolute values of various KPIs data.

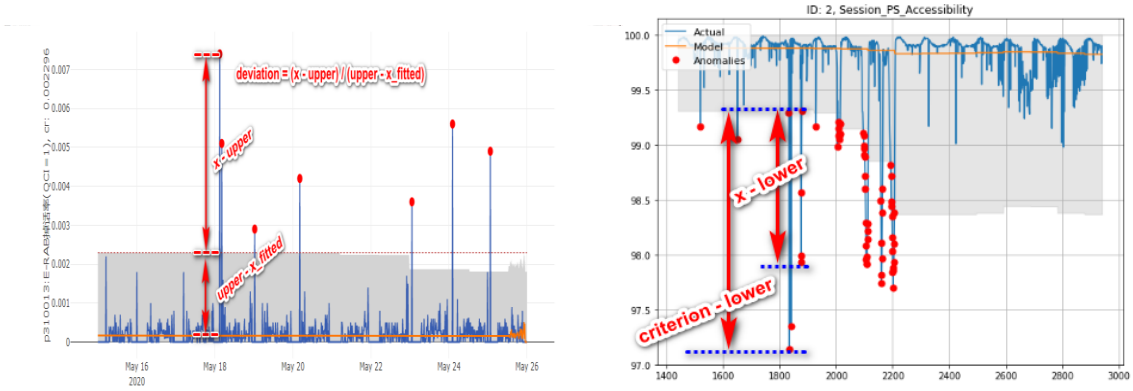


Figure 2.4 subnetwork KPI health score calculation

In Figure 2.5 we show the TopN cell filtering results while an anomaly in subnetwork KPIs data is identified by KPIs anomaly detection algorithm. Through calculating the strength of association between subnetwork data and cells data, the TopN cells analysis algorithm provides top n cells with most significant contribution to subnetwork KPI data fluctuation.

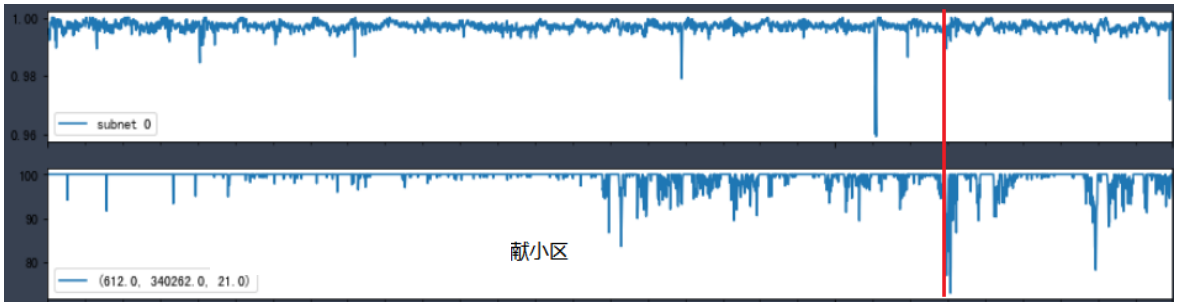


Figure 2.5 TopN cell correlation results of subnetwork anomaly occurrence

As shown in Figure 2.6, when the detected anomaly is a known fault that can be explicitly diagnosed by predefined expert rules, the rule-based diagnosis module could define the root causes according to related information, such as the network topology, the exact mathematical function between the KPI and related counter indicators (counter indicators are more basic performance data, comparing to KPIs), and expert rule library. The rule-based module can generally output exact rule causes and provide direct execution suggestion for recovering.

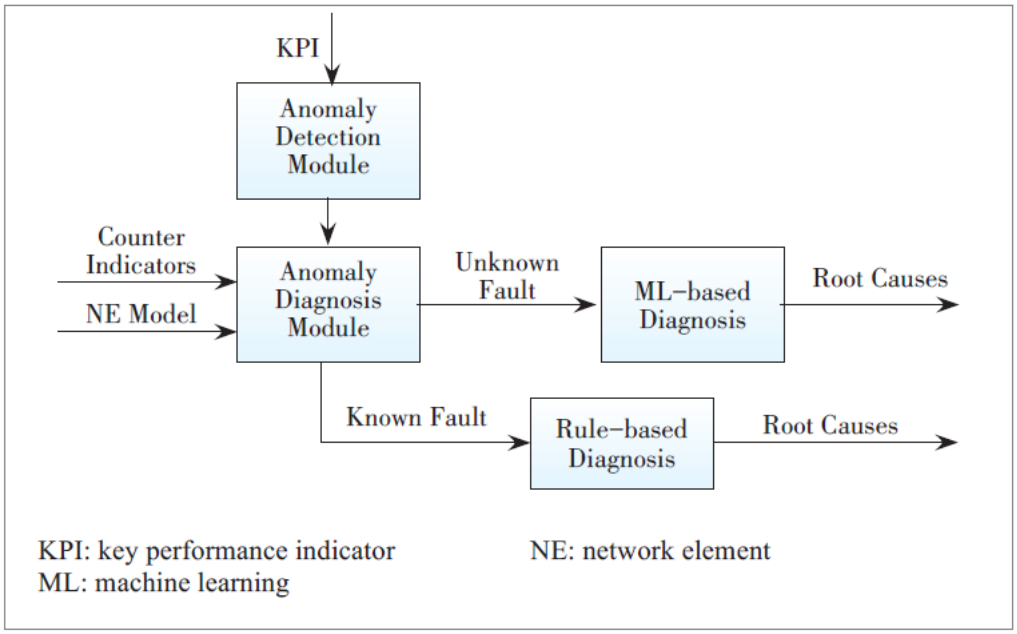
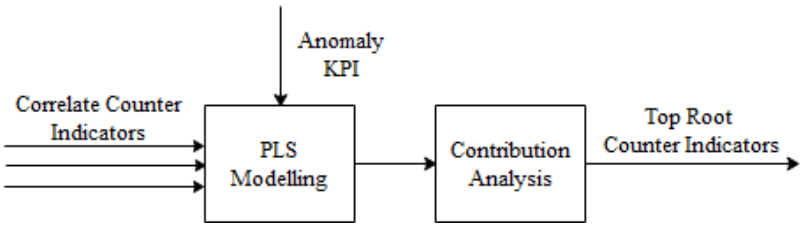


Figure 2.6 A mixed scheme that combines the rule-based and ML-based diagnosis modules.

When the detected anomaly is an unknown fault, the ML-based diagnosis module would define the root causes by using the partial least squares regression (PLS) algorithm as proposed in this paper. The PLS has been used in multivariate monitoring of processing operating performance, which is almost in the same way as PCA-based monitoring . Instead of only finding hyper-planes of maximum variance for independent variables, PLS finds a linear regression model by projecting the response variables and the independent variables to a new space. Compared to standard linear regression, PLS regression is particularly suitable when the dimension of response variables is more than independent variables and when there is multi-collinearity among independent variables. As illustrated in Figure 2.7, when an abnormal KPI is detected, PLS models the KPI as a response variable and the correlated counter indicators as independent variables. Following the PLS modeling, the contribution analysis is conducted to find the top root counter indicators.



KPI: key performance indicator

PLS: partial least squares regression

Figure 2.7 Root cause analysis with PLS model when a KPI is abnormal.

(3) Use case

A typical use case of KPI anomaly detection & TopN cell Root Cause Analysis is shown in Figure 2.8.

Step 1:

Anomalies occurred in subnet 370403 with RRC Drop Rate changing dramatically 18:00 through 18:45 on June 26th. After detecting a sharp increase in RRC Drop Rate from 0.02% on average to 0.2% at 18:00, the tools of KPIs anomaly detection and TopN cell root cause analysis located the problem and identified failure root cause as eNodeB 243747, which followed the same trend as subnet KPI data.

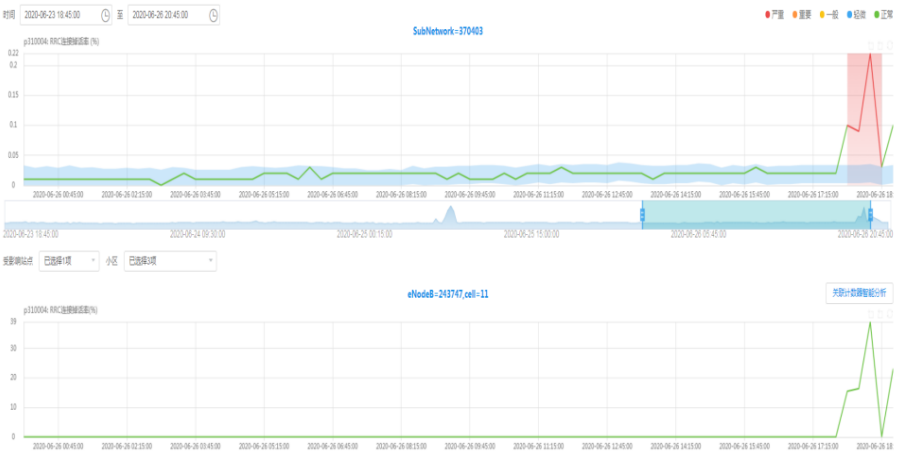


Figure 2.8 A typical use case of KPI anomaly detection

Step 2:

Within this time, top n cells experienced a service failure caused by an RRU abnormality, which associated with link failure alarms detected in an optical receiver. Besides offering a comparison that gauges the degree of performance change in subnetwork, tools of KPI anomaly detection & root cause analysis identify the root cause of failure by associating anomalies with fluctuations in cell counter indicators (see Figure 2.9). The efficiency of fault diagnosis process is significantly enhanced by our system as time consuming decreased from 2 hours with an expert to 10 minutes, which increased by 92%.

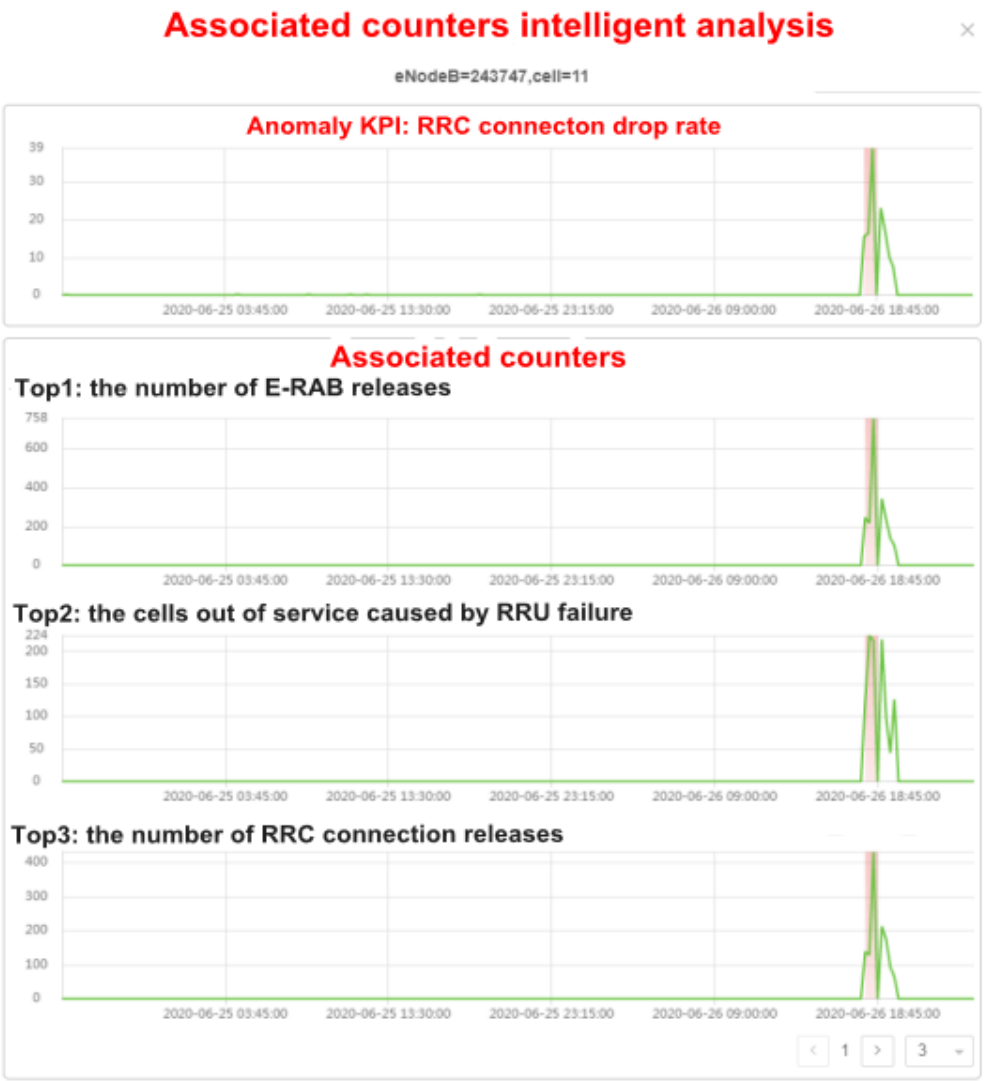


Figure 2.9 associated counters in a typical use case of KPI anomaly detection

(4) Conclusions

The research of intelligent O&M has attracted extensive interest for IT system in recent years, which is known as AIOps . However, this topic is relatively less discussed in wireless networks. As the evolution of wireless networks and the emerging of 5G, the networks become more complicated, emphasizing the disadvantage of manual operation and the desire to automate O&M process with intelligent analysis to handle such a challenge. In this paper, we try to formulate an intelligent operation system based on the layering concept, resulting in a flexible, scaling and manageable framework. And then, one practical use cases, KPI anomaly detection &TopN cell Root Cause Analysis, is studied based on the framework. The real data experiments demonstrate the effectiveness of the proposed method, thus encouraging the further research for intelligent operation with ML technology. In the future, we would develop more use cases to resolve other operation issues in wireless network, for example the automated

log analysis, the prediction analysis, and the optimal parameters configuration.

2.2.2 Use case 2: Geo Data Enhanced Coverage Estimation and Network Planning

With wireless network positioning technology evolution, e.g. Global Navigation Satellite System (GNSS), Finger-printing, TDOA and so on, coverage measurement (RSRP/RSRQ) with location information of UE can be collected by using MDT [5,6]. Base on this, the management and monitor granularity of coverage performance changes from cell level to scenario area or even location bin level. However there are still some problem in the coverage estimation process. From the network management view, different scenario or area should have differentiated coverage requirement. In general, urban area should take tougher criterion than rural area. But actually, some rural areas with high population density and developed economy do exist. Operators need distinguish scenarios' trait more precisely to achieve the target of refined management and planning. And from the trouble shooting view, coverage hole will be false detected by misleading of no man's land or MDT data insufficient. For instance, the scenario of scenic spot in the mountain, users' measurement only appears at the walkway areas. The coverage estimation should focus on the walkway rather than the whole spot area.

To achieve more precise coverage estimation, one approach is adding geography information to network management platform (planning platform, optimization platform, performance monitor platform and etc.). And based on the Satellite Imagery Semantic Segmentation technology, the operator can collect geography information from online satellite image at low cost and time-efficient (high resolution image is not necessary). It can recognize the populated area boundary precisely by analyzing the density of buildings, to assist to set the proper planning KPI requirement of different areas. Moreover it will detect every location bins' landforms (water area, mountainous region, farmland, buildings, roads etc.), than make a Geo-fencing for further scenario-based coverage analysis. These information combined with MDT measurement could be helpful to correct the coverage rate estimation and refined network optimization into scenario and bin level.

1) Dataset Collection and Preprocess

Dataset collection is one difficulty of this work. From internet some open datasets (see Figure 2.10) can be collected .But for the precise of localized use, there still need to collect the raw data and label it manually. Object like Buildings, Misc, Road, Track, Trees, Crops, Waterway, Standing water and Vehicle has been labeled in the form of Polygons and Multi-Polygons (see Figure 2.11), which are simply a list of polygons. The open source format GeoJson and WKT are used for describe geo-spatial shapes.

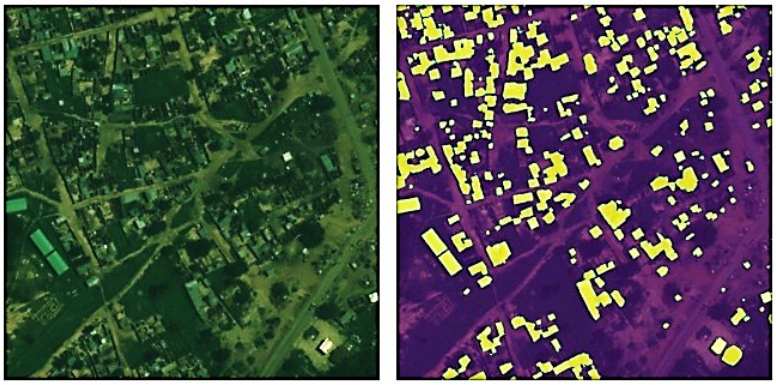


Figure 2.10 Datasets of Building Detection

For the model generalization, the dataset preprocess with rotation, flipping and elastic transform after images be segmented.

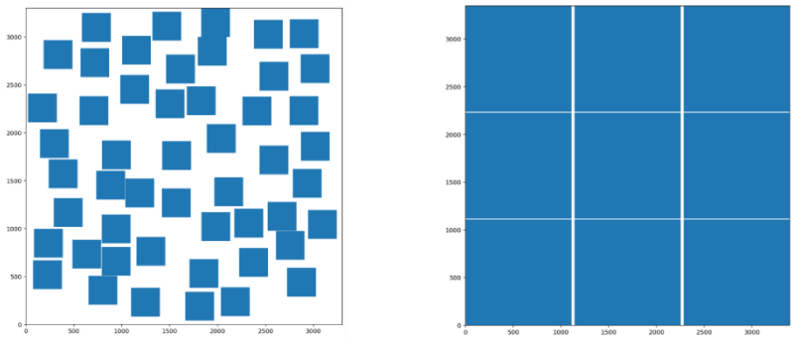


Figure 2.11 Dataset preprocessing

2) Implementation of Imagery Semantic Segmentation model

The implementation adopts U-NET model, one mainstream model of Imagery Semantic Segmentation technology. There are a lot of Open Source Project can be used as a baseline. Figure 2.12 shows the basic structure of U-NET convolutional networks.

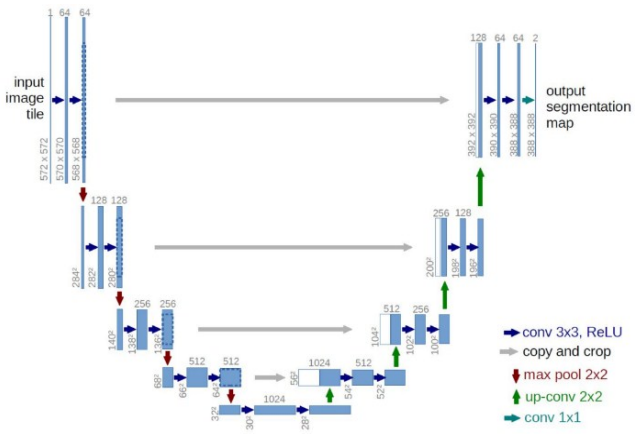


Figure 2.12 The structure of U-NET Convolutional Networks

After training, we use Jaccard index as the model evaluation indication. As it is shown in the Figure 2.13, the score converge to 0.167 on the training-sets and 0.366 on the test-sets. After analysis, we find the performance gap

between training-set and test-set is due to lack of local data. Because most training data are collected from European cities, but the model is applied to Chinese cities.

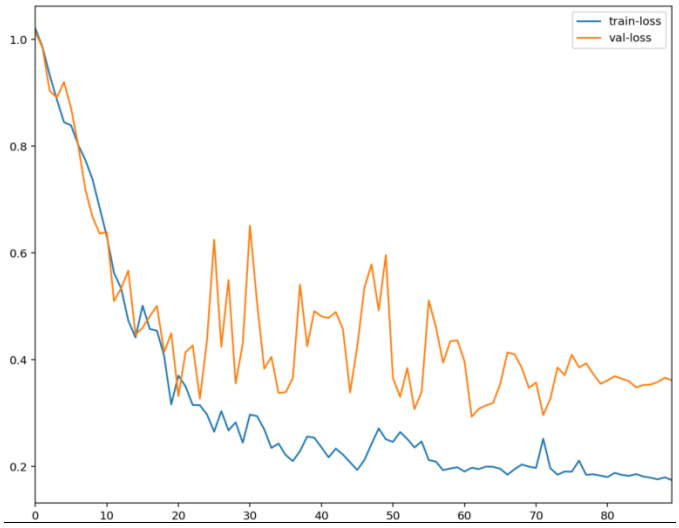


Figure 2.13 Model evaluation

3) Network Management Applications

A typical use of the Geo Data is to adjust the coverage rate. Figure2.14 depicts the coverage of Summer Palace. The major scenic area is waterway, but actually touring line is mostly on land. Moreover, visitors' boats are not allowed to go into the northwest area of lake. Traditionally, we calculate geo coverage rate using two basic method (After Geo binned).

$$\text{coverage rate} = \frac{good\ \ bins}{all\ area\ bins} \quad (1)$$

$$\text{coverage rate} = \frac{good\ \ bins}{all\ \ area\ but\ \ exclude\ no\ data\ bins} \quad (2)$$

It leads a problem that the coverage rate either too bad or too well. These rates can't reflect the customs' experience well. To focus on the touring line area, we wipe out the area of water and re-calculate the coverage rate. It gets much better and shows that the north of the scenic spot is a little bit poor coverage.

	Method (1)	Method (2)	After adjusting
Coverage Rate	68.48%	95.4%	83.01%

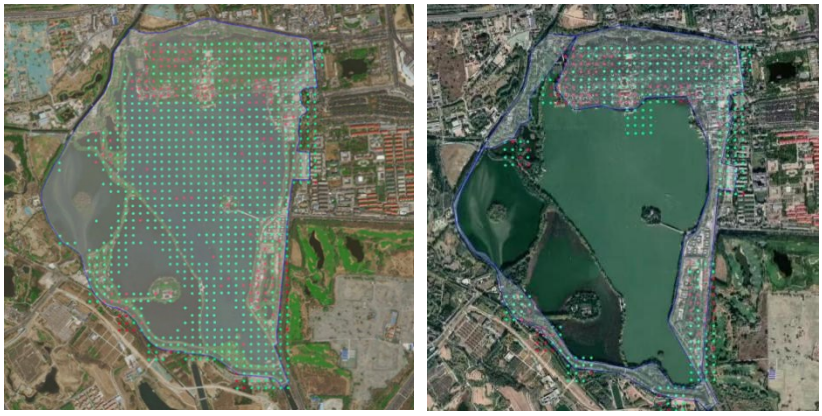


Figure 2.14 Appling the geo data to adjust the coverage rate of scenic spot

Finally, we have integrated the Imagery Semantic Segmentation model into the platform of Scenario-level network performance supervise and analysis (see Figure 2.15). The Geo-data has already been a new dimension to help network management more precisely and individually.

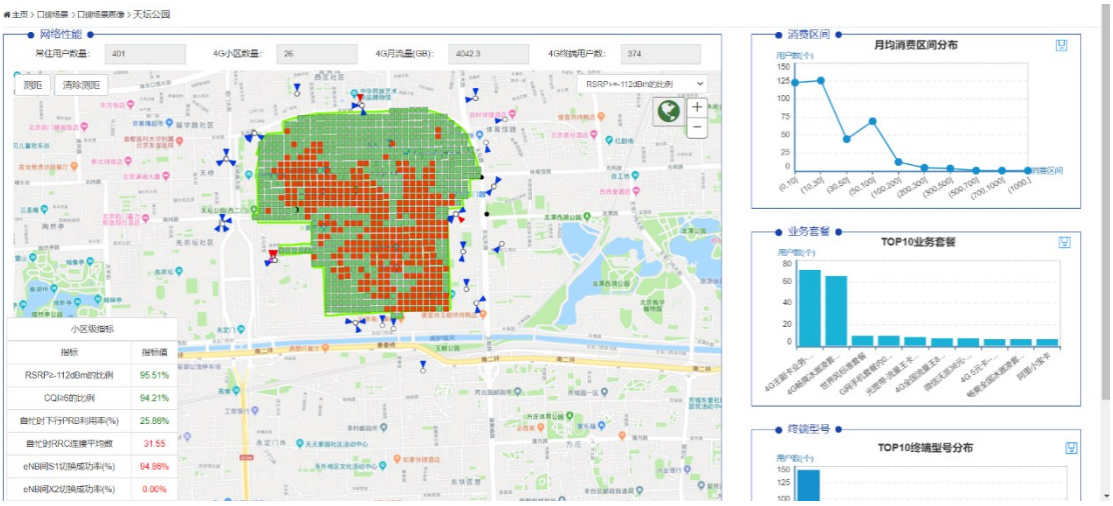


Figure 2.15 Platform of Scenario-level network performance supervise and analysis

2.2.3 Use case 3: Automatic new base station planning via machine learning

Data may be used to assess network performance and reveal existing problems. Data driven wireless network planning has been recognized as tendency among operators. Automatic new base station planning, which analyses and processes multiple dimensions of data collected, including radio environment, geography, available sites and other statistics, derives siting and parameters of planned base stations. It consist of the following methods.

(1) Poor coverage areas estimation

Poor coverage usually account for bad user experience such as frequent dropped calls and low data rate. To identify areas where users suffer from poor service, data of radio environment are collected, including Drive Test Datasets

and Measurement Report Records. Key indicators such as RSRP and RSRQ together with geographic data help to figure out coverage of dispersed spots. Base on poor coverage spots, poor coverage areas are estimated. As shown in Figure 2.16, Machine learning algorithm DBSCAN [7] could cluster poor coverage spots and infer poor coverage areas.

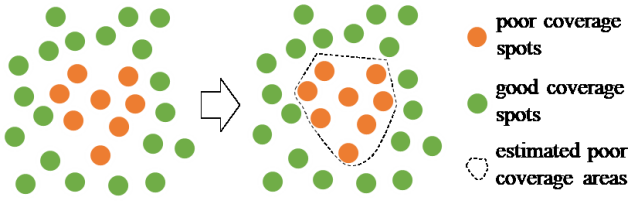


Figure 2.16 Poor coverage areas estimation applying DBSCAN

(2) Newly planned base stations evaluation

The list of available sites is required for site selection. In order to predict coverage when newly planned base stations are deployed, RF simulation is performed. Planned base stations are expected to provide adequate signal strength to estimated poor coverage areas as well as less interference to neighbour cells. To this end, adjusting siting and parameters, for instance antenna direction and down-tilt, transmitting power, until reach satisfying radio condition. Subsequently Planned base stations are listed.

(3) Ranking

A ranked list of planned base stations is generated in this step to decide the priority of deployment. The momentum of deploying new base stations mainly comes from user experience improvement or market needs. Analyses on Call-Detail Records, traffic statistic, complaints and feedback records suggest which poor coverage areas should be paid more attention to. Figure 2.17 presents how Learning to Rank [8] could be applied here to prioritize Planed base stations.

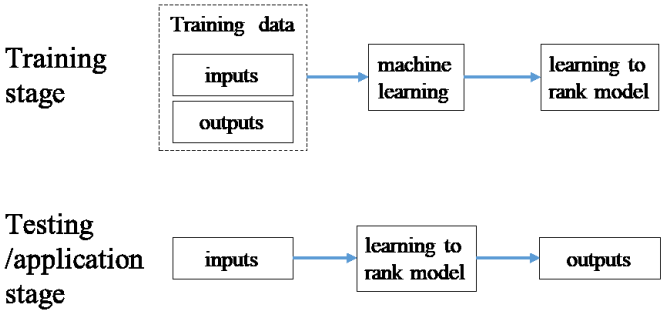


Figure 2.17 Planned base stations prioritization with Learning to Rank

Automatic new base station planning has the potential to offer helpful suggestions to network planning departments. To seek refinements, two following data driven features are expected:

(1) Beam level coverage assessment

The introduction of Massive MIMO in 5G NR makes it possible apply advanced beamforming. The transmitter focus the signal, whose strength in specific direction is optimized. As a result, better coverage and less interference may achieve simultaneously. Coverage will be described more precisely at beam level rather than cell level. Data of radio environment collected at beam level may give more accurate coverage estimation.

(2) Application awareness

5G networks shall support three main use cases: eMBB, URLLC and MMTC. Various applications require different network capabilities. To meet specific requirements, RAN configurations like duplexing, spectrum and frame structure could be optimized. Simple traffic statistic is not capable to tell enough information related to applications. Application aware traffic statistic is in need for advanced network planning.

2.2.4 Use case 4: Coverage Optimization assisted by reinforced learning

In wireless network, continuous coverage is realized by using multi cells. The parameters each cell if not properly configured then it will lead to overlapping coverage and coverage holes among multiple cells. At the same time, because NR is the same frequency network, there will be some interference between overlapping coverage cells, and the coverage hole will lead to weak coverage in some areas, which will lead to the bad user perception. The main parameters that affect these parameter configuration problems mainly related to cell antenna configuration: angle of direction, down tilt, transmission power, and etc. When the traversal method is adopted, N cell networking needs to be calculated about 120000 ^ n times, thus it is difficult to achieve the best combination of above parameters (see Figure 2.18).

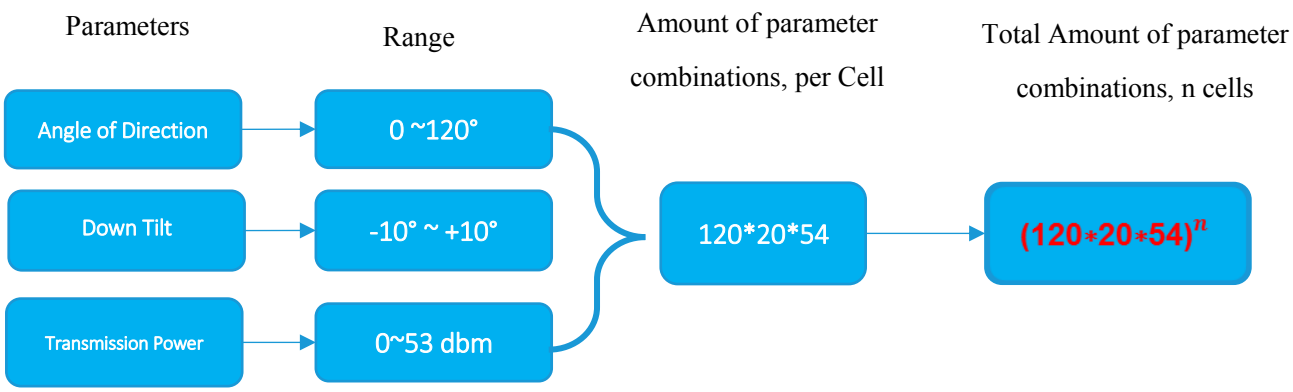


Figure 2.18 AI and simulation combined to reduce planning cost

DQN, DDPG, AAAC and other artificial intelligence algorithms are used to optimize the network coverage, and multiple adjustment parameters and multiple evaluation indexes are used. Since the reinforcement learning needs

reward values, the data collected from real network combined with simulation platform is used to provide such reward values. The procedure is shown in figure 2.19. First step is import the original data, e.g. work parameters, areas need adjustment, then gridding the geographical regions of such areas, constructing the *reward, state* and *action* and then reinforcement learning module would output the *action*. The *action* will be used by simulation platform to work out new *state,* to be used by reinforcement learning module, When iteration converges, the output of the action could be used in actual network.

Considering the global optimization of all cells, when constructing one Q table for all cells, the Q table will be too large and for Q-learning is difficult to meet the requirement. It is proposed to combine the reinforcement learning and deep learning, simulates the value function of Q table through neural network, and then achieves the purpose of calculating the global optimal solution, namely the DQN method.

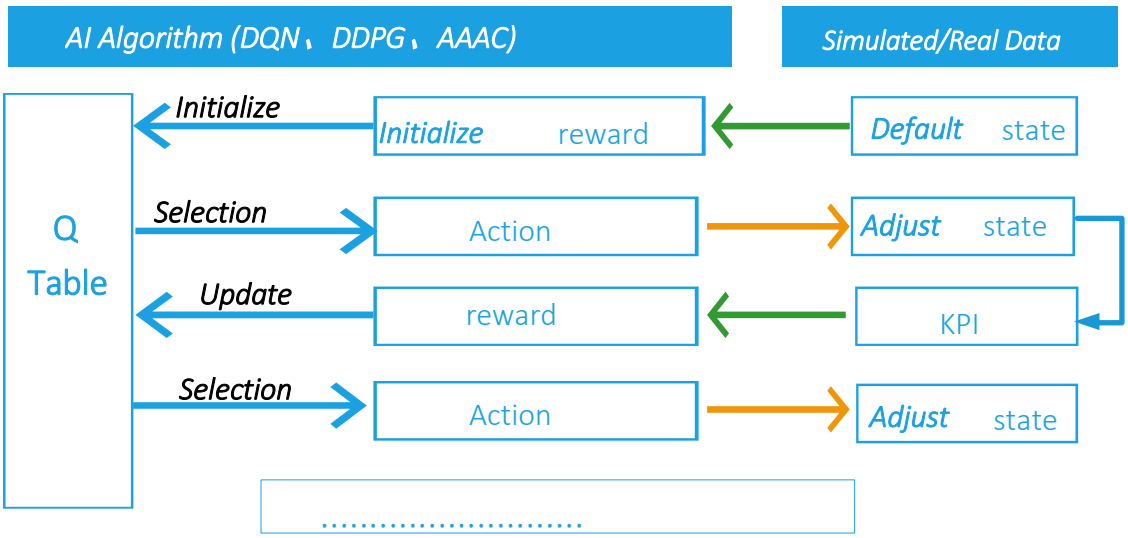


Figure 2.19 procedure of reinforced learning

We used one raid network structure with 3 stations (each has 3 cells) and 9 cells to validate the above method. The initial value of angle of direction, down tilt and transmission power are set to 0 degree, 0 degree and 41 dbm respectively, and average initial reward value is about 0.15. After 30000 iterations, the average reward value is about 0.31(see table 2.2 and Figure 2.20).

Table 2.2 The iteration results for each NodeB

State	Node B	reward	Cell # 1			Cell # 2			Cell # 3		
			AoD	TP	DTA	AoD	TP	DTA	AoD	TP	DTA
Initial	1	0.1432	0	41	0	0	41	0	0	41	0
	2	0.1667	0	41	0	0	41	0	0	41	0
	3	0.1625	0	41	0	0	41	0	0	41	0
Optimized	1	0.3109	294	53	5	246	53	8	150	53	8
	2	0.3099	70	53	8	337	53	6	259	53	4
	3	0.3111	197	53	7	346	53	4	134	53	5

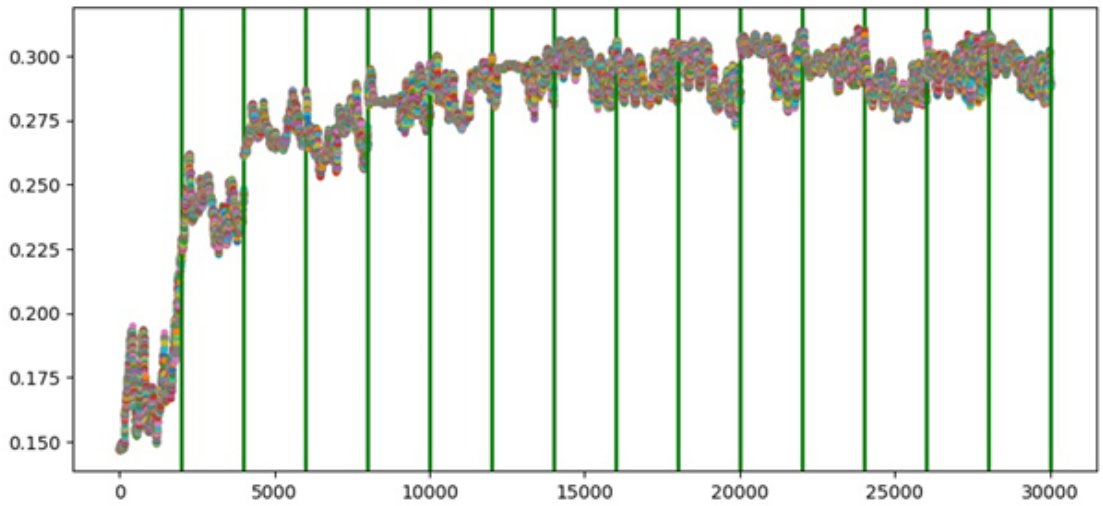


Figure 2.20 the results of reward value with iterations

2.3 MEC Optimization

2.3.1 Use case 5: Intelligent MEC Caching Optimization

(1) Description

Mobile data traffic is anticipated to increase greatly along with a tremendous growth in the number of mobile devices as well as the user interest towards applications with diverse characteristics and demand for network

connection. Supporting such a requirement turns to be a big challenge for developing new network architectures. To this end, Mobile Edge Computing (MEC) equipped with caching functions is introduced as one of the promising technologies.

MEC servers are typically deployed in close proximity to the base stations (BSs) of a wireless network (e.g., a cellular network) to provide delay-sensitive and context-aware applications to the end mobile users. Upon receiving demand from users, MEC server directly delivers the desired content cached locally to the users, instead of downloading the content from data center far away from users. The design of edge-caching network can be classified into two phases, namely, popularity prediction and caching policy. Popularity prediction dominates the design as well as performance of edge-caching network due to the random arrival of content request and limited cache memory.

Popularity prediction has drawn a lot of attention in the context of online media, such as video, photo, news and so on. The increasing of online media has intensified the competition for users' attentions. Only a small part of contents become popular, and the rest are left to languish. It is vital to predict the content popularity and figure out which content is going to be popular ahead of time. Although popularity prediction has been widely studied in recent decades, it is still a challenging problem for following reasons. First, for those methods that leverage content features or user features to build a machine learning model for popularity prediction, the input features are usually sparse and high-dimensional, which make the model overfitted easily. Second, a good prediction relies on high-order combinatorial features, which tends to cost plenty of time to design even by domain experts. Third, feature based methods usually rely on platform dependent features, so it is not flexible to apply them to other platforms and the availability of certain types of these features is often the bottleneck.

In this work [9], we formulate the popularity prediction as a regression problem and present a method of Attention based Long Short Term Memory (LSTM) with Feature Embedding (ALFE). Video influences the MEC caching most due to larger storage cost and longer lifetime and thus is employed as exemplary contents to study the popularity prediction. Video features, user features and popularity history are used as inputs to predict the video popularity at any given time, especially hours after the video is published. The contributions of this work are summarized as follows:

- We present a new popularity prediction method of ALFE and show that it outperforms several competitive baselines. Time series, item features, and user features are utilized to improve the popularity prediction.
- Attention mechanism is used to improve the accuracy of popularity prediction and the interpretability of the proposed method.

- Experiments on a real-world dataset verify the effectiveness of feature embedding and show that the timestamp of popularity sequence is the most informative feature due to the regularity and periodicity of human daily activities.

(2) Solution

1) Formulate the popularity prediction as a regression problem

The popularity dynamic of a video i during time period $[1, T]$ is defined as a sequence $\{x_t^i\}_{t=1}^{T}$, where x_t^i is the accumulated attention received by video i at time t. Features are represented as the feature vectors v^i. The v^i can be time-independent or time-dependent ($\{v_1^i, v_2^i, \dots, v_T^i\}$).

The popularity prediction problem is defined as, given the popularity history $\{x_t^i\}_{t=1}^{t_r}$ and other features v^i, predicting the popularity at any given time t_p. The popularity prediction model aims to learn a nonlinear mapping to $x_{t_p}^i$ given $\{x_t^i\}_{t=1}^{t_r}$ and v^i: $F(x_1^i, x_2^i, \dots, x_{t_r}^i, v^i)$, where $F(\cdot)$ is the nonlinear mapping function and $\hat{x}_{t_p}^i$ is the predicted popularity which is expected to be close to the ground truth $x_{t_p}^i$.

2) Design ALFE to tackle the formulated problem

The structure of the proposed ALFE is shown in Figure 2.21. This solution will be introduced by illustrating how each important component of the structure, namely LSTM, attention mechanism and feature embedding, contributes to the performance enhancement.

LSTM as a variant of Recurrent Neural Network (RNN), has been widely used in a variety of machine learning tasks. LSTM is used in our work to investigate the distribution of features in time-domain, such as **aging effect** capturing the phenomenon that video popularity keeps dropping every day. There are four major components in a typical LSTM unit, including a memory cell with cell state C_t, a forget gate, an input gate and an output gate. The forget gate controls the extent to which the memory cell is going to throw away the cell state of the last time step. As shown in Figure 2.21, LSTM employs the popularity history $\{x_t^i\}_{t=1}^{t_r}$ as input sequence. Once the calculation of this time step is finished, the cell state C_t and hidden state (also output) h_t will be fed into the input recurrently for the next time step calculation until the end of the input sequence is met.

Attention Mechanism in deep learning is inspired by neurosciences. It has been widely used in image classification, Neural Machine Translation (NMT), caption generation and so on. The nature of attention mechanism is focusing on different subsets of the input when dealing with different outputs. In our work, the

attention mechanism is employed to capture the **cycle effect** by elaborated **construction of attention weight** $\{\alpha_t^i\}_{t=1}^{t_r}$. Wherein the **cycle effect** refers to the phenomenon that popularity during the wee hours is relatively small compared with noontime. Legacy attention equations cannot be applied directly in our regression task since the range of input required by the equations differ from that in our ALFE structure. Thus, the final state h_{final} is converted to a popularity value by introducing a linear function $g(\cdot)$.

Feature Embedding comprehensively evaluates video features, user features and popularity history to show how informative each feature is when it is added as a feature vector to the attention mechanism. Specifically, a key feature to be embedded is the timestamp of the input popularity sequence, which has never been considered in previous works. The feature of timestamp contributes to capture cycle effect jointly with construction of attention weight mentioned above.

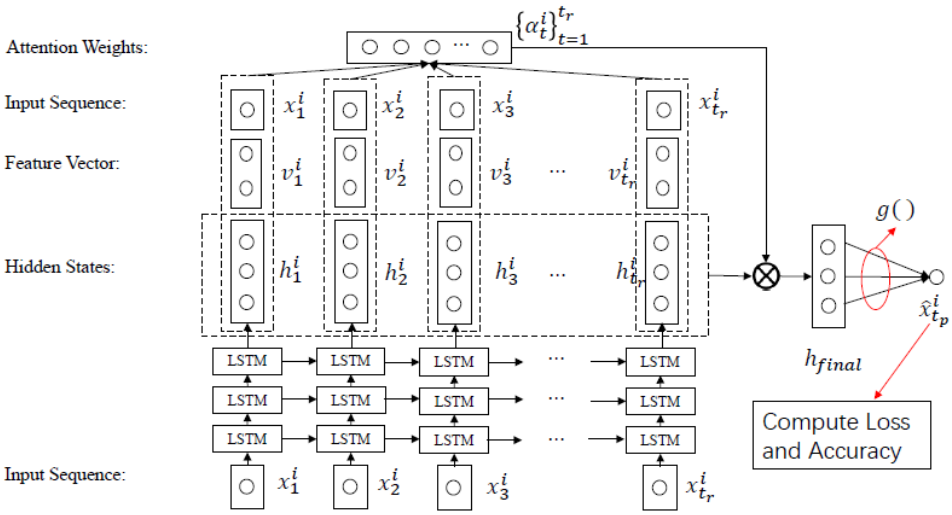


Figure 2.21 The structure of ALFE

(3) Performance evaluations

1) **Dataset**

We crawl the video information and user information through Bilibili APIs [10,11], which require no authentication. The statistics of this dataset are shown in Table. 2.3. Since real-time monitoring of video view count consumes a huge amount of resources, we use comment count instead, which is often used as an indicator of popularity in other literature.

Table 2.3. Statistics of Bilibili Dataset

Entry	Detail
Video count	33548
Video type count	71
Features	upload time, follower count, video duration, like count, average view count of previous videos
Upload time	from 20200101 to 20200301

The popularity distribution of those videos is right skewed with a heavy tail, which means most videos are relatively not successful while only a minority of them are popular, as is shown in Figure 2.22. Because every comment is posted with a timestamp, the time interval of popularity counting can vary from seconds to days even weeks. In our work, the time interval is set as one hour to reflect the regularity and periodicity of human daily activities while reducing the randomness.

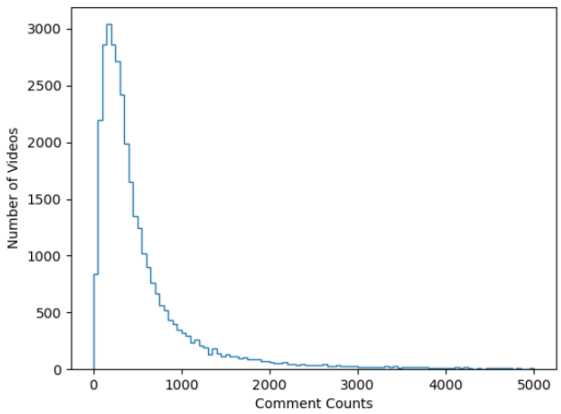


Figure 2.22 The distribution of popularity

2) **Baseline methods**

In order to compare the predictive performance of the proposed ALFE method with other methods, we introduce several methods that have been widely used in popularity prediction tasks or close to our method in a sense. The adopted baselines include **Multivariate Linear Regression (MLR)** model [12], **LSTM** [13] and **Attention-Based Recurrent Neural Network (ABRNN)** [14].

3) Evaluation metrics and loss functions

To measure the effectiveness of the methods mentioned above, we consider two different evaluation metrics and two corresponding loss functions. Mean Squared Error (MSE) is defined as $MSE = \frac{1}{N}\sum_i \left(x_{t_p}^i - \hat{x}_{t_p}^i\right)^2$ and $\mathrm{ACC_{AE}}$ is defined as $ACC_{AE} = \frac{1}{N}\sum_i \mathbb{I}\left[\left|x_{t_p}^i - \hat{x}_{t_p}^i\right| \le \epsilon_1\right]$. Mean Absolute Percentage Error (MAPE) is defined as $MAPE = \frac{1}{N}\sum_i \left|\frac{x_{t_p}^i - \hat{x}_{t_p}^i}{x_{t_p}^i}\right|$ and $\mathrm{ACC_{APE}}$ is defined as $ACC_{APE} = \frac{1}{N}\sum_i \mathbb{I}\left[\left|\frac{x_{t_p}^i - \hat{x}_{t_p}^i}{x_{t_p}^i}\right| \le \epsilon_2\right]$. The $\mathbb{I}[\cdot]$ is the characteristic function

which outputs 1 when the condition holds, and 0 otherwise.

4) Evaluation results

First, we set the input sequence length t_r to 5 and 23, set $\epsilon_1 = 10$, $\epsilon_2 = 0.1$ and $t_p = t_r + 1$. The feature vector used here is the timestamp vector, for example, {21, 22, 23,0,1}, where 21:00 PM is the publication time of the video. The results of different methods are summarized in Table 2.4. The proposed ALFE method has the best performance in terms of accuracy no matter the input length or evaluation metrics. The longer the duration of the training set (the input sequence length), the better the prediction performance.

Table 2.4. Prediction Accuracy

Method	Loss Func:MAPE Evaluation:APE		Loss Func:MSE Evaluation:AE	
	$t_p = 6$	$t_p = 24$	$t_p = 6$	$t_p = 24$
MLR	75.49%	86.96%	81.06%	92.94%
LSTM	76.72%	87.75%	84.21%	93.37%
ABRNN	77.30%	88.34%	85.27%	94.19%
ALFE	80.96%	89.55%	89.49%	95.24%

Then, we change ϵ_1, ϵ_2 and t_p to further prove the superiority of the proposed method while all other parameters are kept the same. Figure 2.23 demonstrates that the prediction accuracy increases with ϵ_1 and ϵ_2. The MLR has the worst performance and the proposed ALFE has the best performance among all the methods no matter the evaluation metrics or thresholds. The reason might be that MLR does not consider the time to be predicted.

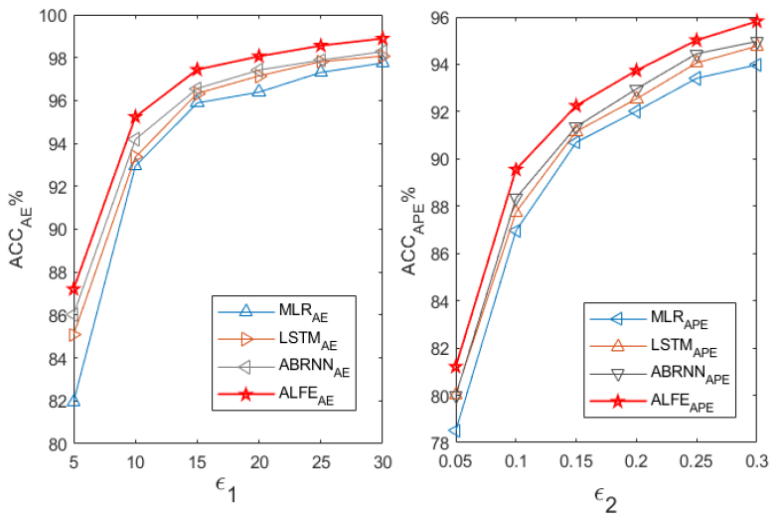


Figure 2.23 The accuracy with varying ϵ

In Figure 2.24, with the increase of t_p, the prediction accuracy decreases while the ALFE method still ranking the top. Different from previous observations, MLR no longer performs the worst. A possible reason is that, as t_p increases, the dependency of x_{t_p} on the previous popularity $x_{t_r+1}, \dots, x_{t_p-1}$ is harder to capture for LSTM. Overall, the performance of our method is relatively stable by varying ϵ and t_p, consistently better than the baselines, which shows the effectiveness of the attention mechanism and feature embedding.

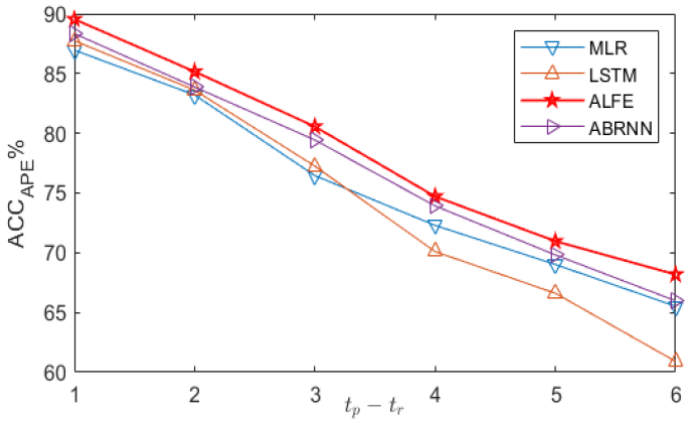


Figure 2.24 The accuracy with varying t_p

However, the choice of feature vector may influence the performance. In order to further figure out which feature contributes the most, we choose every feature one by one as the feature vector and compare them with each other. We set all other parameters the same except feature vector. As shown in Figure 2.25, timestamps of popularity sequence and video duration are the most informative features for our Bilibili dataset. Actually, compared with the least informative feature, which is the type of the video, timestamp feature has 1.1% improvement in terms of prediction accuracy. The surprising thing is that the video type contributes the least to the prediction. The possible reason is, although those videos belong to the same type, the popularity of those videos varies greatly and the type of a video cannot provide useful information.

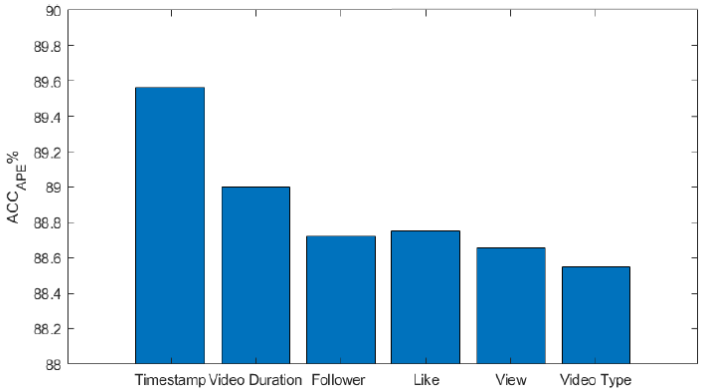


Figure 2.25 The accuracy under different feature embeddings

2.4 Network Resource Optimization

2.4.1 Use case 6: Smart Resource Management for Network Slicing

(1) Description

The emerging fifth-generation (5G) cellular network is envisioned to cater a wide range of services with significantly distinct service requirements like enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low-latency communications (URLLC). Bearing in mind such a goal, the concept of network slicing has recently been proposed by virtually slicing the physical and computational resources of one network infrastructure to meet the diverse needs of a range of 5G users [15]. In order to provide better-performing and cost-efficient services, inter-slice resource management for radio access network slicing (see Figure 2.26) has to track dynamic request patterns of user equipment (UE) and allocate the resources coherently, while guaranteeing an acceptable spectrum efficiency (SE) and satisfying the service level agreements (SLAs). The classical dedicated resource allocation fails to address these problems simultaneously. Instead, it becomes incentive to design an intelligent resource management solution and reinforcement learning (RL) emerges as a promising solution [16].

This year's efforts put much focus on the possible impact of user mobility on the perceived demand. In other words, the mobility of users could exacerbate the fluctuation of service requests and make the on-demand resource management for network slicing more challenging.

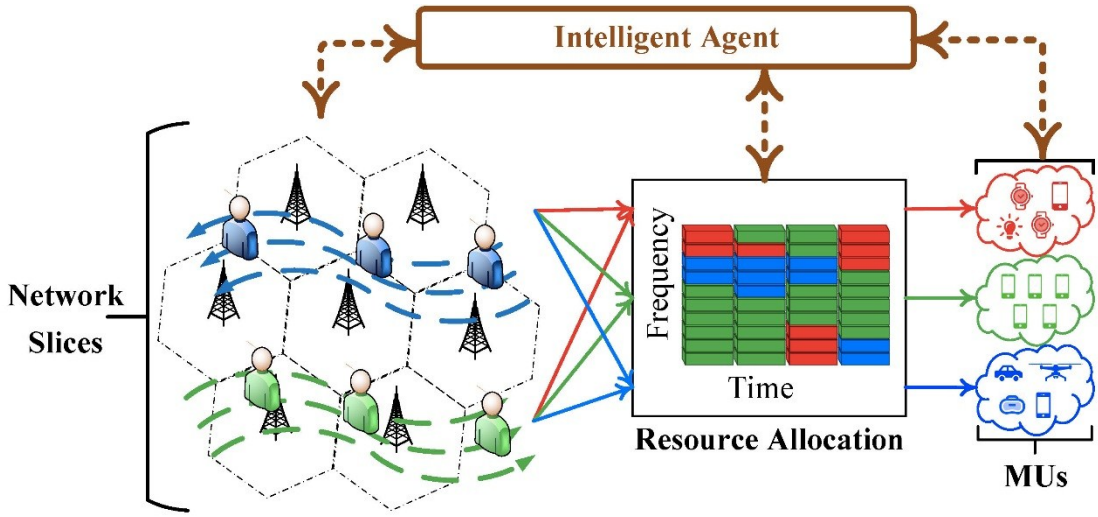


Figure 2.26 An illustration of the access network slicing scenario

Under the framework of hierarchical network slicing, we consider a radio access network (RAN) scenario with multiple base stations (BSs) , where there exists a list of available slices $1, 2, \cdots, M$ sharing the aggregated

bandwidth W and having fluctuating traffic demands $\mathbf{d} = (d_1, d_2, \cdots d_M)$. The objective of our work is to find an optimal bandwidth-sharing solution $\mathbf{w} = (w_1, w_2, \cdots w_M)$ which maximizes the expectation of the whole system utility, which is defined as the weighted sum of SE and QoE satisfaction ratio (i.e., $\alpha \cdot \mathrm{SE} + \beta \cdot \mathrm{QoE}$).

(2) Solutions

Notably, the traffic demands $\mathbf{d}$ at each scheduling period depends not only on the traffic model but also on the dynamic user distribution when users are moving among different BSs. Usually, the user mobility exacerbates the fluctuation of service requests, making the bandwidth allocation problem more complicated and difficult to yield a direct solution. However, we can map the RAN scenario to the context of Markov decision process (MDP) by taking the number of arrived packets in each slice within a specific time window as the state s and the bandwidth allocated to each slice as the action a, as well as deriving the reward r from SE and SSR. Since the traffic demands are unknown apriori, RL is adopted to tackle this inter-slice resource allocation problem and find the optimal policy for resource management in network slicing.

In order to proactively adapt to the dynamic environment and make proper decisions, we could take advantage of the following two means:

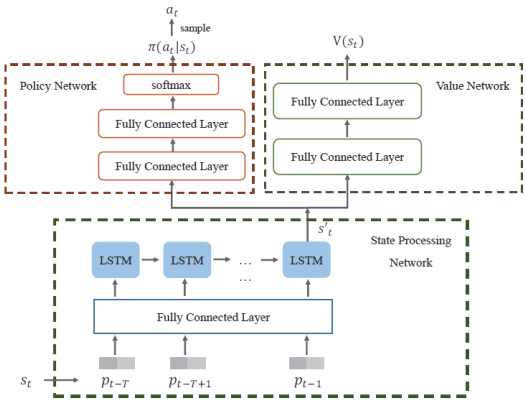


Figure 2.27 The architecture of the LSTM-A2C algorithm

1) We incorporate the LSTM algorithm into the RL algorithm (e.g., advantage actor-critic algorithm) and propose a decision solution, named LSTM-A2C(see Figure 2.27). The LSTM-A2C model takes advantage of the superior series processing capability of LSTM to capture the temporal variation regularity of service requests due to user mobility and further applies the powerful learning and decision-making capability of A2C mechanism to optimize its bandwidth allocation policy based on the comprehensive understanding of the dynamic environment [17].
2) Inspired by the theory of quantile regression, a quantile regression DQN (QR-DQN) and Wasserstein generative adversarial network with gradient penalty (WGAN-GP) to replace the value of $Q(s, a)$ by a

distribution as well as the reputation of generative adversarial network (GAN) to approximate one distribution, a new algorithm based on distributional RL and WGAN-GP, namely GAN-powered deep distributional Q network (GAN-DDQN [18], see Figure 2.28) is considered to realize dynamic and efficient spectrum allocation per slice.

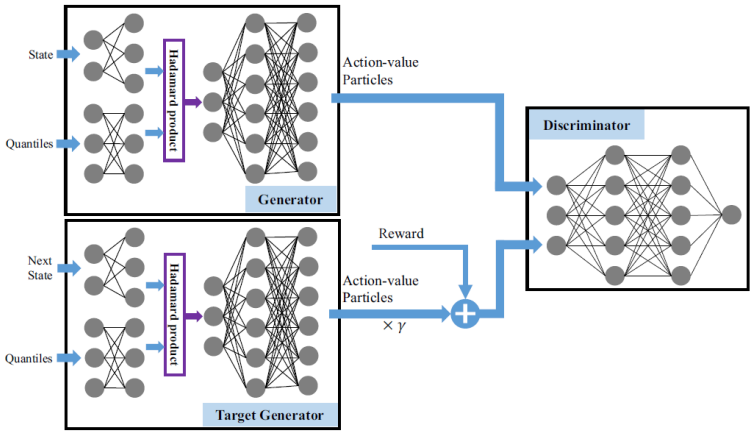


Figure 2.28 The architecture of the GAN-DDQN algorithm

(3) Performance evaluations

Table 2.5 A brief summary of key settings for traffic generation per slice

		VoLTE	eMBB	URLLC
Bandwidth		10 MHz		
Scheduling		Round robin per slot (0.5 ms)		
Slice Band Adjustment		1 second (2000 scheduling slots)		
Channel		Rayleigh fading		
UE No	1200 (*Default*)	200	400	600
	2400	400	800	1200
Speed	Fixed (*Default*)	1m/s	4m/s	8m/s
	Varying	Uniform [Min: 1m/s, Max: 5m/s]	Uniform [Min: 1m/s, Max: 5m/s]	Uniform [Min: 6m/s, Max: 10m/s]
Distribution of Inter-Arrival Time per User		Uniform [Min: 0, Max: 160ms]	Truncated Pareto [Exponential Para: 1.2, Mean: 6ms, Max: 12.5 ms]	Exponential [Mean: 180ms]
Distribution of Packet Size		Constant [40 Byte]	Truncated Pareto [Exponential Para: 1.2, Mean: 100 Byte, Max: 250 Byte]	Constant [0.3 MByte]
SLA	Rate	51 kbps	100 Mbps	10 Mbps
	Latency	10 ms	10 ms	1 ms

We consider a RAN scenario with three types of services (i.e., VoLTE, eMBB, URLLC) and corresponding slices in a simulation area of 240 m × 240 m, where there exist 1200 UEs as a default and multiple BSs. For simplicity, we assume that the UEs within the same slice share the moving pattern (e.g., distribution of velocities and moving direction). When a UE reaches the bound of the simulation area, its direction will bounce. The specific configuration of each UE and its moving speed are described in Table 2.5. In addition, each UE generates service

traffics as summarized in Table 2.5 based on 3GPP TR 36.814 and TS 22.261. The total bandwidth is 10 MHz, and the bandwidth allocation resolution is 200 kHz. We aim to optimize the bandwidth allocation for the central BS with 40 meters' coverage radius in the simulation area.

We evaluate the performance of LSTM-A2C and GAN-DDQN, and compare the results with the classical deep Q-networks (DQN), A2C, and hard slicing. In particular, for the LSTM-A2C, the learning rates of actor-network and critic-network are set to be 0.005 and 0.008 respectively. The observation length of LSTM is set to be 10. And the entropy regularization used for encouraging exploration is set to be 0.001. We first set the speed of UEs fixed and the importance weights in the optimization objective as $\alpha = 0.01$, $\beta = [1,1,1]$. As for the hard slicing solution, each slice is allocated with $\frac{1}{3}$ of the whole bandwidth, since there are three types of services in total.

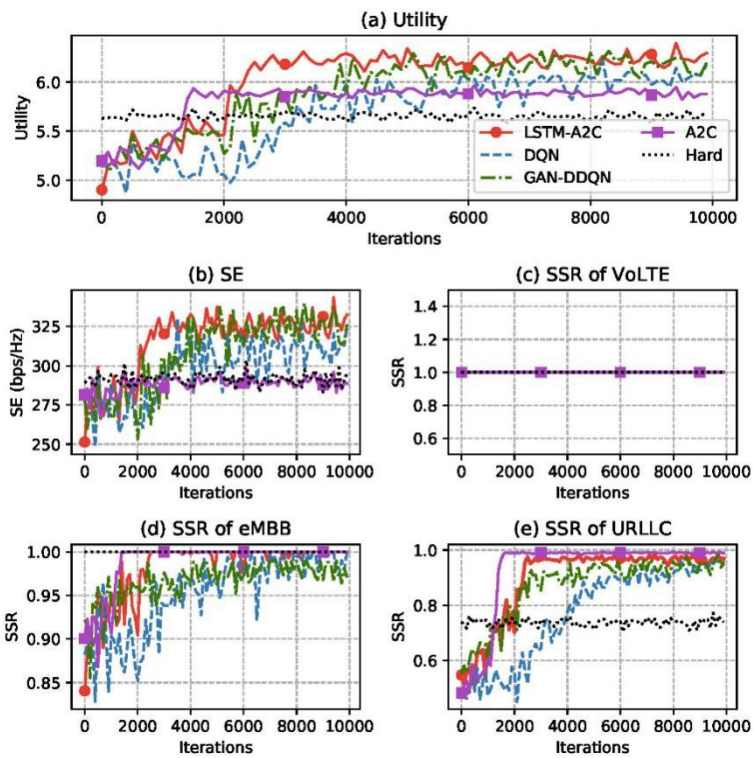


Figure 2.29 The comparison of system utility, SE, and SSRs for different methods.

Figure 2.29 depicts the variations of the system utility with respect to the iteration index. It can be observed that all the RL algorithms have apparent performance improvements through learning and ultimately achieve higher system utility than hard slicing. Among RL algorithms, although A2C shows faster convergence, its performance is not as good as the proposed LSTM-A2C algorithm, which can basically maintain the system utility at above 6.2 after around 3000 iterations. In addition, the LSTM-A2C algorithm also exhibits superior performance than the DQN algorithm terms of both convergence rate and obtained utility. Compared with GAN-DDQN, the state-of-the-art

method, LSTM-A2C gives slightly superior performance and convergence rate. Fig. **(b) - Fig. **(e) further presents the performance comparison on SE and SSR of each service respectively. From the perspective of SSRs, it can be observed that the SLA of all three slices have reached our expectations through learning. For VoLTE slice, it's easier to achieve high SSR due to its low requirements of throughput and latency. For the other slices, despite their high requirements of throughput or latency, both LSTM-A2C and A2C can achieve almost 100% SSR after about 3000 iterations. Compared with LSTM-A2C, the performance of DQN is slightly worse, while A2C shows slightly faster convergence and higher SSR. From the perspective of SE, the proposed LSTM-A2C achieves the highest SE among the three methods, indicating that LSTM-A2C can capture the temporal variation regularity of service requests and adjust the bandwidth allocation flexibly so as to improve SE on the basis of guaranteeing SLAs of different services. However, A2C shows trivial improvement in SE because it tends to allocate bandwidth conservatively to get stable yet inferior SSRs than LSTM-A2C. Compared with GAN-DDQN, LSTM-A2C converges to the same final level as GAN-DDQN but exhibits a more stable convergence curve.

Figure 2.30 further compares the LSTM-A2C and GAN-DDQN under extensive settings, where the speed of UEs varies according to Table 2.5 It can be observed that contrary to our previous findings, LSTM-A2C converges slightly slower than GAN-DDQN but is capable of leading to encouragingly comparative results, if other settings remain unchanged. Such an observation is consistent with the claim that GAN-DDQN performs well under different varying scenarios. If we change $\boldsymbol{\beta}$ as [1,1,5], LSTM-A2C gives slightly superior performance and such a performance gain becomes larger along with the increase in the number of UEs. This implies that instead of capturing the variations by approximating the complicated distributions like GAN-DDQN, embedding the variation prediction into RL sounds feasible as well.

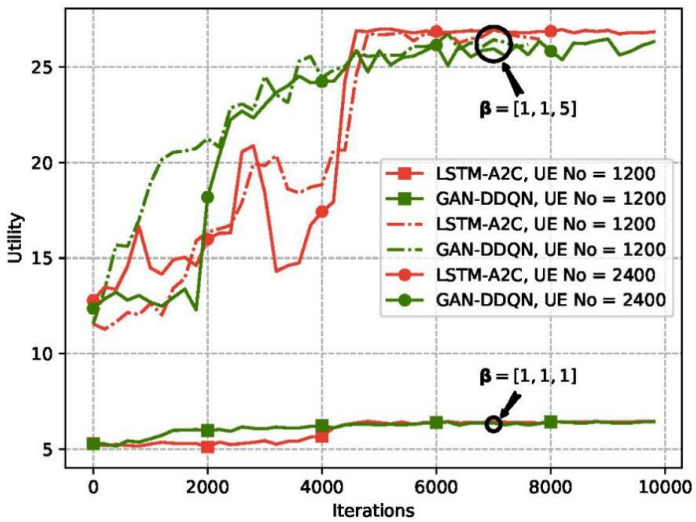


Figure 2.30 Extensive comparison between LSTM-A2C and GAN-DDQN.

2.5 Radio Transmission Technologies

2.5.1 Use case 7: Deep Learning Aided Channel Estimation

(1) Description

Channel estimation is often regarded as a critical component for modern wireless systems. By inserting pre-known pilot sequences along with the transmit data, receivers shall be able to estimate the real-time wireless environment accordingly and perform coherent detection thereafter. Since the pilots consume air interface resources without useful information delivery, the existing literature spends tremendous efforts in improving channel estimation accuracy, especially for multiple input multiple output (MIMO) orthogonal frequency division multiplexing (OFDM) configurations, which is commonly adopted in current cellular or wireless local area networks (WLAN) systems. Nevertheless, the existing schemes mostly rely on abstracted channel models with certain characteristics, such as channel sparsity, and the application to practical systems is still challenging especially when the hardware imperfectness has been considered.

In addition to the model-based channel estimation schemes, the model-free channel estimation approaches, together with recent development of deep learning technologies, have been considered for practical wireless communication systems. For example, deep neural networks (DNN) and convolutional NN (CNN) are directly introduced into the channel estimation in the conventional OFDM systems. The above methods provide insightful results on facilitating the deep learning technology for wireless application. However, the existing literature for channel estimation focuses on the channel state recovery, which obtains channel state information (CSI) knowledge from the received pilot signals, while the interpolation schemes to calculate CSI values at wireless data transmission areas receive limited research attention. This is partially because the non-linear interpolation relation is challenging to characterize and the real-time requirement of channel estimation prevents complicated interpolation algorithms. To this end, we exploit the super-resolution (SR), a deep-learning based image reconstruction technique, to model the non-linear interpolation relations in the conventional channel estimation problem and recover the CSI from the limited observations of pilot signals [19]. Furthermore, in the commercial long term evolution (LTE) networks, the channel estimation needs to be performed on a sub-frame (10 milliseconds duration) basis and the corresponding delay budget is less than one millisecond in general. Therefore, how to strike a balance between the processing delay and the channel estimation performance using the SR learning framework, is also be carefully investigated and addressed in our work [20].

(2) Solutions

a) SR based Channel Interpolation

Consider a typical MIMO-OFDM system, with N_t transmit and N_r receive antennas in the wireless fading environment. The received symbols at the ith subcarrier, after the Fast Fourier Transform (FFT) processing, can be modeled through,

$$\mathrm{y}_i(t) = \mathrm{H}_i(t)\mathrm{x}_i(t) + \mathrm{n}_i(t) \tag{1}$$

where $\mathrm{H}_i(t) \in \mathbb{C}^{N_r \times N_t}$ denotes the MIMO fading coefficients, $\mathrm{x}_i(t)$ denotes the transmitted symbols, and $\mathrm{n}_i(t)$ denotes the additive white Gaussian noise (AWGN) with zero mean and unit variance. In the practical system, to minimize the channel estimation overhead, the entire process is performed on a resource block (RB) basis (usually in accordance with the coherence time and coherence bandwidth) with N_s time slots and N_{sc} subcarriers. In addition, to control the resource usage for channel estimation, only limited locations are selected for pilot allocation. Denote Ω_p to be the collections of pilot positions and the aggregated CSIs for pilots in j^{th} RB can be expressed through, $\mathrm{H}_p^j = \{\mathrm{H}_{i_p}^j(t_p), \forall(i_p, t_p) \in \Omega_p\}$, where i_p and t_p denote the indexes of subcarrier and time slot within RB.

In the conventional channel estimation process, receivers perform channel state recovery to obtain an estimated channel conditions at the pilot positions, and then apply interpolation mechanisms to get the entire channel coefficients. For all the possible $(i_p, t_p) \in \Omega_p$, the classical least square (LS) and minimum mean square error (MMSE) channel state recovery can be described through,

$$\widehat{\mathrm{H}}_{i_p}^{j,LS}(t_p) = \mathrm{y}_{i_p}^j(t_p)\mathrm{x}_{i_p}^H(t_p)\left[\mathrm{x}_{i_p}(t_p)\mathrm{x}_{i_p}^H(t_p)\right]^{-1}, \tag{2}$$

$$\widehat{\mathrm{H}}_{i_p}^{j,MMSE}(t_p) = \mathrm{R}_\mathrm{H}\left\{\mathrm{R}_\mathrm{H} + \left[\mathrm{x}_{i_p}(t_p)\mathrm{x}_{i_p}^H(t_p)\right]^{-1}\right\}^{-1}\widehat{\mathrm{H}}_{i_p}^{j,LS}(t_p), \tag{3}$$

where $\mathrm{R}_\mathrm{H} = E[\mathrm{H}_{i_p}(t)\mathrm{H}_{i_p}^H(t)]$ is the channel correlation matrix.

With the estimated channel states at pilot positions, the deep learning based SR image reconstruction technique is applied to accomplish the channel interpolation. Based on the existing literature, two types of neural networks (see Figure 2.31) show superior restoration performance over the traditional SR technology, which are SR convolutional neural networks (SR-CNN) and enhanced deep residual networks for SR (EDSR), and the architecture of these two networks are shown in Fig. 1.xx.

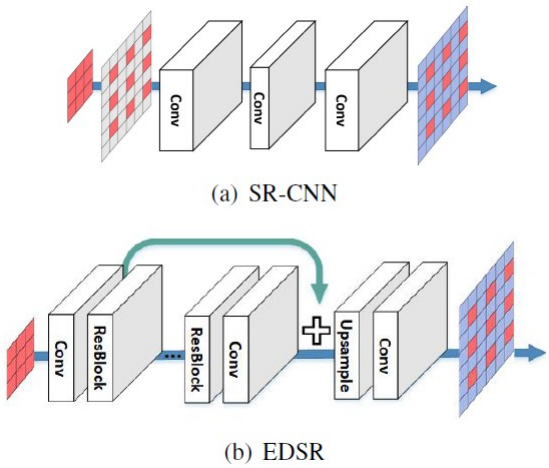


Figure 2.31 Architectures of two typical SR neural networks

For SR-CNN, in the first stage, the low-resolution (LR) image $\widehat{\mathrm{H}}_p^j = \left\{\widehat{\mathrm{H}}_{i_p}^j(t_p)\right\}$ is expanded to the high-resolution (HR) image $\widehat{\mathrm{H}}_{HR}^j = \left\{\widehat{\mathrm{H}}_{i,HR}^j(t)\right\}$ which can be obtain through linear or Gaussian interpolation (GI), and then using neural networks to approximate the non-linear relation between $\widehat{\mathrm{H}}_{HR}^j$ and $\widehat{\mathrm{H}}^j$. While for EDSR, a residual learning network is utilized to progressively predict the SR images $\widehat{\mathrm{H}}^j$ from LR images $\widehat{\mathrm{H}}_p^j$. Since SR-CNN focuses on the non-linear relationship between linear or Gaussian interpolated channel conditions and the real channel measurements, the optimization space for SR-CNN is in general limited, while EDSR applies massive residual network blocks to gradually generate the fine-grained result. As a result, EDSR can provide better performance than SR-CNN approach under the same condition.

Different from the traditional training data sets generation in the previous SR research, where a HR image is given in advance and the associated LR image is obtained through down-sampling, the true channel conditions H^j are difficult to collect in general. To overcome this obstacle, we generate H^j through a statistical channel model, e.g. COST 2100, and simulate the pilot transmission process as defined in (1) via numerical examples. Based on the classical LS and MMSE estimation methods, e.g. (2) and (3), we generate LR images. Through this approach, we can generate sufficient large size of data sets for training and evaluation.

b) Low Complexity Implementing

Inspired by the EDSR scheme, we further divide the entire process into two steps, where the first step focuses on the feature extraction and fusion processes to obtain several N_{fc}-dimension features, and the second step scales up the N_{fc}-dimension features into $N_{sc} \times N_s$ resource elements by up-sampling and maps the high dimensional features back into the complex valued channel responses. The time domain correlations are much stronger than the frequency domain, which can be regarded as the domain knowledge for SR based wireless channel estimation tasks.

To exploit this effect, we propose the Long-Short term memory based Residual Network (LSRN, see Figure 2.32) framework as shown in Fig. 1.xx, where a recurrent learning block is applied to extract the time domain correlations.

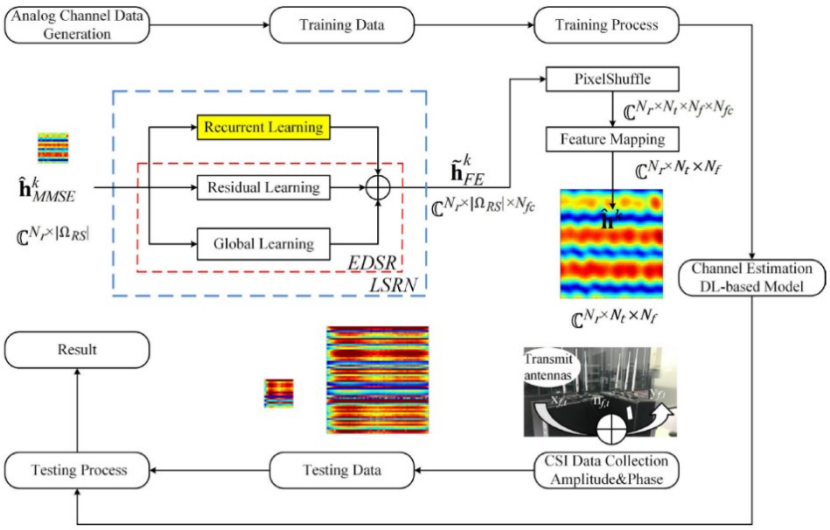
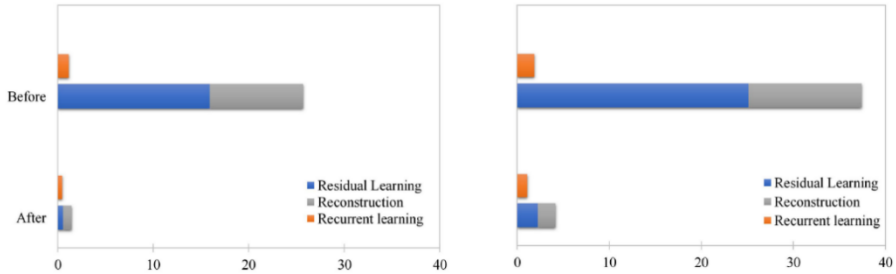


Figure 2.32 Overview of LSRN framework

Although we designed a recurrent learning based LSRN scheme to recover the channel states, as the channel estimation is typically a delay critical task, the process delay of the proposed scheme needs to be carefully studied as well. Therefore, we further focus on proposing a low complexity implementing strategy, denoted as LSRN-L, to satisfy the potential delay requirement. To solve this issue, a straight forward approach is to simplify the number of convolutional layers in each residual blocks (ResBs) of the original residual network (ResNet) architecture, due to the fact that more than 60% of the processing time is occupied by the residual learning process as shown in Figure 2.33.



(a) Pilot configuration (14, 14) (b) Pilot configuration (28, 28)

Figure 2.33 Processing delay of LSRN before and after network simplification

Another scheme to reduce the complexity of the local residual learning path is to reduce the number of ResBs and convolutional kernels. In the conventional EDSR scheme, to reduce the number of ResBs and kernels may greatly affect the image SR performance. However, in the specific task, to reduce the number of ResBs and convolutional

kernels, the corresponding performance degradation is controllable due to the following reasons. First, since the correlations between different REs is much stronger than the pixels in the convolutional image SR task, the differences of extracted feature maps among different ResBs are much smaller, which makes the possibility to reduce the number of ResBs with marginal performance. Second, with the proposed LSRN architecture, part of the feature losses due to the smaller number of ResBs and convolutional kernels can be compensated by the recurrent learning branch. As shown in Figure 2.34, the overall processing delay for 14×14 pilot configuration can be reduced average from 20 ms to about 0.9 ms.

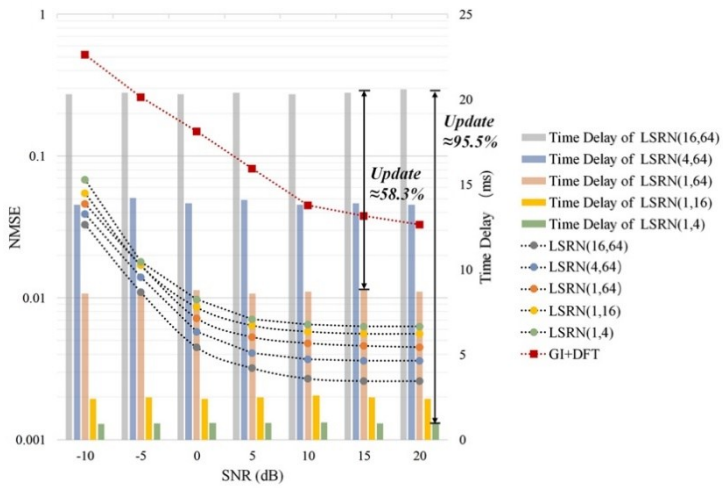


Figure 2.34 NMSE and processing delay comparison of different network structures

(3) Performance evaluations

In this section, we deploy the SR based channel estimation scheme in the WiFi prototype systems using the trained SR neural network and provide some empirical results to show the effectiveness of the proposed channel estimation scheme by comparing with other traditional methods. As shown in Figure 2.35, two commercial desktops are communicating with each other through IEEE 802.11n WiFi protocols.

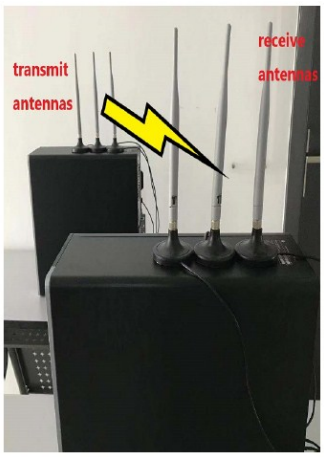
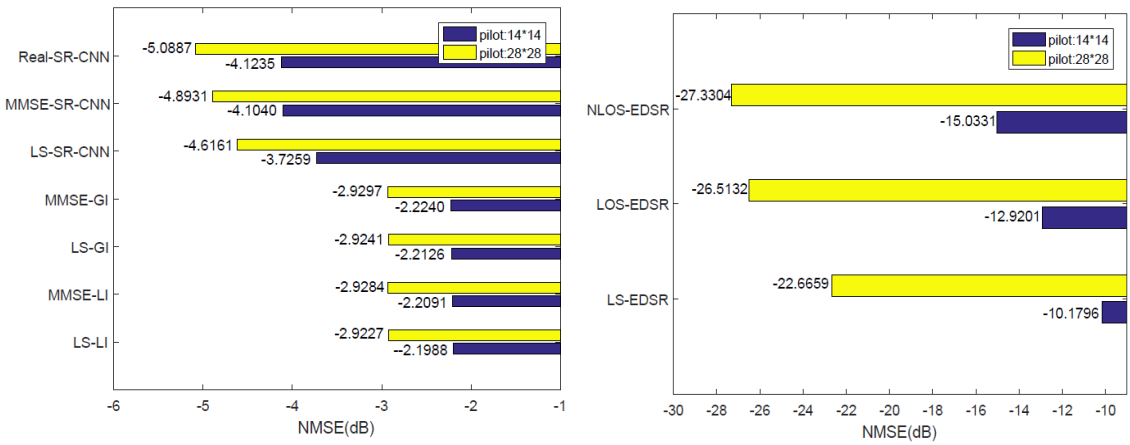


Figure 2.35 Real deployment of the WiFi prototype system

Since some of the subcarriers are muted as guard subcarriers in WiFi prototype systems, we slightly modify the input and output matrices, where $N_{sc} = N_s = 56$ is selected and the number of pilots is scaled accordingly.

In the first experiment, we compare the SR-CNN based channel estimation method with conventional schemes to show the effectiveness of the proposed SR based framework. LS and MMSE algorithms are adopted for channel state recovery at the pilot locations and we test two different pilot arrangements with 14×14 and 28×28 configurations. As is shown in Figure 2.36, the SR-CNN based interpolation scheme outperforms the traditional LI or GI approach under both LS and MMSE channel state recovery methods, where the normalized MSE (NMSE) performance reduces from more than -3 dB to -4 dB. In addition, by comparing Real-SR-CNN and MMSE-SRCNN curves, we can show that the SR networks trained from COST 2100 model is already sufficient for channel estimation in the practical system, and the NMSE loss is less than 0.2 dB for 28×28 case and 0.02 dB for 14×14 cases.

In the second experiment, we redo the above experiments by replacing the SR neural networks, e.g. from SR-CNN to EDSR. As EDSR provides better PSNR performance as shown in previous section, the better NMSE performance can be expected. As shown in Figure 2.36, for both LOS and NLOS scenarios, EDSR based schemes achieve less than -10 dB NMSE for 14×14 configuration and less than -20 dB NMSE for 28×28 configuration. This is partially because the channel conditions are more close to sparse images and the gradually learning approach can adopt to the tiny variations.



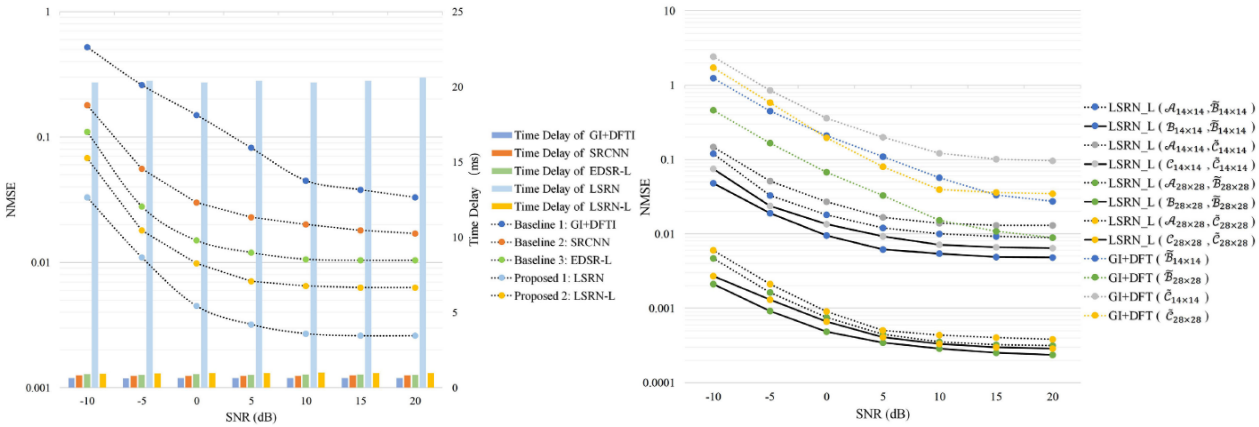
(a) SR-CNN based (b) EDSR based

Figure 2.36 NMSE performance of SR based channel estimations

In the third experiment, we compare the channel estimation performance in terms of NMSE as well as the processing delay versus SNR results under different pilot configuration, where we rely on COST 2100 model to generate the training and testing data sets. The corresponding numerical results are depicted in Figure 2.37. As we

can see from this figure, in the condition of 14×14 pilot configuration, the proposed LSRN and LSRN-L schemes outperforms the baseline schemes in terms of NMSE among 14 dB to 15 dB and 10 dB to 11 dB, respectively. Meanwhile, in terms of the processing delay, LSRN-L performs much better than LSRN scheme and eventually achieves less than 1 ms delay budget.

In the fourth experiment, we compare the channel estimation performance between the tradition GI plus discrete fourier transform interpolation (DFTI) scheme and our low complexity LSRN-L scheme. As the corresponding numerical results shown in Figure 2.37, we can see that our LSRN-L scheme outperforms the baseline scheme GI plus DFTI in terms of NMSE in all circumstances. By comparing the test results of different data set training models, we can conclude that the network trained by model generated data can also be applicable to the channel estimation scenarios under the WiFi environment LOS and NLOS, which represents our network is robust to the actual application scenarios. Last but not least, this experiment shows that the complex correlation characteristics between pilots can be learned through network training, so that the network trained by model generate data training can adapt to different channel fading environments in actual communication systems.



(a) Comparisons among different schemes (b) Comparisons between LSRN-Land GI-DFTI

Figure 2.37 NMSE comparison between traditional and LSRN based channel estimation schemes

2.5.2 Use case 8: Deep learning aided MIMO Detection

(1) **Description**

Multiple-input multiple-output (MIMO) systems have been selected as one of the key features in the current LTE systems and the massive deployment of MIMO systems are often regarded as a breakthrough for 5G systems. Although the transmit side processing for MIMO transmission is often straight forward, the information detection problem at the receiver side is in general difficult. Traditional MIMO detection methods include linear and

non-linear approaches. For example, linear MIMO detector includes zero forcing (ZF) or minimum mean square error (MMSE) equalizers, while non-linear MIMO detector often rely on minimum distance based maximum likelihood (ML) detection. Although the ML-type decoding provides optimal detection performance in theory, the associated decoding complexity is in general unaffordable with current technology[21].

Recently, with the development of machine learning, more complicated problems in wireless communications can be formulated and efficiently solved by this framework. Although the existing machine learning based approaches show promising gain over the traditional signal detection methods, the following issues in the conventional MIMO detectors have not been addressed based on our current investigation. One of the key issues in the MIMO detection design lies in the imperfectness of channel knowledge due to the practical channel estimation method, the time-varying nature of communication devices and limited number of reference signals. Another issue that has not been solved is the efficient deep learning network architecture for the MIMO detection. To this end, we exploit the generalization capability of neural networks to address the robust MIMO detection problem with imperfect channel knowledge and we use a neural network to directly get a mapping function of received signals, channel matrix and transmitted bit streams.

(2) Proposed methods

Consider a downlink MIMO transmission as shown in Figure 2.38, with N_t transmit and N_r receive antennas. Given the time slot t and the modulation size 2^M, a binary information stream $\mathrm{b}(t)$ are modulated through function $f(\cdot, M)$ and the modulated symbols are given by $\mathrm{x}(t) = f(\mathrm{b}(t), M)$. The mathematical model for the received symbols $\mathrm{y}(t)$ are given by, $\mathrm{y}(t) = \mathrm{H}\mathrm{x}(t) + \mathrm{n}(t)$, where H denotes the flat Rayleigh fading channel coefficients and $\mathrm{n}(t)$ is the additive white Gaussian noise with zero mean and unit variance.

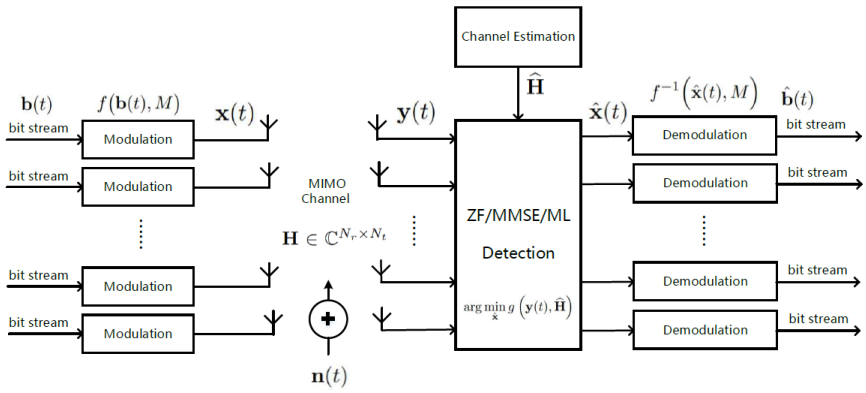


Figure 2.38 Overview of MIMO system

In the practical systems, due to the limited power and resources for pilot symbols, the estimated channel state information (CSI) at the receiver side can not be perfect in general. For illustration purpose, we denote $\hat{\mathrm{H}}$ to be the

imperfect CSI which can be obtained through ML based channel estimation. Therefore, the detected bits for time slot t is given by,

$$\hat{\mathrm{b}}(t) = h\left(\mathrm{y}(t), \hat{\mathrm{H}}\right), \tag{1}$$

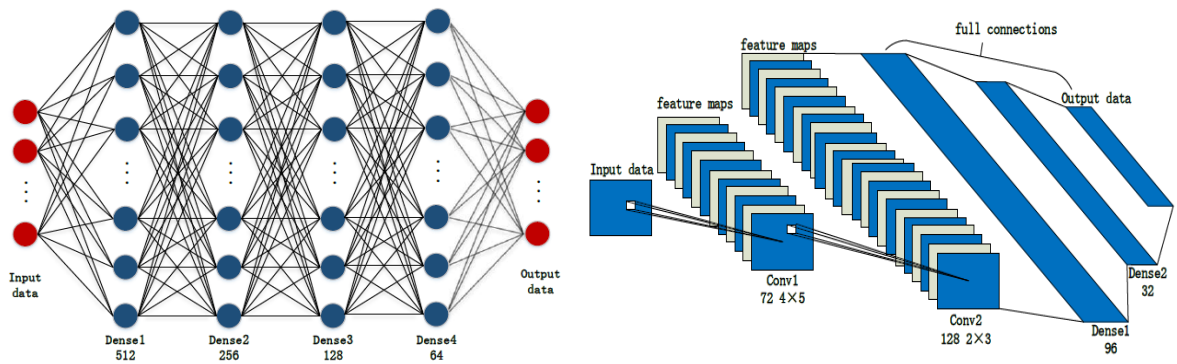
where $h(\cdot)$ denotes a combined procedure of symbol detection and demodulation, and $\hat{\mathrm{H}} = \{\hat{\mathrm{H}}_n\}$ with subscript n representing the nth fading period and assumption that the channel condition remains static within each period T.

After formulating the bit error rate (BER) minimization in terms of $\hat{\mathrm{b}}(t)$ through a general optimization framework, we can directly solve the formulated problem with sufficient training data, due to the non-convex approximation capability provided by neural networks. In addition, since the neural networks provide outstanding generalization capability, the BER performance under imperfect channel conditions can be improved with the deep learning framework, whose loss function is the cross-entropy between the estimation $\hat{\mathrm{b}}(t)$ and the original bits $\mathrm{b}(t)$, i.e.,

$$\mathcal{L} = -\sum_{n=1}^{N}\sum_{t=(n-1)T+1}^{nT}\left[\mathbb{CE}\left(\hat{\mathrm{b}}(t), \mathrm{b}(t)\right)\right] \Big/ M \times N_t \times NT, \tag{2}$$

where $\mathbb{CE}(a, b)$ is defined to be $a\ln b + (1-a)\ln(1-b)$.

To find a network structure suitable for MIMO detection, we have tested two neural networks, deep neural networks (DNN) and convolutional NN (CNN). DNN is often used for joint channel equalization and decoding, while CNN is widely applied to do feature extraction and process correlated noise. For DNN, there are four hidden layers, besides the output layer, input layer and a Batch Normalization layer used to accelerate convergence and prevent overfitting, and for CNN, two convolutional layers are followed by two dense layers, whose architectures are shown in Figure 2.39.



(a) DNN model (b) CNN model

Figure 2.39 Architectures of the networks used in MIMO detection

More specifically, the input of the neural network is the joint matrix of the received symbols and imperfect CSI which have been converted to real domain from complex domain by separating the real and imaginary parts of the

original complex vector. In the hidden layers, the rectified linear unit (ReLU) activation function is adopted. The output of the network is $\hat{b}(t)$ and the activation function in the output layer is the sigmoid function.

Besides, it proves that DNN has better BER performance than CNN through the experiment that we apply these two networks with a close number of total parameters to fairly perform 4×4 MIMO detection with BPSK modulation, and the result is shown in Figure 2.40. In addition, the run time to detect 7.2×10^5 symbols of CNN is nearly 4 times that of DNN. In summary, DNN is more suitable for MIMO detection in term of BER performance and decoding rate.

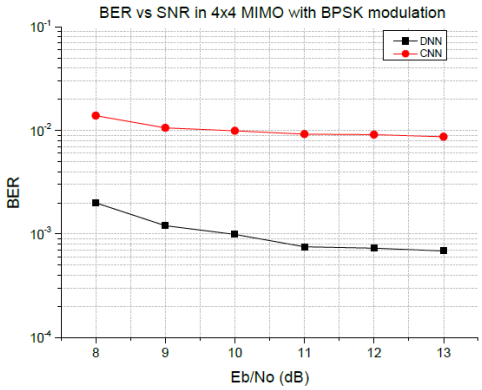


Figure 2.40 BER performance of CNN versus DNN

(3) Performance evaluation

We have compared the DNN-based method with ZF (Baseline 1), MMSE (Baseline 2), DetNet (Baseline 3) and ML-based detection algorithm (Baseline 4) in term of BER performance, robustness and decoding rate. Besides, we consider the spatial correlation and we assume that the fading of the receiver is spatially uncorrelated but transmit-correlated, which is typical in the downlink channel of a mobile communication system. In all the simulations, our networks are trained on data sets generated at 8 dB SNR.

In the first experiment, DNN-based approach is compared with DetNet and traditional methods for 4×4 MIMO detection under different SNRs with full knowledge of the CSI. As shown in Figure 2.41, the BERs of DNN-based method are lower than MMSE and ZF at various SNRs, and even has an advantage of more than 5 dB. So our network has a strong capability of generalization. Besides, for BER of 10^{-3}, the DNN-based method outperforms DetNet in our channel model for 4.5 dB.

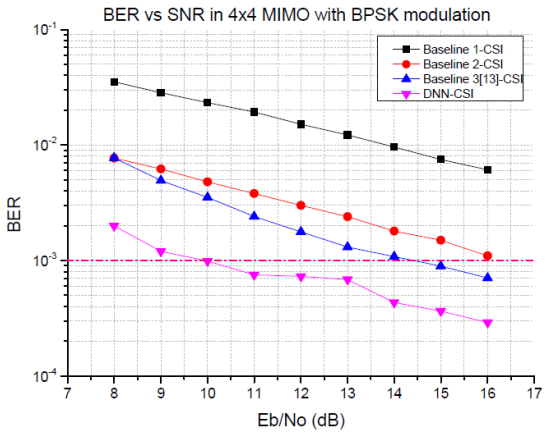
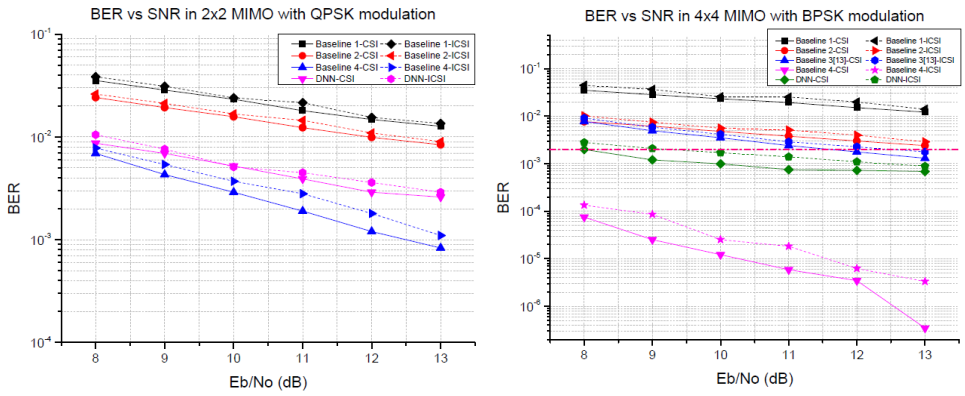


Figure 2.41 BER versus SNR with perfect CSI

The proposed method is compared with traditional methods, including ML with imperfect channel information in the second experiment. The result for 2×2 MIMO detection with QPSK modulation and 4×4 MIMO detection with BPSK modulation are shown in Figure 2.42. We noticed that with imperfect channel knowledge, for BER of 10^{-2}, the proposed DNN-based method still outperforms the MMSE and ZF methods for about 4.5 dB and more than 5 dB when performing 2×2 MIMO detection with QPSK. Particularly, for 4×4 MIMO detection with BPSK, DNN-based approach outperforms the DetNet for about 3.5 dB for BER of 2×10^{-3}, ZF and MMSE for more than 4 dB respectively. This indicates that the proposed method still has a good robustness with imperfect CSI. Moreover, we can notice that the BER of the ML method in the case of imperfect CSI are significantly higher compared with that of perfect CSI. But the fluctuation of DNN caused by the perfection of CSI is similar to other methods. It is worth mentioning that the detecting result of DNN-based method using imperfect CSI is even better than that of DetNet MIMO detection method with perfect channel information. Although the BER of the proposed method is higher than that of the ML method, using the model for MIMO detection has much lower complexity than ML algorithm, which is enough to make up for the deficiency in BER.



(a) 2 MIMO with QPSK modulation (b) 4 MIMO with BPSK modulation

Figure 2.42 BER versus SNR with perfect and imperfect CSI

The higher time complexity the algorithm has, the slower the decoding rate is. To compare the time complexity of different detection algorithms, we calculate the average throughput of them, which equals to the number of detected bits divided by the time consumption. We program and run all MIMO detection algorithms in Python 3.5.2 using an Intel core i5-6500 CPU @3.20GHz with a 12GB memory. We run the test to detect 7.2×10^{-5} symbols and record the time consumption for 3 times, and then calculate the average throughput. In Table 2.6, we compare the detection efficiency of different schemes for 4 × 4 MIMO detection with BPSK modulation. From Table 2.6 we can see the throughput of ZF is highest, and the proposed DNN-based method has very similar performance with ZF algorithm. MMSE followed and ML detection method is lowest. We can conclude that DNN-based method has a near-ZF throughput performance and achieve more accurate decoding performance at the same time, which means that the time complexity of DNN-based method is also low.

Table 2.6 Throughput comparison among different schemes

	ZF	MMSE	ML	DNN-based
Throughput (Kbps)	4.8196×10^4	4.6933×10^4	8.8377×10^3	4.8×10^4

2.6 Terminal Intelligence

2.6.1 Overall Framework Description

(1) Background Description

Current researches mainly focus on the optimization of the whole network, like SON/MDT. The NW may collect groups of UE's info and infer overall situation of one area by extracting the key points from these info and integrating them into a common metric. Finally, a strategy will be made and applied for the cell [22, 23]. Though the strategy is useful for operator to promote the utilization of cell radio resources, it may not appropriate for UE itself.

AI enabled network commonly needs interaction between UE and NW (e.g. gNB). As for single UE, it is gNB/5GC who receives the UE's information (e.g. measurement info, data flows, etc.) and informs UE of NW configurations by dealing with the issues extracted from those information. Surely, gNB/5GC is stronger than UE because of its powerful computing capability and abundant energy supply which means the power consumption for gNB/5GC is not as sensitive as UE. However, time has witnessed the ability of UE to become more and more powerful. Nowadays, UE could also have ability to do some AI model inference work (i.e. on-device capability), it is time to study what can be done if combined with terminal intelligence.

(2) Overall Framework

The overall framework assumed for terminal intelligence is shown in Figure 2.43. As illustrated below, the types of terminals may be various, such as industrial arms, vehicles and robots, and of course including smart phones. For convenience, UE will be called instead of terminal in the following paragraphs.

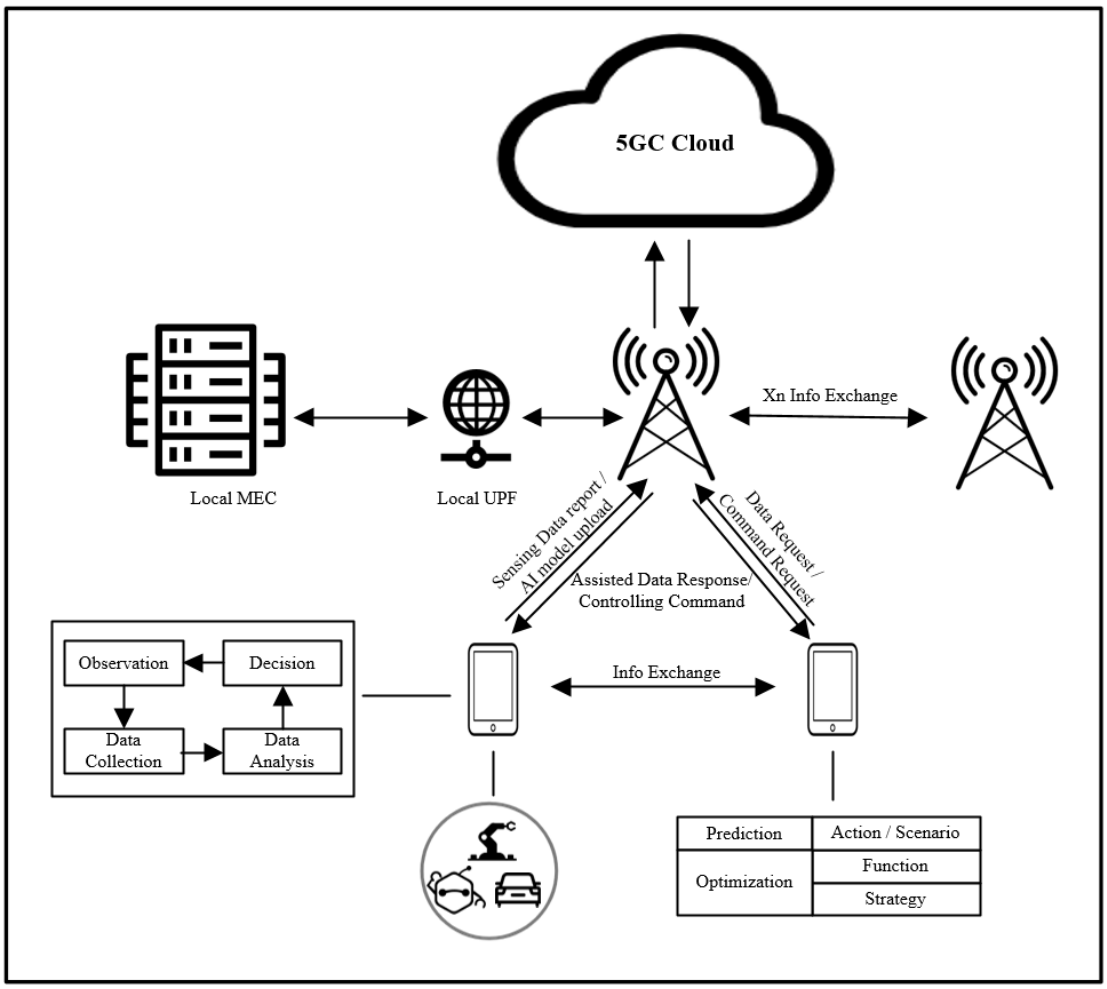


Figure 2.43 The overall framework for terminal intelligence

For terminal intelligence, there exists two kinds of methods: a). Terminal assisted intelligence, which means the enhancement to traditional methods (i.e. the decision is done by NW) with UE accurate assistance information; b). Terminal decided intelligence, which means the decision is made by UE itself.

a. Terminal assisted intelligence

This method will help NW configure a more suitable strategy for UE with UE's specific info though UE may need to tolerate the round-trip delay before receiving the instructions indicated by NW, which could affect delay-sensitive behaviors.

As for the data resources, the gNB who receives UE's unique info (e.g. sensing data/measurement report) will make a UE specific strategy based on UE report, stored UE context, UE specific info retrieved from 5GC (e.g. may

collect from AMF, UDM, OAM, etc.), neighboring cell info via Xn interfaces, and may also include its analysis and prediction of UE packets arrival rate, interval time, etc.

b. Terminal decided intelligence

For this method, the biggest difference from traditional methods is that the UE may have ability to achieve a complete cycle including data collection, data processing and analysis, AI model inference or even training, and making the decision with/without the help from gNB. The advantage of this method is to make a more accurate, more instant and more appropriate strategy for UE, which is helpful to improve user experience.

It has to be explained here that this kind of terminal intelligence doesn't means UE will do all the things alone and without any help or assistant information from NW. In fact, the more various data UE collects, the more appropriate decision UE will make. The data here could be environment info observed by UE itself, neighboring info via PC5 (such as environment info and capacity, etc.), the data from gNB which could be retrieved from 5GC (e.g. AMF, OAM, UDM) and collected from neighboring cell via Xn interface between gNBs.

UE could input the above info into the AI model which is preconfigured in UE or delivered from NW, extracts the key points UE interested and uses them to complete its tasks like prediction of mobility pattern and scenarios where UE is located. Based on the prediction and the assistance info from NW, the UE specific strategy could be optimized.

For now, though the second method challenges UE's capacity and is hard to realize, it still has wide prospects considering the rise of distributed computing and intelligence. Local MEC, as illustrated in Figure 2.43, is deployed near the base station, and will interact with UE to exchange data (like data collection and distribution) or execute task offloading via local UPF frequently [24].

With the help of local MEC, traffic load of NW will be reduced and user experience such as transmission delay and task complete time will be guaranteed [25]. With the growing of UE computing capacity, UE will play an important role in distributed computing and intelligence system, e.g. used for federal learning. The potential use cases that get benefits from terminal intelligence will increase rapidly. Following will show some typical use cases.

2.6.2 Use case 9: Terminal assisted network configuration optimization

(1) Description

There exists almost fixed cells or tracking areas (TAs) where UE visits frequently (see Figure 2.44), like way to company and way home. For these cases, UE will send requests to gNB to get the network configuration when enters a new cell/ TA in traditional methods, which will cause severe signaling overhead.

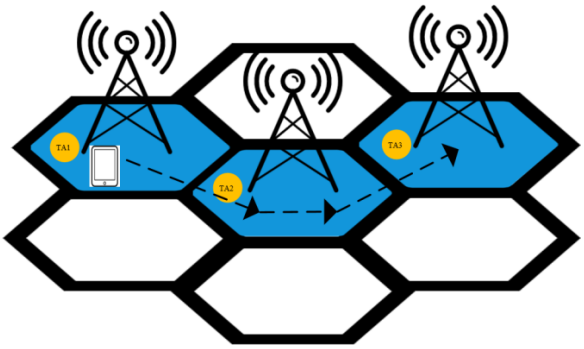


Figure 2.44 TA prediction using terminal intelligence

(2) Solution

Things will be different when combined with terminal intelligence, and the signaling overhead will be reduced either. The key point is to take consideration of user measurement results (some metrics may be inferred by UE itself including the position, direction and even velocity of UE) and the history info stored in UE.

As for history info, although 5GC may have this kind of information either, but the UE ID is unknown to gNB and the info stored in UE is more comprehensive and more representative. With the help of terminal intelligence, the above info will be analyzed, and used by UE to predict its residential cell/TA List.

UE reports its prediction of cell/TA list to serving gNB. And according to UE's report, serving gNB could set and send network configuration of those cell/TA list back to UE in advance, including measurement configuration, handover strategies, dual connectivity configuration, RAN Notification Area (RNA) configuration and so on. In this way, the signaling overhead will be reduced.

2.6.3 Use case 10: Terminal decided model update

(1) Description

The arising of AI makes it popular in communication especially in physical layer module enhancement. As illustrated in Figure 2.45, the receiver is designed to decode RX signal timely and properly and has independent modules relatively. Among these modules, the most concerned module combined with AI in academic are Channel estimation [26], MIMO detection and Channel decoding, respectively.

Taking channel estimation as an example here. Limited by the generalization of AI model, it cannot be used for all the cases [27]. Especially, as the condition changes rapidly, such as the range of signal to noise ratio (SNR) varies widely, the mismatch of input data (the PDCCH size), the performance of AI will suffer severely and lower than the traditional methods or even unusable completely.

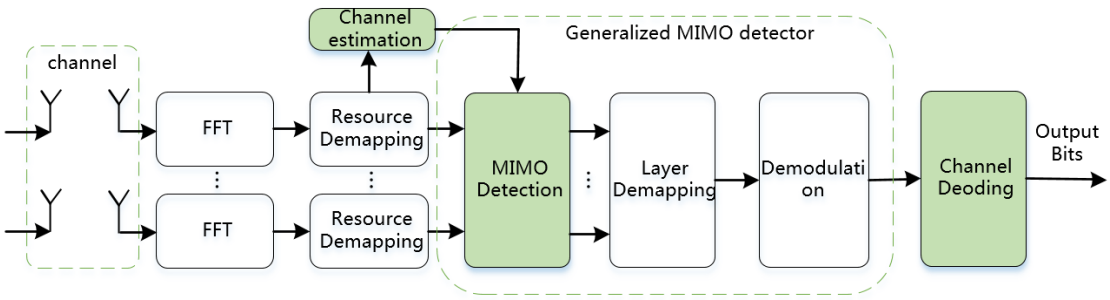


Figure 2.45 the relationship of main modules in Receiver

(2) Solution

Here lists some possible solutions by utilization of terminal intelligence:

1. Data enhancement

It requires UE to check whether the input size match the requirement of the deployed AI model. If mismatch, UE should do date enhancement and ensure the data could be used by the deployed AI model. As for the AI model, it should also consider this situation and be trained with the data set including these kind of enhanced data.

2. AI Model update

When the condition changes and the performances suffers severely, UE should discover this issue no matter by notification from NW or detected by its terminal intelligence. And it could be solved by following methods:

a. By UE itself

Due to the limitation of model generalization, UE may store a set of candidate AI models. In this cases, when the issue happens, UE could pick up an appropriate AI model based on its measurement results.

If the on-device AI capability of UE is powerful enough, UE may refine the AI model using its personal data and doesn't need to worry about the security issue of privacy leakage which may happen if UE sends its privacy data to NW. With the help of terminal intelligence, UE could even predict the probability of the happening of this issue and do preparation in advance.

b. Fallback to traditional methods

UE could fallback to traditional methods to guarantee the bottom line of its performances.

2.6.4 Use case 11: Network-Assisted Multi-Vehicle Collaboration

(1) Description

The Internet of Vehicles (IoV) is an integration of three networks: an inter-vehicle network, an intra-vehicle network, and vehicular mobile Internet [28]. Within IoV, how to efficiently

coordinate the multi-vehicle system (or multi-agent system, MAS) to exhibit intelligent behaviors and complete high-complexity tasks by each single agent remains a severe issue. However, in a distributed MAS, it is difficult to coordinate each agent who can only observe local information, while a centralized MAS faces severe issues due to huge amounts of computing and communications overhead.

Stigmergy is an indirect communication mechanism existing in social insect species, which relies on "pheromones" deposited in the environment to coordinate individual insects' activities, and exhibit strong robustness with a simple form [29]. Together with federal learning, we propose a decentralized control model based on stigmergy mechanism in network-assisted MAS [30]. In particular, we employ digital pheromones (DP) to simulate existence of pheromone in the environment and use them as medium in stigmergy mechanism to enable indirectly communicate between agents. Besides, we design an experimental scene to mobilize several distributed agents to arrive at tracking points with as few rounds as possible, and exploit KHEPERA IV robots to implement the algorithm in practical settings.

(2) Solution

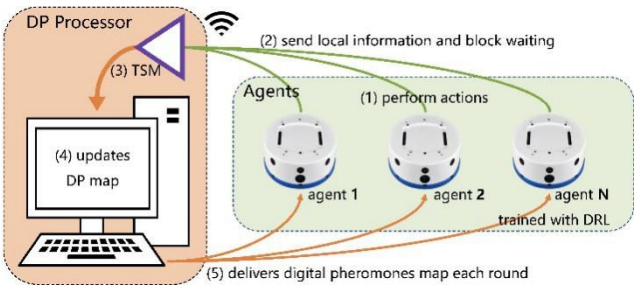


Figure 2.46 Decentralized control model based on stigmergy mechanism.

Based on implicit information such as digital pheromones, agents can obtain the local environmental state within the perceivable range without direct communication. Figure 2.46 shows this decentralized control model based on the stigmergy mechanism. It consists of two parts: DP processor and agents, which form a network to connect multiple agents to establish communication.

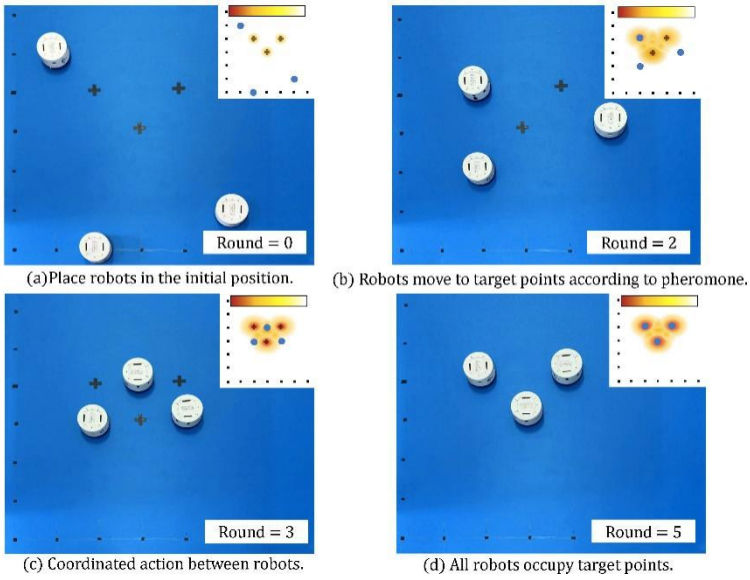
- **DP processor**. DP processor collects local information with the procession of temporal synchronization mechanism (TSM), and calculates the DP of agents' location based on the distance between agents and target points. With simulating the characteristics of superposition, decay and diffusion, DP processor generates a DP map and delivers it to agents each round. To prevent information mismatch when all agents communicate with DP processor, TSM is introduced to follow strict timing requirements when interacting between agents and DP processor. Similar to a buffer, TSM sacrifices part of the agents' moving time to coordinate the behavior of distributed agents, as well as reduces the probability of collision between the agents to some extent. We will

investigate the impact of asynchronous methods in the future.

- **Agents**. All agents only obtain the local environmental information through interaction with DP processor without direct communications with each other. Each agent has a definite perception range for pheromone, and the volume of pheromones decreases with the increase of distance, which attracts agents to gather at location with high pheromones volume. However, distributed agents often cannot cooperate well to accomplish tasks efficiently. Under the guidance of pheromone, we use asynchronous advantage actor-critic (A3C) algorithm in deep reinforcement learning (DRL) to train the agents, encode local environment information of agent as *state*, and calculate the difference between distance to the target points before and after each action as *reward*. This learning method can train agents' sense of collaboration to take global optimal *action* in the current state.

Figure 2.47 illustrates the whole execution procedure of the system, which repeats recursively until all agents reach the target points.

(3) Problem Evaluations



(a)Place robots in the initial position. (b) Robots move to target points according to pheromone.
(c) Coordinated action between robots. (d) All robots occupy target points.

Figure 2.47 The experimental platform of 3 robots with above model.

The "+" in the above figure represents the target points. The upper right corner of each figure represents the DP map. The darker the color, the higher the pheromones volume.

We establish an experimental platform based on a DP processor and 3 KHEPERA IV robots, and evaluate performance in terms of degree of completion (DoC) and number of total steps.

- Use a PC with a WiFi module as the DP processor, whose CPU processor is Intel(R) Core (TM) i5-3230 CPU @ 2.60GHz. Actually, a controller connected with legacy cellular networks is also applicable.
- Use 3 KHEPERA IV robots as moving agents. Since the robots are not equipped with positioning modules, we

employ odometry and leverage dead reckoning method to estimate the position based on given initial position and orientation of the robots. The CPU processor of robots are Texas Instruments DaVinci DM3730 @ 800MHz.

Figure 2.48 gives the experimental results of 3 randomly placed robots, which shows robots reach the target points (marked by "+") from its initial position within a certain number of rounds. Similar to the real environment, we assume that a small number of pheromones exist at the target points, as shown in (a) of the figure. The amount of DP sensed by agents increases, thus motivating the agents to approach the target points in a stronger manner, which can attract more agents to approach the target points, as shown in (b), (c) and (d) of the figure.

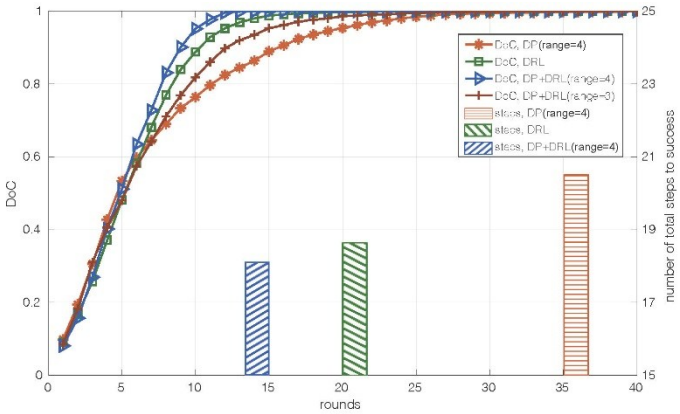


Figure 2.48 The performance of DP and DRL combined compared with the cases of adopting DP or DRL only.

Furthermore, Figure # presents the numerical results in detail. When DoC arrives at 1, it means that all robots have reached the target points, and the histogram shows the number of rounds and total steps that the robots take to complete the goal. It can be observed that compared to the cases of adopting DP or DRL only, the proposed combination of DP and DRL yields superior performance. It also can be seen that the performance is greatly reduced when the range of agents' perceptual pheromone becomes smaller (from 4 to 3). In particular, DP can attract agents to places with high pheromones volume at the beginning, and DRL can coordinate the behavior of agents when agents move around the target points.

3 Intent Aware Wireless Network

3.1 Intent API

Fundamentally, Intent API should include operations for lifecycle management of intents, which allows its consumer to express intents for managing the network/services and obtain the feedback of intent fulfillment result. The lifecycle management operations of intents may include intent creation, intent activation/de-activation, intent

deletion, intent modification and intent query. Decoupled from lifecycle management operations, intent expressions act as the parameters of these operations and is delivered to the producer. Intent expressions should be written in a certain domain-specific language, e.g. {Intent Driven Action, Intent Driven Object} which is defined in 3GPP SA5 TR 28.812.

To support flexible customization of intent expression, the grammar of intent language should be editable and related API should be exposed to intent designers. For example, intent schema could be applied to define and restrict the grammar of intent language where legal words could be derived from a pre-defined dictionary. Additionally, an intent schema corresponds to an intent-driven management scenario and the extension of intent language could be realized by adding new intent schemas. Consequently, the operations for intent schema management should be provided, which may include schema creation, schema deletion, schema query, etc.

3.2 Use Cases for intent aware wireless Network

3.2.1 Use case 12: Intent-Driven Mobility Load Balancing

(1) Description

Mobility Load Balancing (MLB) is an important use case of the self-organized networks (SON), which can transfer the load from heavy-loaded cells to light-loaded cells through the handovers of users and achieve a balanced load distribution [31]. However, there exist several limitations in current MLB methods [32][33][34]. On the one hand, the MLB methods focus on the offloading of heavy-loaded cells but ignore the service and experience of transferred user equipments (UEs). On the other hand, the adjustment of mobility parameter may cause a large number of handover, many of which are unnecessary in fact. The increasing transmission capacity brought by mobile network evolution will cause serious challenges of network optimization and resource allocation. So it is necessary to research the novel MLB scheme to meet the demand of the next-generation communications.

To solve above issues, we introduce IDN [5] into MLB and propose an intent-driven MLB (IDMLB) method to reduce the handover times of UEs during load-balancing. We take the network intent (e.g., offloading) and UE intent (e.g., bandwidth) into consideration to design a more fine-grained handover scheme and avoid the deterioration of the QoE (Quality of Experience) of UEs after the handover. Specifically, we design the user utility function of bandwidth and the network utility function of load to represent UE intent and network intent respectively. Then the base station transfers the UEs for offloading according to the two utility functions, which will result in less handover times than traditional methods.

(2) Solutions

The intent in MLB can be classified into the UE intent and the network intent. And there exists a potential conflict between the network and UEs. Because UEs always tend to choose the network with high QoS (quality of service), which will cause the overload of a network. On the other hand, the network needs to maintain the level of the load within a certain range, which will deny the access of some UEs when a network is overloaded. Such conflict will lead to the instability of the network and one of the most obvious results is a large number of handover times of UEs. Hence, it is reasonable and feasible to implement IDN in MLB. We design an IDMLB scheme to solve the conflict of different intents in MLB, which considers UE intent and network intent in the network to perform a more finely offload and reducing the unnecessary handover during MLB [35].

The architecture of IDMLB is shown in Figure 3.1. The controller, which generally represents the control module or system of a network, receives the UE intent and the network intent through the Northbound API and receives the state of networks and the UEs through the Southbound API. The IDMLB module in the controller generates handover policies and delivers them to the network infrastructure as configurations through Southbound API.

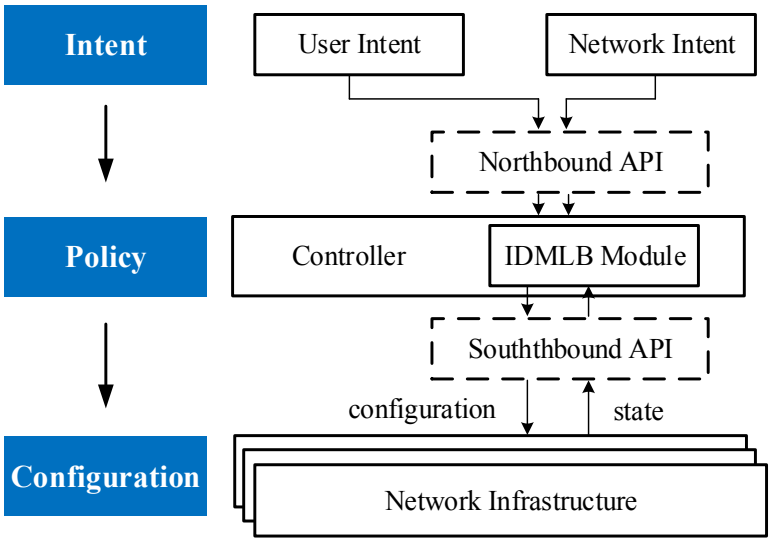


Figure 3.1 The framework of IDMLB

We model the user intent and the network intent using utility functions [36] and integrate them into an MLB utility function to guide the implement of handovers. Then we give the procedure of IDMLB.

a) UE intent

UE intent indicates the user demand for various network characteristics, which can be network bandwidth, network delay, and network cost.

There are various kinds of network services, which potentially indicate different UE intents. UEs tend to be

associated to the networks which can satisfy their intents. Different networks are usually with different network characteristics. Therefore, we design a utility function for the UE whose intent is $\{L_0, D_0\}$ as

$$G_u = g^{1/2}(L^{-1}, L_0^{-1}) \times g^{1/2}(D, D_0),$$

where L and L_0 are the current delay and demanded delay of the user. D and D_0 are the current rate and demanded rate. And g is a utility function for a specific network characteristic, which is formulated by the a type of sigmoid function and can be expressed as

$$g(x, x_0; \eta, \sigma) = (1/2)\{\tanh[\log(x/x_0) - \eta]\sigma + 1\},$$

where x and x_0 are the current network indicator value and demanded value of the user respectively. The terms η and σ are the threshold parameter and the spread parameter to make the value of utility function range in [0,1] and set $g = 0.5$ hold when $x = x_0$.

b) Network intent

The network intent indicates the network demand for load balance or the usage ratio of PRBs (Physical Resource Blocks), which can be predefined parameters such as the thresholds of heavy-loaded state and light-loaded state.

A network need avoid the state of heavy-loaded. In MLB, a network includes the source cell (heavy-loaded cell) and the target cell (one of the neighbour light-loaded cells). The intent of the source cell is to offload the network load as soon as possible, but the target cell need avoid being overloaded while accepting the users from source cell. Therefore, the network utility function represents the overall utility of a network when transfer a specific user, and consists of the utilities of the target cell and the source cell.

We first define the utility function about the load for a common cell as

$$H_i(\rho) = \begin{cases} 1 & \rho < \rho_{lth} \\ -\dfrac{(\rho - \rho_{lth})^2}{(\rho_{hth} - \rho_{lth})^2} + 1 & \rho_{lth} \le \rho \le \rho_{hth} \\ 0 & \rho > \rho_{hth} \end{cases},$$

where ρ is the current load state, ρ_{lth} and ρ_{hth} are the predefined light-loaded and heavy-loaded thresholds respectively. When a cell is in a light-loaded state, users are easier to access and the more the user access, the higher the resource utilization rate is, thus the value of utility is 1. When the cell is in the heavy-loaded state,

receiving more UEs will cause a negative impact to the quality of service of the cell, thus the utility is 0. When the cell is in the middle load state, with the increase of load, the utility of the receiving user decreases. Therefore, the heavy-loaded cell tends to offload the UE with heavy load to obtain a larger gain of network utility, and the target cell tends to receive a user with smaller load. The utility function implies the intent of the cell in MLB.

For a network composed of a target cell and a source cell, the utility is expressed as

$$G_{net}\left(u_{ho}\right)=\Delta H_{i_s}\left(\rho_{u_{ho},i_s}\right)+\Delta H_{i_t}\left(\rho_{u_{ho},i_t}\right),$$

where u_{ho} is the user which can be transferred from source cell i_s to target cell i_t, ρ_{u_{ho},i_s} and ρ_{u_{ho},i_t} are the load of the transferred UE to the source cell and target cell respectively, which can be calculated by

$$\rho_{u_{ho},i}=\frac{N_{u,i}\cdot\omega}{W},$$

where W is the bandwidth of a cell, and $\Delta H_{i_s}\left(\rho_{u_{ho},i_s}\right)$ and $\Delta H_{i_t}\left(\rho_{u_{ho},i_t}\right)$ are the variation of the load utility of source and target cells, which can be expressed as

$$\Delta H_i\left(\rho_{u_{ho},i}\right)=H_i\left(\rho\pm\rho_{u_{ho}}\right)-H_i\left(\rho\right).$$

c) Intent-Driven MLB

Considering the user intent and the network intent, the utility of one handover can be calculated by

$$U\left(i_s,i_t,u\right)=G_u\times G_{net}\left(u\right).$$

The utility function represents the utility of transferring a specific user from the source cell to the target cell, which can be used to direct the handover of users.

(3) Performance evaluation

To evaluate the performance of the proposed IDMLB, we simulate the proposed IDMLB in LTE network and compare the number of handover and the throughput to traditional A3-event-based MLB method. The simulation parameters are given in the following table 3.1.

Table 3.1 Simulation parameters

Parameter	**Value**
System Bandwidth	10MHz (50PRBs)
Inter-Site Distance	1000m
Transmit Power	46dBm
ρ_{lth}、ρ_{hth}	0.6 0.9
Resource Schedule	Max CIR
Network Model	19 * LTE Cell
UE Moving Speed	3 km/h
Channel Model	SCM MIMO 2 * 2
Traffic Model	Poisson Distribution

The number of handover and the throughput of the 19 cells during simulation duration are shown in Figure 5 and Figure 6. As shown in Figure 5, our method can significantly reduce the number of handover compared to the conventional A3-event-based MLB method. On the other hand, it can be seen from Figure 6 that the IDMLB method has a lower throughput in comparison with the A3-event-based MLB method, which is acceptable and reasonable.

As can be seen from Figure 3.2, the handover number of MLB increases after 20 minutes. Meanwhile the throughput of MLB and IDMLB in Figure 3.3 also become different at 20 minutes. It indicates that the network load surges at 20 minutes. As for traditional MLB method, the user will be preferentially transferred to a cell with a larger RSRP, which can provide a larger SINR, thereby achieving greater throughput, while IDMLB is not simply pursuing RSRP. Therefore it is a trade-off between the number of handover and the throughput.

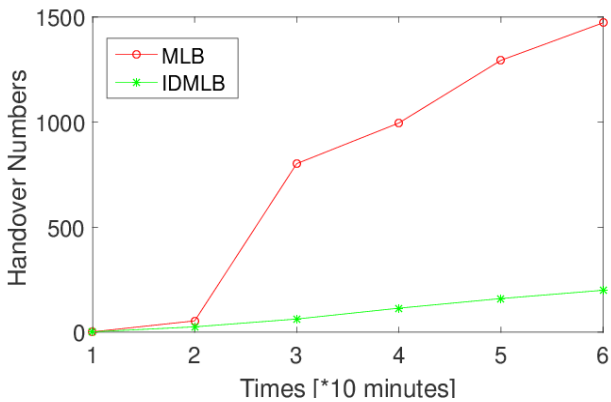


Figure 3.2 The comparison of handover numbers

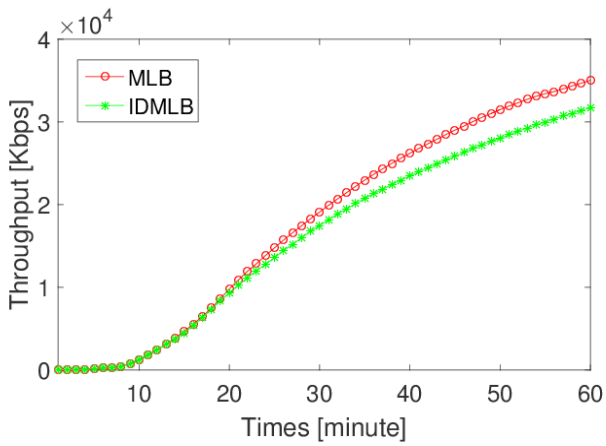


Figure 3.3 The comparison of throughput

3.3 Enhanced Architecture for the intent aware wireless Network

As a building block of network automation and autonomy, intent aware wireless network relies heavily on a closed control loop, which includes network context awareness (Observe), network data understanding and analysis (Orient), choosing the right commands to maintain the intent objective (Decide) and command delivering (Act). As one of the few SDOs that follow such an OODA model, ETSI ISG ENI works on defining a system architecture which contributes to reduce network operators' operational expenditure (OPEX). ENI system applies policy-driven closed control loops that use emerging technologies, such as big data analysis, analytics, and artificial intelligence mechanisms, to adjust the configuration and monitoring of networks and networked applications, and dynamically updates its acquired knowledge to understand the environment, including the needs of end-users and the goals of the operator [38]. Since intent can be regarded as a declarative policy, ENI system can be used as a reference implementation architecture for intent aware wireless network.

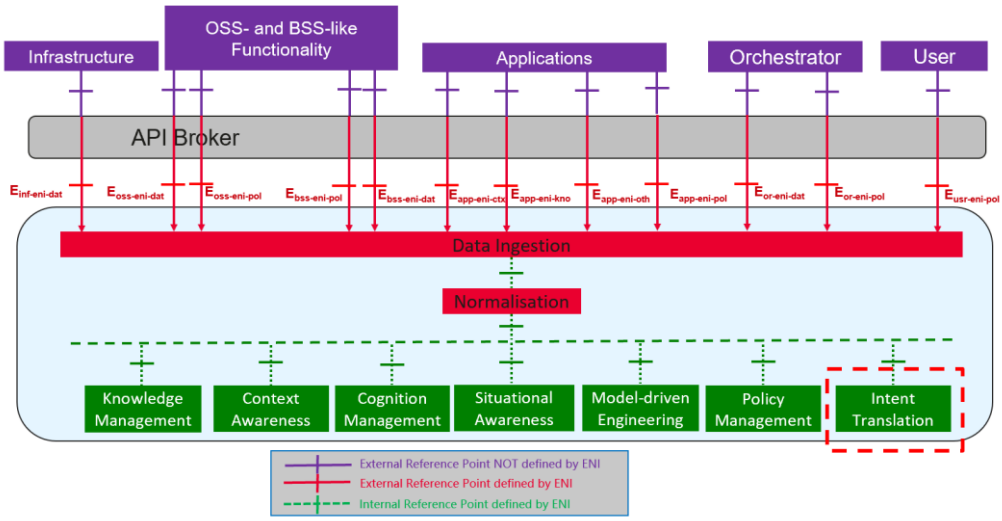


Figure 3.4 Enhanced ENI architecture for intent aware wireless network

Figure 3.4 is a high-level Functional Block diagram for enhanced ENI architecture for intent aware wireless network. This proposed architecture separates Intent Translation Function Block from Policy Management, which is another option for the architecture defined in ENI-005[38]. It includes the following main Functional Blocks for intent processing:

- Intent Translation. It performs lexical analysis, syntactic analysis, semantic analysis and augmentation, compiling of intents. An intent is translated from a high-level of abstraction to a more concrete form in order to be fulfilled.
- Knowledge Management. It stores the knowledge that is used in the process of Intent Translation. The knowledge contains the relevant metadata (e.g. a time period that this intent is valid, as well as version information, including a minimum version that shall be used) and the generated new knowledge (e.g. policies or rules for RAN domain, the corresponding actions to meet certain requirements of an intent).
- Context Awareness. The purpose of the Context-Aware Management Functional Block is to describe the state and environment in which a set of entities in the Assisted System (i.e. EMS or gNodeB being assisted) exists or has existed. In an intent aware wireless network, Context Awareness can be used as a network sensor which measures and collects wireless network performance data.
- Cognition Management. The purpose of the Cognition Management Functional Block is to enable the ENI System to understand normalized ingested data and information, as well as the context that defines how those data were produced; once that understanding is achieved, the Cognition Management Functional Block then evaluates the meaning of the data, and determines if any actions need to be taken to ensure that the goals and objectives of the intent are met.
- Situational Awareness. The purpose of the Situation Awareness Functional Block is to enable the ENI System to be aware of events and behavior that are relevant to the Assisted System (e.g. EMS or gNodeB being managed). This includes the ability to understand how recommended commands given by the ENI System impact the management and operational goals of the intent, both immediately and in the near future
- Model-driven Engineering. The purpose of the Model Driven Engineering Functional Block is to use a set of domain models that collectively abstract all important concepts for managing the behaviors of managed objects in the RAN governed by the ENI System. The use of reusable models defines a set of concepts that are shared by all constituencies that use them. It is permissible for a given constituency to use them directly or indirectly

- Policy Management. The purpose of the Policy Management Functional Block is to provide decisions to ensure that the intent goals and objectives are met. When an intent is violated, Policy Management should be able to adjust the actions to assure the intent fulfillment.

$E_{oss\text{-}eni\text{-}pol}$/$E_{bss\text{-}eni\text{-}pol}$/$E_{usr\text{-}eni\text{-}pol}$ can be used to define intents and associated information and/or metadata exchanged between the OSS-like Functionality/BSS-like Functionality/User and the ENI System that control behavior (including services and resources) for the Assisted System. Intents and information received by the ENI System shall be acknowledged by the ENI System.

New internal reference points is defined for intent implementation inside the ENI System. The internal reference points are shown in Figure 3.5. Table 3.2 provides brief descriptions of the internal reference points of ENI for intent implementation.

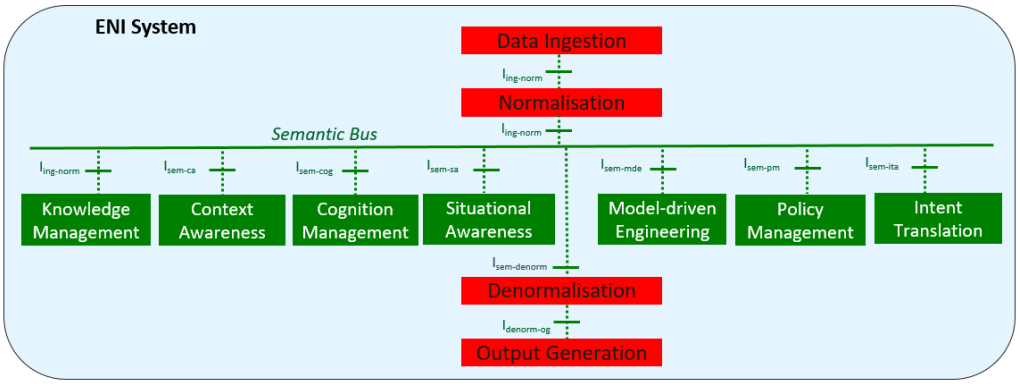


Figure 3.5 Overview of ENI internal reference points for intent processing

Table 3.2 Information and data exchanged for intent processing

Name	receives from Internal Reference Point	publishes to Internal Reference Point
Data Ingestion Functional Block	/	Publishes intent processing messages (e.g. the "newPolicy" message) and network performance data for intent to Normalisation through $I_{ing\text{-}norm}$.
Normalisation Functional Block	Receives intent processing messages (e.g. the "newPolicy" message) and network performance data for intent from Data Ingestion through $I_{ing\text{-}norm}$.	Publishes normalised intent processing messages (e.g. the "normalisedPolicy" message) and network performance data for intent to the Semantic Bus through $I_{norm\text{-}sem}$, where subscribed Functional Blocks may consume the normalised data and information.
Knowledge Management Functional Block	Receives intent processing messages (e.g. "normalisedPolicy" message), network performance data, translated intent, additional context for intent from the Semantic Bus through $I_{sem\text{-}km}$.	Publishes intent processing messages (e.g. the "translatePolicy" message) and intent knowledge to the Semantic Bus through $I_{sem\text{-}km}$, where subscribed Functional Blocks may consume the data and information.

Intent Translation Functional Block	Receives intent processing messages (e.g. “translatePolicy” message) and intent knowledge from the Semantic Bus through $I_{sem\text{-}ita}$.	Publishes translated policy and intent processing messages (e.g. the “polComplete” message) to the Semantic Bus through $I_{sem\text{-}ita}$, where subscribed Functional Blocks may consume the data and information.
Context Awareness Functional Block	Receives intent processing messages (e.g. “polComplete” message), translated intent, network performance data for Intent from the Semantic Bus through $I_{sem\text{-}ca}$.	Publishes intent processing messages (e.g. “ctxComplete” message), translated Intent with additional context to the Semantic Bus through $I_{sem\text{-}ca}$, where subscribed Functional Blocks may consume the data and information.
Situational Awareness Functional Block	Receives intent processing messages (e.g. “ctxComplete” message), network performance data, translated intent from the Semantic Bus through $I_{sem\text{-}sa}$.	Publishes intent processing messages (e.g. “ctxComplete” message), translated intent with relevant situation information to the Semantic Bus through $I_{sem\text{-}sa}$, where subscribed Functional Blocks may consume the data and information.
Model- Driven Engineering Functional Block	Receives intent processing messages (e.g. “ctxComplete” message), translated result for intent from the Semantic Bus through $I_{sem\text{-}mde}$.	Publishes intent processing messages (e.g. “mdeComplete” message) and transformed intent to the Semantic Bus through $I_{sem\text{-}mde}$, where subscribed Functional Blocks may consume the data and information.
Policy Management Functional Block	Receives intent processing messages (e.g. “mdeComplete” message) and transformed intent from the Semantic Bus through $I_{sem\text{-}pm}$.	Publishes intent processing messages (e.g. “genPolicyOutput” message) and formatted intent which needs to be reformatted the Semantic Bus through $I_{sem\text{-}pm}$, where subscribed Functional Blocks may consume the data and information.
Denormalisation Functional Block	Receives intent processing messages (e.g. “genPolicyOutput” message) and formatted intent from the Semantic Bus through $I_{sem\text{-}denorm}$.	Publishes denormalized final results and intent processing messages (e.g. “genPolicyOutput” message) to the Output Generation Functional Block through $I_{sem\text{-}denorm}$.
Data Ingestion Functional Block	Receives denormalized intent processing messages (e.g. “genPolicyOutput” message) and reformatted intent from Denormalisation through $I_{denorm\text{-}og}$.	/

3.4 Key Technologies

3.4.1 Intent knowledge base

Intent knowledge base helps to translate an abstract intent expression to specific network configuration operations. The operation could be a command with specific/dynamic parameter values, an optimization problem or a sub-intent. In addition, translated operations may be different in different network scenarios, so the applicable conditions of the operations should be provided. Besides, since an intent only describes the network goals that need to be achieved and does not involve specific implementation details, the criteria for determining whether the intent is achieved or not that can be understood by the network should be also provided in the intent knowledge base.

O&M expert experience can be accumulated in intent knowledge base through knowledge graph technology. Consolidated O&M expert experience and accumulated intent data can be easily promoted to other domain's intent aware network (e.g. vertical industry networks). In addition, knowledge graph can be used to discover the association between experts' experience. O&M experts can mark relationships between intent knowledge for specific scenarios in a centralized and visualized manner and can conveniently manage intent knowledge base of multiple versions.

3.4.2 Cross-layer intent collaboration

When a high-level intent system receives an intent from the operator or higher level system, it decomposes the intent to sub-intents and send them to multiple low-level intent systems (e.g. NMS sends sub-intents to multiple EMS). For intents with specified measurement target (e.g. throughput, loss rate), the high-level intent system only needs to ensure that the statistical measurement (e.g. average throughput, total loss rate) meets the intent target and there is no need for each low-level systems to achieve the same target. In this case, cross-layer intent collaboration should be taken into consideration.

After a high-level intent is decomposed, it may be unsatisfied in the following situations:

- Due to different available network resources, the network capability of some low-level intent systems is insufficient, and the network capability of other low-level intent systems may be higher than that of sub-intents.
- The change of the network status affects the ability of the low-level intent system to achieve the intent. The decomposed sub-intents based on the current network status may not adapt to the future network

status. Once a sub-intent is not satisfied, the whole intent will fail to be satisfied.

Therefore, intent decomposition of high-level intent system should be adjusted as the network status changes to improve the success rate of achieving the overall intent.

The energy saving intent is used as an example. The NMS of a network receives an intent that the energy consumption of all base stations in area A decreases by 20%. Base stations in this area are managed by two EMSs (EMS1 and EMS2) because they are located at the boundary of different EMSs or provided by different vendors. Therefore, the NMS needs to decompose the overall intent to two sub-intents. Whether the intent is achieved or not is related to overall energy consumption reduction of base stations managed by EMS 1 and EMS 2, rather than energy consumption reduction of base stations managed by EMS 1 or EMS 2.

Given an initial intent decomposition (for example, energy consumption of base stations managed by two EMSs is reduced by 20% each), consider the base station load of EMS1 is too heavy to achieve the goal of 20% due to network changes. However, the base station load of EMS2 is light and is sufficient to achieve the goal of 20%. In this situation, the overall intent is not achieved. To ensure the achievement of the overall energy saving intent, NMS could properly raise the goal of EMS2 (e.g. 30%) to try to achieve the overall 20% energy saving target through negotiation with EMS2.

3.5 Technical Key Issues and challenges

Although intent aware wireless network has been studied widely, there are still a few technical key issues and challenges to be solved:

- The completeness of intent API. Network operation and maintenance is typically divided into four stages, including planning, provisioning, maintenance and optimization. So far intents for network provisioning and network optimization are discussed most frequently and deeply. Considering that network planning and maintenance involve a large amount of manual operations currently, whether they could be driven by intents as well and implemented in a closed-loop manner still needs to be studied.
- The ultimate form of intent language. As discussed in Section 3.1, intents are currently written in domain-specific language (DSL) which is a type of human-understandable language with an abstraction of underlying implementation. To further simplify the operations and make the user interface more intuitive, intent language may possibly evolve from DSL to natural language gradually. On the one hand, natural language contains richer semantics than DSL and could support more expression scenarios where intents are needed. On the other hand, the redundancy and ambiguity in natural language bring great difficulty to the

design and implementation of intent translation and validation. Due to the trade-off between expression capability and implementation complexity, intent language may compromise to an intermediate form.

- Integration between AI and intent aware wireless network. Wireless network has huge data, which is the biggest advantage of intelligent operation and maintenance. However, the use of wireless data has become a challenge for AI in network operation and maintenance because of the high data dimension, huge amount of data and many missing data in the current wireless network. How to use artificial intelligence algorithms to support powerful analysis, judgment, and prediction capabilities, combined with the design, construction, maintenance, operation and optimization of wireless network, is the focus of the industry.
- The network side and the user side are loosely coupled, and it is difficult to accurately align network delivery with business intents. In the future, the network and users will be regarded as a unified whole, and the intelligence needs of users will be further explored and realized. Facing the needs of the Internet of Everything in all walks of life, new services continue to emerge. Existing networks use a single network feature to support multiple services, and it is difficult to provide customized network guarantees based on service features. The separation of application services from the network cannot achieve timely delivery of the network and closeness to business requirements, thereby affecting the QoS and QoE. The precise matching of network delivery and user needs is an important research content to improve user experience.

4 4 Progress in SDOs and Consideration

4.1 ITU-T ML5G

In July 2019, ITU-T SG13 approved the Architectural framework for machine learning in future networks including IMT-2020 submitted by the focus group in March 2019 as a technical specification.

Now, ITU-T FG-ML5G has started the second phase (which lasts until July 2020). The following FG ML5G specification were submitted to ITU-T SG13 for consideration at its 20-31 July 2020 meeting:

- "Requirements, architecture and design for machine learning function orchestrator"
- "Serving framework for ML models in future networks including IMT-2020"
- "Machine Learning Sandbox for future networks including IMT-2020: requirements and architecture framework"
- "Machine learning based end-to-end network slice management and orchestration"
- "Vertical-assisted Network Slicing Based on a Cognitive Framework"

4.2 3GPP

4.2.1 SA1

3GPP SA1 approved the “Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS” in Dec 2019, which target finish at March 2021. The objectives of the study include:

- Studying the use cases and potential service and performance requirements for identifying traffic characteristics of AI/ML model distribution, transfer and training for various applications, e.g. video/speech processing, automotive, other verticals, including:
- Gap analysis on performance requirements for AI/ML model distribution and transfer.

4.2.2 SA2

3GPP SA2 Enablers for Network Automation for 5G’s fRel-16 work finished this June 2019. It’s output has been documented in New TS23.288, "Architecture enhancements for 5G System (5GS) to support network data analytics services (Release 16)". In this TS, following aspects have been defined:

- the reference architecture for data analytics;
- network data analytics functional description;
- procedures for analytics exposure, for data collection,
- procedures for specific analytics for following use cases: Slice load level, Observed Service experience related, NF load, Network Performance, UE mobility, UE communication, and Expected UE behavioral parameters;
- NWDAF services description.

3GPP SA2 now continues this topic’s phase 2 work in Rel-17, and the study target will be finished in December 2020. This study focuses on: logical decomposition of NWDAF function, multi-NWDAF architecture, trained model sharing, leftover topics from Rel 16 like Slice SLA guarantee, UE driven data analytics etc., new use cases such as UP optimization for Edge Computing, NWDAF assisted RFSP adjustment, . The output is capture in the TR 23.700-91 " Study on enablers for network automation for the 5G System (5GS); Phase 2 (Release 17)".

4.2.3 SA5

3GPP SA5 Intent driven management service for mobile network (IDMS_MN) started in Sep. 2018, including two phases: study item and work item. This work aims to study mobile network communication intent-driven

management scenarios that improve O&M efficiency and define intent-driven management service interfaces to implement automatic closed-loop control, and to implement simplified control of complex networks or services.

The study item is expected to finish in end of 2020. The objective of it is as follows:

- Investigate the potential scenarios which could benefit to the operators from the intent driven management services.
- Study the scenarios which may utilize automation mechanisms including SON to satisfy the intent driven purposes.
- Investigate the appropriate mechanism to describe the intent.

Currently (V0.a.0, March. 2020), the content of its document (3GPP TR 28.812) contains the concepts, 17scenarios,the standard consideration of intent driven management service and 2 key issues.

The work item will start in Aug. 2020, whose objective is:

- Specify the typical management scenarios which operators could benefit from the intent driven management services in the multiple vendor environment. The scenarios should help to improve the multiple vendor operational efficiency on network management.
- Specify the intent driven management services' requirements which are derived from the scenarios above.
- Specify the intent driven management services which may include the management operations, management entities and management information.

In September 2019，3GPP SA5 also started a new study on enhancement of Management Data Analytics Service (MDAS), its objectives are as follows [21]:

- The use cases, potential requirements and possible solutions regarding the relation and interaction between MDAS and other NF functionalities (including NWDAF and SON functionalities).
- Management aspects (e.g., in connection with Machine Learning techniques) of Management Data Analytics.
- Development of scenarios (including exploration of potential consumers), use cases, potential requirements and possible solutions (including the recommended probable actions) for the following aspects, based on the MDAS concept:
- Root cause analysis for ongoing network and service configuration and performance related issues;
- Prevention of potential issues affecting the network and service operation, e.g., the preventative actions are required to avoid network and service performance degradation;
- Prediction of network demands (e.g. capacity upgrades and new/re-deployments) to accommodate

potential load change and/or new service deployments;

- SLA assurance, for example related to network slices.
- Support of the optimization of networks, services and functionalities;
- Support of the automation of network and service management;
- The management data analytics capability to support B2B services;
- The management data analytics capability exposure and data collection mechanism with the vertical management system (e.g. AF management system);
- Input and output data of the MDAS to realize the items above-mentioned.

Currently (V0.4.0, June. 2020), the content of its document (3GPP TR 28.809), contains the concepts and overview, MDA process and role, use cases, potential requirements and possible solutions.

3GPP SA WG5 started Autonomous Network Levels SI（ANL) in September 2019. The objective is to study the concept definition, typical scenarios and classification of network autonomy levels.

Currently (V1.0.0, June. 2020), the content of its document (3GPP TR 28.810), contains the concepts and overview, potential dimensions for classification of network autonomy, use cases and potential solutions.

In July 2020, the corresponding WI was approved in SA#88 meeting，its objectives are as follows:

- Specify the concept of autonomous network levels, typical management use cases which operators could benefit from network autonomy.
- Specify related workflows and management requirements for each autonomous level for the typical management use cases corresponding to network and service planning & designing, provisioning, maintenance and optimization phases.
- Specify the architecture requirements and enhancements of management and orchestration to support various autonomous network levels if needed.
- Identify the important data and data collection policy for NG-RAN and 5GC (includes NWDAF information) and requirements on management analytics capabilities to support various autonomous network levels.
- Specify the key management requirements on SON features to support various autonomous network levels corresponding to network self-establishment, self-configuration and self- optimization.
- Specify the key management requirements on control loops of SLS assurance to support various autonomous network levels corresponding to service assurance.
- Specify the key requirements on intent driven management to support various autonomous network levels.

- Identify the information exchange within the management systems to support various autonomous network levels.

In September 2019, 3GPP SA WG5 also started a work item on Closed loop SLS Assurance（COSLA）for 5G networks, this project aims to specify a closed loop assurance solution that supports a service provider to continuously deliver the expected level of communication service quality. The closed loop assurance solution allows a service provider to create a closed loop management service that automatically adjusts and optimizes the services provided by NG-RAN and 5GC based on the various performance management and QoE input data, and the state of the 5G network, using data analytics provided by a MDAS.

Currently the WID has finished in June 2020, and a new WID for COSLA enhancement is expected to finish in Release 17.

4.2.4 RAN3

3GPP RAN WG3 started RAN-centric Data Collection and Utilization SI in June 2018. The objective is to study the wireless big data collection and application oriented to network automation and intelligence, including the process and information interaction required for studying different use cases. During the study phase, the work focuses on studying the use cases and benefits of RAN centric Data utilization, e.g. SON features including mobility optimization, RACH optimization, load sharing/balancing related optimization, Minimization of Drive testing (MDT) etc. Based on the evaluations on the use cases, Identify necessary standard impact and the procedure for configuration and collection of UE measurements, L2 gNB/en-gNB measurements and signalling procedure for problem analysis.

In June 2019,the study phase finished and the corresponding WI was approved in RAN#84 meeting which aimed to be completed in February 2020. Detailed discussion on the impact to the spec are ongoing in RAN2 and RAN3, including impact to NGAP.S1AP,x2AP,XnAP,E1AP,F1AP and also the Uu interface. Until now,50% is completed.

The detailed objectives of the SI are listed as follows:

Study high level principles for RAN intelligence enabled by AI, the functional framework (e.g. the AI functionality and the input/output of the component for AI enabled optimization) and identify the benefits of AI enabled NG-RAN through possible use cases e.g. energy saving, load balancing, mobility management, coverage optimization, etc.:

a) Study standardization impacts for the identified use cases including: the data that may be needed by an AI function as input and data that may be produced by an AI function as output, which is interpretable for multi-vendor support.

b) Study standardization impacts on the node or function in current NG-RAN architecture to receive/provide the input/output data.

c) Study standardization impacts on the network interface(s) to convey the input/output data among network nodes or AI functions.

One general objective for the work is that the studies should be focused on the current NG-RAN architecture and interfaces to enable AI support for 5G deployments.

4.3 ETSI ISG ENI

4.3.1 Architecture

ETSI GS ENI 005[38] specifies the functional architecture of an ENI System, which is a high-level decomposition of an ENI System into its major components, along with a characterization of the externally visible behavior (e.g. as defined by a set of reference points) of the components. In the ENI System, intent is treated as a type of policy (i.e. Intent Policy) that uses statements from a controlled language to express the goals of the policy, but not to indicate how to accomplish those goals. Therefore, each statement in an Intent Policy may require the translation of one or more of its terms to a form that another managed functional entity can understand.

The document also discusses the differences between Intent Policy and other types of Policies. The first important difference between processing Intent Policies and other types of Policies is that Intent Policies are written in a Controlled Language (e.g., a dedicated external DSL) and then submitted to the ENI system. The second important difference between processing Intent Policies and other types of Policies is that Intent Policies, once translated, may need to be rewritten. The third important difference between processing Intent Policies and other types of Policies is that Intent Policies may need to be translated to a different level of abstraction.

4.3.2 Intent Aware Network Autonomicity

ETSI GR ENI 008[39] describes the motivation, requirements, and key issues of using intent policies to manage the operation of networks and networked applications in various domains. This document discusses various design

options, in terms of a set of new stand-alone and/or nested Functional Blocks, for using intent within the ENI System Architecture. This includes accepting and validating intent policies, determining how intent affects the goals and operation of the ENI system, and how it is used by business users, application developers, and network administrators.

Specifically, the report identities architecture which is enhanced to process the Intent Policy, including several enhanced Functional Blocks and their connection to the semantic bus. The report also proposes the translation and assurance procedures for processing Intent Policy and lifecycle management of Intent Policy, including intent states and operations for transition between different intent states. Finally, the report identities several use cases related to intent aware network autonomicity, e.g. context-aware VoLTE service experience optimization.

4.3.3 Intent-based user experience optimization

This a PoC project, which aims to demonstrate the use of intent policy in the wireless domain as defined in GS ENI 002 and GR ENI 008. In particular, the PoC aims to verify that when the wireless network state changes, the intent goal of the user can still be satisfied by the ENI system. Automatic closed-loop management and intra-RAN autonomy can be achieved through intent policy translation and maintenance.

This PoC aims to demonstrate the use case [#2-2: Radio Coverage and capacity optimization]. In particular, the proposed mechanism is compliant with its initial context configuration, triggering conditions, operational flow, and post-conditions, as defined in GS ENI 001.Coverage and capacity optimization (CCO) is one of the typical operational tasks of the radio access network (RAN). CCO aims to provide the required capacity in the targeted coverage areas, to minimize the interference and maintain an acceptable quality of service in an autonomous way. To achieve these targets, antenna power and configuration (pilot power, antenna down tilt, antenna azimuth, or massive MIMO pattern in 5G) play a critical role, as they affect the direction of the antenna radiation pattern, therefore can be used to improve the received signal strength in the own cell as well as to reduce the interference to neighboring cells.

This PoC consists of three parts.

- The first scenario demonstrates the standard north-band webservice interface in intent translation function block.
- The second scenario demonstrates the intent translation procedure between Intent Translation Function Block and Knowledge Repository Function Block for intent policy.

- The third scenario demonstrates the intent-execution and intent-maintenance procedure between intent policy fulfillment module inside Policy Management Function Block and Knowledge Repository Function Block for intent policy.

4.4 ONAP

Intent technology was first proposed by China Telecom and Huawei at the beginning of ONAP Guilin release. The proposal was reviewed by ONAP TSC as Intent-Based Network (IBN) POC during Guilin release. Since the establishment and adjustment of the network environment requires expertise in different roles and complex network operations, Intent-Based Network (IBN) aims to bridge the gap between the business department and IT department. The goal of IBN POC is to establish a scalable intent network implementation framework, which can automatically convert the user's natural language network intent into the operation of network devices and interfaces, so as to simplify the construction and adjustment of the environment.

According to the proposal of IBN with semantic analysis capabilities，ONAP could identify users' natural language-based network intents, and translate the intents into network configurations and device / interface operations. With the help of the intent assurance function in IBN, ONAP can be aware of whether the network environment meets the user's intent requirements, and if not, make corresponding intervention adjustment.

5 Summary

Towards the smart wireless network, this whitepaper tries to investigate the data driven and intent aware wireless network. The use cases and potential solutions for data driven wireless network are discussed and updated based on the recent progress and our previous three whitepapers. The ML algorithms utilized in the all the use cases of the series of whitepaper are also discussed and summarized to provide some insight for the researcher on the future works. Leveraging the data driven solutions, to further enhance the network intelligence, simplify the network management complexity and offer simplified management capability towards the vertical industries, the intent aware wireless network is proposed. The evolution, a use case of "mobility load balancing" and enhanced architecture are presented. The key technologies, technical key issues and challenges are discussed as well. The standards progress related to wireless big data and AI as well as the potential standards impacts are also discussed.

As the key guiding documents for future works, the series whitepaper on smart 5G/wireless network are expected to help readers better understand how to bring the WBD and AI/ML into wireless network towards the intelligence wireless ear. The latest achievements will be presented in annual updates of this white paper or

technology review reports. A collective wisdom and especially more efforts among SIG members and others partners are continuously expected.

We sincerely invite all mobile operators, telecom equipment manufacturers and IT system manufacturers, as well as industrial and academic research institutions of interest to join us and work closely to deliver the smart wireless network vision.

6 Reference

[1] FuTURE Forum 5G SIG, Whitepaper, "Wireless Big Data for Smart 5G", Nov. 2017, online available: http://www.future-forum.org/dl/171114/whitepaper2017.rar.

[2] FuTURE Forum 5G SIG, Whitepaper, "Mobile AI for Smart 5G-Empowered by Wireless Big Data", Nov. 2017, online available: http://www.future-forum.org/dl/171114/2018mwc.rar.

[3] FuTURE Forum 5G SIG, Whitepaper, "Wireless Big Data and AI for Smart 5G & Beyond, 2018".

[4] FuTURE Forum 5G SIG, Whitepaper, "Data Driven and Intent Aware Smart Wireless Network, 2019"

[5] Hernogorov F , Nihtila T . QoS Verification for Minimization of Drive Tests in LTE Networks, IEEE Vehicular Technology Conference. IEEE, 2012.

[6] Riaz Mondal, Jussi Turkka, Tapani Ristaniemi. Performance evaluation of MDT assisted LTE RF fingerprint framework, 2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU). IEEE, 2014

[7] M.Ester, H.Kriegel, J.Sander, X.Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", 1996.

[8] T.Liu, "Learning to Rank for Information Retrieval", in Foundations and Trends® in Information Retrieval, 2009.

[9] L.Yang, X.Guo, HM.Wang, and W.Chen, "A Video Popularity Prediction Scheme with Attention-Based LSTM and Feature Embedding", in IEEE GLOBECOM, 2020.

[10] https://api.bilibili.com/x/web-interface/view?aid=91835249

[11] https://api.bilibili.com/x/v2/reply?pn=1&type=1&oid=91835249&sort=2

[12] H. Pinto, J. M. Almeida, and M. A. Gonc¸alves, "Using early view patterns to predict the popularity of youtube videos," in Proceedings of the sixth ACM international conference on Web search and data mining, 2013.

[13] H. Dou, W. X. Zhao, Y. Zhao, D. Dong, J.-R. Wen, and E. Y. Chang, "Predicting the popularity of online content with knowledge-enhanced neural networks," 2018.

[14] S. Yuan, Y. Zhang, J. Tang, H. Shen, and X. Wei, "Modeling and predicting popularity dynamics via deep learning attention mechanism," arXiv preprint arXiv:1811.02117, 2018.

[15] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: Enable industries own software-defined cellular networks," IEEE Commun. Mag., vol. 54, no. 7, pp. 146–153, Jul. 2016.

[16] R. Li et al., “Deep reinforcement learning for resource management in network slicing,” IEEE Access, vol. 6, pp. 74429–74441, Nov. 2018.

[17] R. Li, C. Wang, R. Guo, Z. Zhao, and H. Zhang, “The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility,” IEEE Commun. Lett., vol. 24, no. 9, pp. 2005–2009, Sep. 2020.

[18] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, “GAN-powered deep distributional reinforcement learning for resource management in network slicing,” IEEE J. Sel. Area. Comm., vol. 38, no. 2, pp. 334–349, Feb. 2020.

[19] Q. Shi, Y. Liu, S. Zhang, S. Xu, S. Cao, and V. Lau, “Channel estimation for WiFi prototype systems with super-resolution image recovery,” in Proc. IEEE ICC, 2019.

[20] S. Zhang, Y. Liu, Q. Shi, S. Xu and S. Cao, "LSRN: A Recurrent Residual Learning Framework for Continuous Wireless Channel Estimation Using Super-Resolution Concept," IEEE Access, vol. 8, pp. 38098-38111, 2020.

[21] Q. Chen, S. Zhang, S. Xu and S. Cao, "Efficient MIMO Detection with Imperfect Channel Knowledge - A Deep Learning Approach," in Proc. IEEE WCNC, 2019.

[22] 3GPP TS 38.300 V16.3.0, “NR; NR and NG-RAN Overall Description; Stage 2 (Release 16)”, Sept. 2020.

[23] 3GPP TS 37.320 V16.2.0, “Radio measurement collection for Minimization of Drive Tests (MDT); Overall description; Stage 2 (Release 16)”, Sept. 2020.

[24] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, K.-W. Wen, K. Kim, R. Arora, A. Odgers, L. M. Contreras, and S. Scarpina, “MEC in 5G networks”, in ETSI White Paper, no. 28, pp. 1–28, Jun. 2018.

[25] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob and A. Ahmed, "Edge computing: A survey", in Future Gener. Comput. Syst, vol. 97, pp. 219-235, Aug. 2019.

[26] H. Ye, G. Y. Li and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," in IEEE Wireless Communications Letters, vol. 7, no. 1, pp. 114-117, Feb. 2018.

[27] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, “Exploring generalization in deep learning”, In Advances in neural information processing systems, pp. 5947-5956, 2017.

[28] R. Li, Z. Zhao, X. Xu, F. Ni, and H. Zhang, “The collective advantage for advancing communications and intelligence,” IEEE Wireless Commun., vol. 27, no. 4, pp. 96–102, Aug. 2020.

[29] X. Xu, Z. Zhao, R. Li, and H. Zhang, “Brain-inspired stigmergy learning,” IEEE Access, vol. 7, pp. 54410–54424, Apr. 2019.

[30] K. Chen, R. Li, Z. Zhao, J. Crowcroft, Z. Zhao, and H. Zhang, “DEMO: The Implementation of Stigmergy in Network-assisted Multi-agent System,” presented at the ACM Mobicom 2020 (Demos Session), Los Cabos, Mexico, Sep. 2020.

[31] 3GPP TR36.902, “Self-Configuring and Self-Optimizing Network (SON) Use cases and solutions,” 2011.

[32] I. Viering, M. Dottling and A. Lobinger, “A Mathematical Perspective of Self-Optimizing Wireless Networks,” Proc. IEEE International Conference on Communications, Dresden, Jun. 2009, pp. 1-6.

[33] Z. Gao, C. Chen, Y . Li, B. Wen, L. Huang and Y . Zhao, “A mobility load balancing algorithm based on handover optimization in LTE network,” Proc. 10th International Conference on Computer Science & Education (ICCSE), Cambridge, Jul. 2015, pp. 611-614.

[34] Z. Li, H. Wang, Z. Pan, N. Liu and X. Y ou, “Joint optimization on load balancing and network load in 3GPP LTE multi-cell networks,” Proc. International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, Nov. 2011, pp. 1-5.

[35] L. Pang, C. Yang, D. Chen, Y. Song and M. Guizani, "A Survey on Intent-Driven Networks," IEEE Access, vol. 8, Jan. 2020, pp. 22862-22873.

[36] J.Shen, X.Zhu, L.Jiao, L.Pang, R.Li, Y.Ouyang and C.Yang,"Intent-Driven Mobility Load Balancing," IEEE Communications Letters, 2020.10.26. (Submitted, CL2020-2594)

[37] Y . Zhao, S. Mao, J. O. Neel and J. H. Reed, “Performance Evaluation of Cognitive Radios: Metrics, Utility Functions, and Methodology,” Proceedings of the IEEE, vol. 97, no. 4, Apr. 2009, pp. 642-659.

[38] ETSI GS ENI-005 (V2.0.17): "Experiential Networked Intelligence (ENI); System Architecture".

[39] ETSI GR ENI-008 (V0.0.20): "Experiential Networked Intelligence (ENI); Intent Aware Network Autonomicity ".

Acknowledgement

Grateful thanks to the following contributors for their wonderful work on this whitepaper in alphabetical order:

China Mobile:	Chih-Lin I, Gang Li
China Telecom:	Zhibin Yang, Xu zhao
China Unicom:	Feibi Lv
Datang Mobile:	Ming Ai
Huawei:	Xianming Li
LENOVO:	Xin Guo
Shanghai University:	Shunqing Zhang, Qi Shi
Unisoc:	Xiaoyu Chen
XIDIAN University	Chungang Yang
Zhejiang University:	Rongpeng Li
ZTE:	Dong Chen, Jing Han

Abbreviation

3GPP	the 3rd Generation Partner Project
5G	the Fifth-Generation mobile communications
AI	Artificial intelligence
CSC	Communication Service Consumer
CSP	Communication Service provider
CP	Central Processor
CNN	Convolutional neural network
DNN	Deep Neural Network
DNNrx	DNN-based receiver
DRL	Deep Reinforcement Learning
DQL	Deep Q learning
GNSS	Global Navigation Satellite Systems
GPU	Graphic Processing Unit
KPI	Key Performance Indicator
LSTM	Long Short Term Memory
NPU	Neural Processing Unit
NRM	Network Resource Model
PLS	Partial Least Squares Regression
PRA	Predictive resource allocation
PRA-ICIC	PRA with ICI coordination
QoE	Quality of Experience
RAN	Radio Access Network
RSRP	Reference Signal Receiving Power
RSRQ	Reference Signal Receiving Quality
VOD	Video-on-Demand

未来移动通信论坛
FuTURE MOBILE COMMUNICATION FORUM