Article:
Ma, B., Yang, B., Zhu, Y. et al. (2020) Context-aware proactive 5G load balancing and optimization for urban areas. IEEE Access, 8. pp. 8405-8417.

https://doi.org/10.1109/access.2020.2964562

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 30, 2019, accepted December 30, 2019, date of publication January 7, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964562

Context-Aware Proactive 5G Load Balancing and Optimization for Urban Areas

BO MA[1], BOWEI YANG[2], YUNPENG ZHU[3], AND JIE ZHANG[1]
[1]Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S1 3JD, U.K.
[2]School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China
[3]Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield S1 3JD, U.K.

Corresponding author: Bowei Yang (boweiy@zju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61501399, in part by the European Commission Horizon 2020 Project Data Aware Wireless Network for Internet of Everything under Grant 778305, and in part by the Department of Science and Technology of Guangdong Province under Grant 2018A050501008.

ABSTRACT In the fifth-generation (5G) mobile networks, the traffic is estimated to have a fast-changing and imbalance spatial-temporal distribution. It is challenging for a system-level optimisation to deal with while empirically maintaining quality of service. The 5G load balancing aims to address this problem by transferring the extra traffic from a high-load cell to its neighbouring idle cells. In recent literature, controller and machine learning algorithms are applied to assist the self-optimising and proactive schemes in drawing load balancing decisions. However, these algorithms lack the ability of forecasting upcoming high traffic demands, especially during popular events. This shortage leads to cold-start problems because of reacting to the changes in the heterogeneous dense deployment. Notably, the hotspots corresponding with skew load distribution will result in low convergence speed. To address these problems, this paper contributes to three aspects. Firstly, urban event detection is proposed to forecast the changes in cellular hotspots based on Twitter data for enabling context-awareness. Secondly, a proactive 5G load balancing strategy is simulated considering the prediction of the skewed-distributed hotspots in urban areas. Finally, we optimise this context-aware proactive load balancing strategy by forecasting the best activation time. This paper represents one of the first works to couple the real-world urban event detection with proactive load balancing.

INDEX TERMS Context-aware, data analytics, proactive load balancing, 5G, machine learning.

I. INTRODUCTION

In 5G, the proactive network optimisation boosts the network in disposing the exponential traffic growth (600x to 2500x capacity increase [1]), stringent service requirements (10,000 or more low-rate devices per cell site [2]), and reducing capital and operational expenditure ($\approx 60\times$ expenditure increase [1]). Proactive load balancing is one of the optimisation methods, which can automatically pre-configure the cell margins for fast convergence when the environment changes. Nevertheless, traditional schemes have to satisfy not only the regular-time demands but also the peak-hour demands of users, which can cause network energy efficiency reduction [3]. To fill the gap, the context-aware proactive load balancing will forecast high-resolution traffic demand and quantify the environmental context to drive decision making. In other words, the ultra-dense heterogeneous networks will have the intelligence to generate and utilise the context, such as traffic patterns and user behaviours, to help the network proactively deciding an energy-efficient optimisation scheme [3]. The context is often challenging to uncover, unfold over time, and it is difficult to collect personal data due to privacy concern. Therefore, public online data, such as online event calendars and social networks, are usually applied for research.

For this purpose, this work mainly contributes to three aspects. Firstly, we use a 3-stage data-analytic based on Twitter data to design the context-aware module for forecasting the changes in traffic hotspots during events in urban areas. Secondly, the prediction of hotspots is fed to a proactive load balancing strategy to automatically configure cell margins. Thirdly, we optimise this proactive load balancing strategy through forecasting the earliest activation time with minimising the prediction errors.

The associate editor coordinating the review of this manuscript and approving it for publication was Irfan Ahmed.

Generally, load balancing is required as a dynamic optimisation process to cope with the varying traffic distribution [4]. Specifically, it handovers the UEs at the edge of overlapping or adjacent cells from a congested cell to an idle cell through adjusting handover parameters, such as offset values [5], thresholds [6], and the number of handovers [7]. These adjustments are usually conducted according to the output of the continuous optimisation iterations or the logic controllers. For example, Fuzzy logic controllers were used in [8], [9] for auto-tuning handover margins. These controller-base methods have a cold-start problem because it has insufficient information at the start-up and requires time to converge. More challenges for these controller-based load balancing methods are listed as follows:

- The controller-based algorithms are reactive to the problems, which is time-consuming in adjusting the offset, and limited in adapting the fast-changing load [10].
- The target cells may have a re-overload occurrence because the controllers potentially have an occurrence of oscillations [8].
- The controller-based algorithms trigger all the neighbouring cells to balance the load, which is inefficient for intra-cell skewed UE-distribution [10].

The causes of the above challenges come from two main categories. Firstly, the drawbacks are the nature of controllers. Secondly, traffic is unpredictable. In order to overcome these issues, several proactive load-balancing methods based on machine learning have been proposed to deal with the unpredictable traffic by learning, forecasting, then adjusting cell margins [11]–[14]. Cellular data, such as cell load, call blocking ratio, UE and BS distribution, are usually used to learn prior knowledge for the proactive optimisation. The related works will be reviewed in the next section.

The paper is organised as follows: Section II reviews the related works about proactive load balancing and highlights the gaps for requiring context-awareness. Section III introduces the framework of context-aware proactive load balancing for an urban area. Section IV states a 3-stage data-analytics of generating spatial-temporal traffic pattern and executing event hotspots detection through using Twitter data. Then, Section V proposes a strategy of balancing load based on the production from the 3-stage data-analytics. This section also provides an urban-area simulation example of the context-aware proactive load balancing and its quantification optimisation. Finally, Section VI concludes this paper.

II. RELATED WORKS

Traditional proactive load balancing methods are developed based on machine learning algorithms to forecast the cell load condition in which the profit will be maximised. In detail, the algorithms adjust cell margin (offset) according to the relationship among the traffic load, packet loss ratio, and the offset. Many machine learning methods can be applied for modelling this relationship, such as regression or reinforcement learning. Specifically, these methods decide the offset with maximising reward in a particular load condition with maintaining the packet loss ratio. For example, Q-learning is used in this way to continuously self-tune the Reference Signal Received Power (RSRP) margin [12], [13]. Moreover, the paper [11] proposed a polynomial regression following a similar way to highlight the proper decision. These traditional algorithms can relieve the pressure of slow convergence and potential oscillations. However, they have some common drawbacks because the learning process is only cellular data-driven. For example,

- They lack the knowledge of user behaviours, such as mobility and preference [10].
- The conditions with burst high traffic demands are not considered, such as during popular events. In other words, they are proactive to usual conditions (daily traffic) but reactive to the irregularly burst traffic (during events).
- There exists unnecessary cell expansion for the skewed intra-cell demands distribution [10].

If the mobility of each user is known, the first two gaps can be filled and free up the high-load cell proactively based on the exact locations and their associated distance with the BS [16], [17], but the user tracking in such a detailed level is still challenging. To address the gaps mentioned above, context-aware methods were proposed based on heterogeneous data sources, such as social networks and GPS. The basic idea of the context-awareness is to extract any valuable information from the homogeneous datasets and forecast the demands and behaviours of users. For example, the work [15] forecast the user distribution of potential high-loaded areas and suggested to activate the nearest idle cell to expand margins by considering the interference and the QoS. Another similar work is done in [10].

Nevertheless, two problems need to be considered before the actual implementation of real-world data analytics. The first one is estimating the correlation between social network data and cellular key performance indicators (KPI). The second problem is modelling and forecasting users' behaviours, especially the upcoming events.

The correlation between the social network data and the cellular KPIs can be modelled by linear regression or a statistic correlation estimation. The linear regression fits the observations and gives parameters of the positive correlation. The strength of correlation becomes higher with a lower spatial granularity (resolution), so it is a trade-off to choose a better spatial resolution or higher correlation. The papers [18], [19] worked on correlation modelling. Moreover, the work [20] studied trade-off with different spatial granularity.

The event detection is to retrieve necessary user-behaviour information of a planned public occasion, such as schedule and attendance. The attendees glance at events from online events calendar and social networks in advance, which make the occurrence of events detectable [21], [22]. The general process is to model the daily seasonal trend as regularity and detect the event as outliers. Such outliers are classified into the category of irregularity (anomaly). For example,

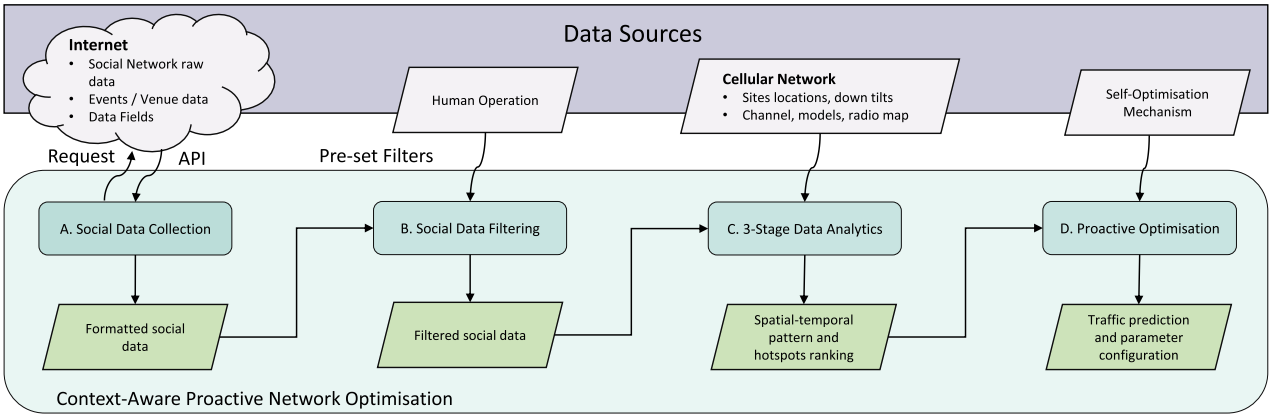


FIGURE 1. The detailed framework of proposed algorithm.

the work [22] proposed a Twitter-based festival detection algorithm using the boxplot. Also, the Twitter-based event detection is introduced in [21], [23], [24].

However, these techniques have not been coupled with the proactive optimisation to enable the network with context-awareness. This work aims to use a 3-stage data-analytics on Twitter data to detect the events in a real-world urban area associated with hotspots and correlate the social network traffic to cellular traffic, which is introduced in detail in the next sections.

III. THE FRAMEWORK OF CONTEXT-AWARE PROACTIVE 5G LOAD BALANCING FOR URBAN AREAS

The framework is visualised in Fig. 1 of integrating social-network data analytics into the proactive load balancing for gaining the desired context-awareness. This framework owns two functional blocks, data sources and context-aware proactive network optimisation. The optimisation block is divided into four minor functions: social data collection, social data filtering, 3-stage data analytics, and proactive optimisation. The first two functions (collection and filtering) are common in the related works, such as in the event-based network managements [24] and the context-aware load sharing [10]. The core contributions of this work are the functions of the 3-stage data analytics and proactive optimisation.

A. SOCIAL DATA COLLECTION

This function is designed to capture raw Tweets from online platforms and pre-process them to produce a formatted dataset.

1) Capture: There are three main ways to capture raw social network data. The first way is crawling through the Application Programming Interface (API), such as the Twitter search and stream API. This method is economical in implementation but facing difficulties in efficiently gaining a complete dataset. The second way is collecting from volunteers to have a complete dataset but time-consuming. The third way is to cooperate with the service providers to obtain the privacy-free datasets. We use the third way to capture 0.6 million geo-tagged Tweets for the Greater London and surrounding suburbs area for two weeks (time resolution in seconds).
2) Parse: The raw data have many fields which need to be parsed into the required fields. In this paper, we select the geolocations, time, and Tweets text as the fields to be analysed.
3) Format: The parsed data fields need to be formatted and stored as the proper file that can be fed to data-analytics programs. We choose to use the Comma-Separated Values (CSV) format which is well supported by Python, Matlab, and Excel.

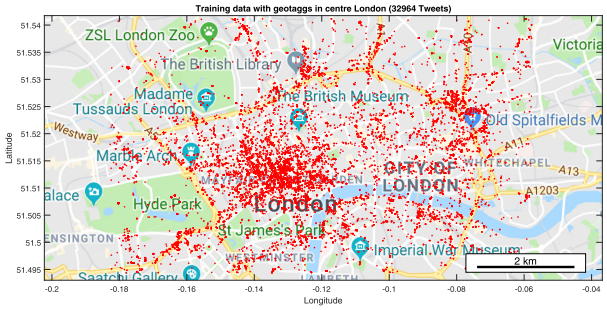
The formatted data are ready-to-use and contain the whole information in the required fields. To pre-process them for particular research objectives, a filter is required.

B. SOCIAL DATA FILTERING

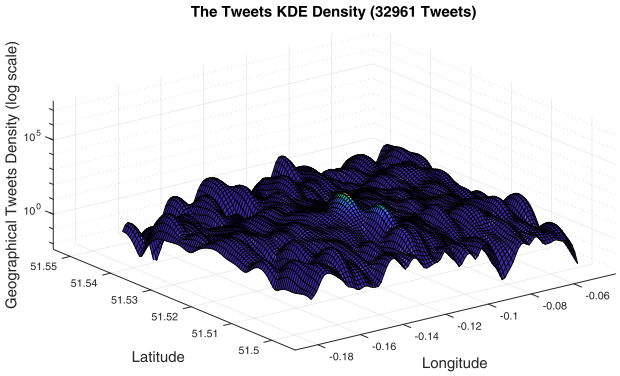
The filter is designed by researchers to reduce irrelevant data from the dataset and further provide a numeric expression of the data in dimensions with interests.

1) Coordinates: This process is to filter the formatted data according to the region of interests. For example, the raw Tweets in this research come from the Greater London and surrounding suburbs, but our interest is in the urban region. Therefore, the coordinates (bottom left: [51.494417, −0.182733], top right: [51.541160,−0.057710]) are selected to filtered the Tweets in the region for research. The Tweets locations in the first week are visualised on the map in Fig. 2a. It shows that the spatial distribution varies with different density.
2) Numeric: A numeric expression is to statistically count the density of data in different dimensions, such as using the histogram to describe the discrete Tweets density on the map. Fig 2b provides a continuous density map using Kernel Density Estimation (KDE). The density map shows varying density with 'peak' and 'valley'. In detail, the 'peak' indicates a high-traffic region with the 'valley' as the boundary.

After generating the numeric expressions, the pre-process is finished. The data are ready to be fed to the 3-stage data-analytics.



(a) Geo-tagged Tweets plotted on map



(b) Tweets density described by KDE

FIGURE 2. The Tweets locations (15/02/2016-21/02/2016) are plotted on map and their density is described by Kernel Density Estimation (KDE). The map has corner coordinates (bottom left: [51.494417, −0.182733], top right: [51.541160,−0.057710]).

C. 3-STAGE DATA ANALYTICS

This function analyses the Twitter data by machine learning and statistic methods, such as using unsupervised learning and box-plot. They are used to build a spatial-temporal traffic pattern in state 1 and 2. The final stage is detecting the hotspots changes in anomaly (events) based on the modelled traffic pattern. (see Section IV for the detailed procedures)

1) Stage 1 - Spatial Traffic Pattern: As stated in [3], the traffic variations usually followed the regular behaviours of users day by day, so it became the access point for modelling traffic distribution and variation rules which will allow a better prediction of where diverse traffic volumes are requested by users. For example in Fig 2b, each 'peak' is a high-traffic region with the 'valley' as the boundary. It is naturally to partition the region into several traffic-based regions and model the traffic in different Region of Interest (RoI).
2) Stage 2 - Hotspots: In each RoI, the traffic distribution is also skewed. Some spots have high traffic demands, so they are named traffic hotspots. The histogram is an efficient way to find the hotspots in each RoI. However, the distribution of hotspots is not constant. The anomaly conditions also exist, such as during events.
3) Stage 3 -Anomaly detection: This stage is to detect the time and location of the irregularities (anomaly conditions) to provide a new ranking of the hotspots in the RoI with an event happening. The popular public events can influence the spatial traffic pattern with new temporal-spatial hotspots emerging [10] resulting in a dramatic rise of dropped calls [24].

Such forecasting of new hotspots can be an alert that the cells associated with the hotspots require optimisation. The cell margins are estimated to be proactively set according to the predicted cell load.

D. PROACTIVE OPTIMISATION

This function provides a framework of coupling the predicted hotspots with the network model to forecast network performance following the fuzzy rules of proactive load balancing. An optimisation of the proactive load balancing strategy is provided to estimate the best activation time. (see Section V for the detailed procedures and an example of implementation)

1) Irregularity Check: Anomaly detection may cause wrong alerts because of scarce data or other reasons. The network requires a mechanism for checking if the network indeed operates in the way of predictions. In that case, the system will continue monitoring the network performance and reserving the choice to switch back to traditional optimisation. When the check is passed, a network model is required to forecast network performance.
2) Network Model: This model simulates the network operation according to the context-awareness. It is needed to quantify the profit and the cost of future optimisation. Proactive decisions are made if the profit is acceptable.
3) Decision Maker: This function aims to associate the hotspots with cellular network optimisation. The cells adjust their margins to prepare for the upcoming traffic. Finally, such a simulated performance provides a suggestion if the current parameter configuration is beneficial. The profit also depends on the activation time.
4) Activation Time: The proactive optimisation requires to decide the best time to be activated. It is better to start the optimisation earlier, but more errors will occur. Therefore, a design for balancing the trade-off is required.

The description of the framework ends here. The next section starts the detailed explanation of the 3-stage data-analytics for forecasting the events hotspots.

IV. 3-STAGE DATA-ANALYTICS FOR TRAFFIC PATTERN AND EVENT HOTSPOTS DETECTION

On the one hand, the traffic pattern includes regular user demands distribution which owns natural convenience for prediction. On the other hand, the anomaly in traffic pattern

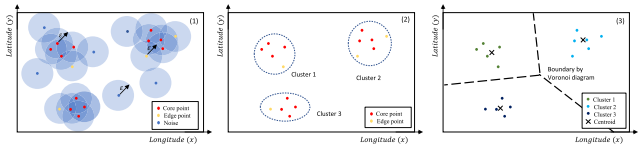


FIGURE 3. The process of using DBSCAN to partition the region into clusters according Tweets density.

(e.g., traffic burst during popular events) also exists so that the system needs to alert the changes in advance. This section proposes the process of generating traffic pattern and the forecasting of hotspots' changes.

A. STAGE 1: SPATIAL TRAFFIC PATTERN

The RoIs are used to divide the urban area into small regions. Each region has a high-density 'peak' which represents the aggregation of demands in the urban area. To configure the RoIs, we select an unsupervised learning algorithm, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [25]. This clustering algorithm groups together the data at 'peaks' (with many nearby neighbours) and marks the data in 'valley' (with nearest neighbours far away) as noise.

We present a brief process of DBSCAN in Fig. 3 explained as the following steps:

1) In the first step, each point has a range with a radius ε. In the ε radius circle, the core points in clusters own neighbours $\geq$ *minPoints* (which is 2 in this example). The edge points hold neighbours but $<$ *minPoints*. Also, the noise has no neighbour in the circle.
2) Then, the DBSCAN ignores the noise and clusters the data points into different groups (dash line circles).
3) Finally, we calculate the centroid which is the mean of all the locations of points in the cluster. Moreover, the Voronoi diagram visualises the boundaries of different clusters.

The above data analytic is tested with the London Twitter data. We denote the 'peaks' as the areas with Tweets densities that are larger than the average density. The map is approximately a rectangle with ignoring the Earth curvature. In that case, the average density is the total number of Tweets over the map area, so the $minPoints = 5$ and $\varepsilon = 4.90 \times 10^{-4}$. Under this circumstance, the result of DBSCAN is shown in Fig 4. It divided the urban area into 541 clusters (RoIs) according to Tweets density. The dense cells usually located at commercial areas and tourist attractions, such as Westminster and the British Museum.

B. STAGE 2: HOTSPOTS

In each RoI, the Tweets have a skewed distribution which will generate some hotspots. The histogram counts Tweets in the pixels of each RoI and products a heatmap with highlighting the hotspots. For example, in the RoI consisting of the Leicester Square and Trafalgar Square (RoI 13 in the spatial traffic pattern as shown in Fig 5), the pixel of Trafalgar Square (yellow hotspot) is more crowded than its neighbours.

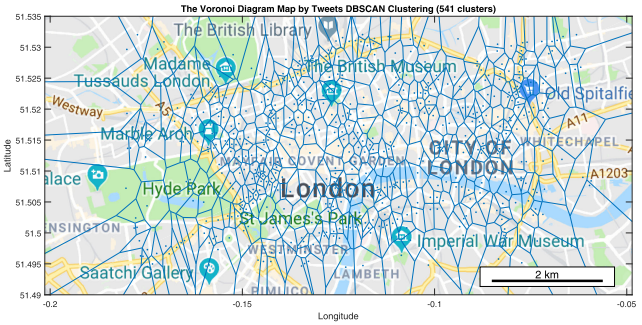
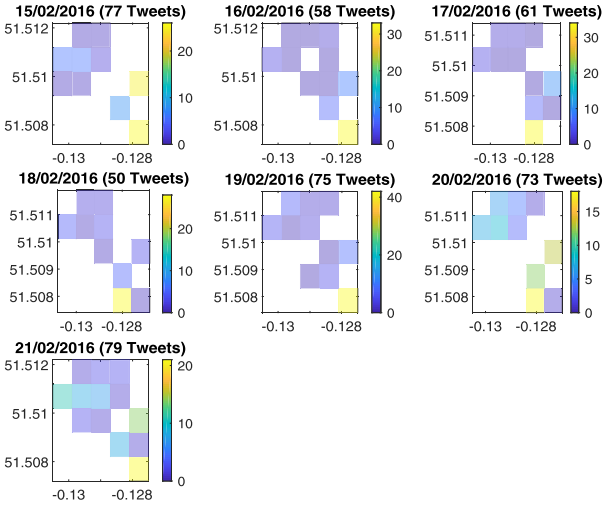
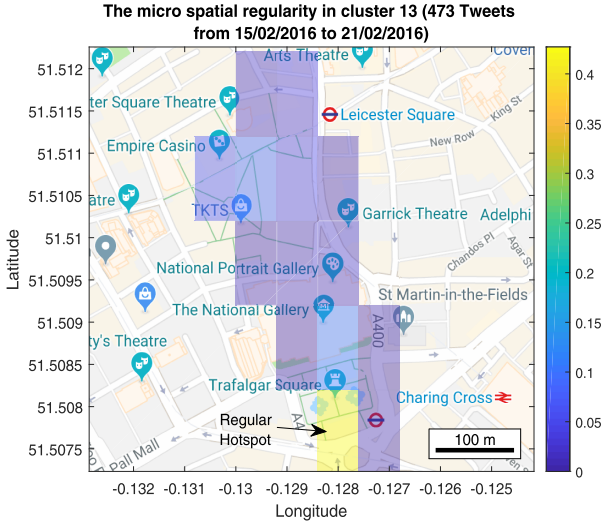


FIGURE 4. The Tweets density-based clusters on map. Corners coordinates (bottom left: [51.494417, −0.182733], top right: [51.541160,−0.057710]).



(a) The histogram of Tweets of 7 days in one week (daily pattern).



(b) The hotspots in one week (weekly pattern).

FIGURE 5. The usual aggregation of users in each day generates some hotspots that users like to stay and use the network. For example, in the region of Trafalgar Square and Leicester Square, the first one attracts more people. It is indicated by the higher number of Tweets in the histogram.

The Fig. 5a displays the daily from 15/02/2016 to 21/02/2016. It shows that the hotspots are commonly distributed around

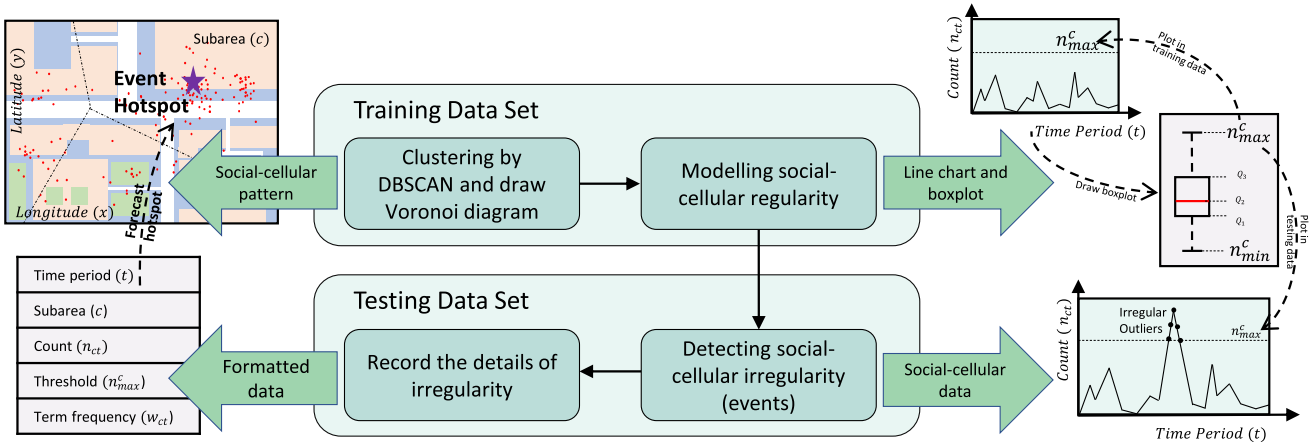


FIGURE 6. The process of building regularity and detecting network anomaly (irregularity).

a coordinate (−0.128, 51.508) in the seven days, and the hotspots point to Trafalgar Square. This phenomenon becomes more evident in the weekly hotspot pattern (Fig. 5b). In consequence, the pixel becomes hotspot if it is regularly hot in an extended period.

However, the hotspots will change along with the occurrence of social events because the users will aggregate at a new location. The network self-optimisation requires a long time to converge for balancing the changed distribution of load. To alleviate this problem, the event detection (or the network irregularity/anomaly detection) is required to pre-configure the network before the changes of the hotspots.

C. STAGE 3: NETWORK ANOMALY DETECTION

There are two main steps of anomaly detection:

- Model the regularity (trend and seasonality) of the training data set.
- Detect the anomaly by finding the outliers in the modelled regularity.

In this work, we use the temporal traffic to model the regularity and detect the outliers. Fig. 6 illustrates the process. The detailed steps are shown as follow:

1) Firstly, the spatial traffic pattern has used the DBSCAN to divide the region into several RoIs. We denote the number of Tweets $\mathcal{N} \subset \mathbb{R}$ in the region $\mathcal{K} \subset \mathbb{R}^2$, then let $n \in \mathcal{N}$ and the clusters (RoI) $c \in \mathcal{K}$, so the number of Tweets in a RoI c in different time intervals $t \subset \mathbb{Z}+$ is n_{ct} (Tweets per hour).
2) Then, for each RoI, a line chart of temporally changing traffic and its corresponding box-plot can be generated. Each box-plot offers a box of the majority and a maximum threshold (n^c_{max}). Such a threshold describes the regular traffic range that regular social network traffic is lower than it.
3) Finally, in the testing data set, if current Tweets per hour (n_{ct}) grows higher than the threshold (n^c_{max}). This hour will be regarded as the start of an event in this RoI. Then the algorithm automatically highlights the irregular outliers and records the details of these outliers, including the time period t, subarea (RoI) (c), count (n_{ct}), threshold (n^c_{max}), and term frequency (w_{ct}). The term frequency refers to the five most-appeared key works in Tweets.

The number of Tweets n_{ct} is an indicator of the network traffic. As indicated in [19], the estimated Down-Link (DL) traffic load $\hat{r}_{DL}$ in cluster c in time interval t can be described as

$$\hat{r}^{ct}_{DL} = 10^{b_{DL}}\left(\frac{n_{ct}}{\tau}\right)^{a_{DL}}\tau \tag{1}$$

where [$a_{\mathrm{DL}} = 0.88\ kb/Tweet \quad b_{\mathrm{DL}} = 2.37\ kbps$] and τ is ratio between time interval and one second (e.g, in this work $\tau = 3600\ s/hour$). Note that, $\hat{r}_{DL}$ is service-neutral that it represents the generated traffic from all services. According to this correlation, the burst of Tweets' traffic during an event represents an irregular increase of network traffic. In this paper, n_{ct} is selected as the indicator of network traffic changes and the occurrence of events.

We collected the advertisements of four events in London from the online event calendar as shown in Table 1. The information from online event calendars has been published at least four days before the events happen. We match the published event location with the spatial traffic pattern to get which RoI the event belongs. Then, the 3-stage data-analytics method is applied to the London dataset to detect the network anomaly.

The results of anomaly detection are shown in Table 2. It displayed the irregular time when the Tweets per hour were higher than the cluster threshold. For example, the first event 'Grimsby World Premiere' caused network traffic irregularity from 18:00 until 23:00, and the traffic peaks arrived at 19:00 (16 Tweets/hour) and 21:00 (14 Tweets/hour). The third event 'Craft Beer Rising' event attracted high traffic demand from 13:00 to 23:00 on 26/02/2016. We find that the start time of Tweets irregularity does not precisely match the start time on the event calendar. It is because users have different arriving time and network usage behaviours. As the number of Tweets

TABLE 1. The details of events form the online calendars. We allocate the nearest cluster to each event location according to the distance to each centroid.

Event Topic	Day	Start Time	End Time	Location	Cluster	Cluster centroid	Event Published
Grimsby World Premiere	22/02/2016	17:00	21:00	Odeon Leicester Square, London	13	(51.50872, -0.12812)	18/02/2016
Curtain Up Exhibition	09/02-31/08/2016	10:00	17:45	Victoria and Victoria and Museum	128	(51.49669, -0.17228)	09/02/2016
Craft Beer Rising 2016	26-27/02/2016	12:00	00:30	Old Truman Brewery in Brick Lane	12	(51.52098, -0.07195)	23/02/2016
Stop Trident Demo	27/02/2016	12:00	-	Trafalgar Square, London	13	(51.50872, -0.12812)	08/12/2016

TABLE 2. The results of event (irregularity) detection.

Grimsby World Premiere

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms, e.g., [('term 1', frequency), ('term 2', frequency). . .]
22	18	13	9	7	[('Square', 4), ('Leicester', 3), ('#Grimsby', 2), ('premiere', 2), ('#GrimsbyWorldPremiere', 2)]
22	19	13	16	7	[('Square', 7), ('Odeon', 7), ('Leicester', 6), ('de', 2), ('amp', 2)]
22	20	13	8	7	[('Leicester', 3), ('London', 3), ('#elvispresley', 2), ('Square', 2), ('Picadilly', 2)]
22	21	13	14	7	[('#London', 10), ('#GrimsbyWorldPremiere', 9), ('#redcarpet', 8), ('#igerspinoy', 5), ('#litratongpinoy', 5)]
22	23	13	13	7	[('#London', 10), ('#GrimsbyWorldPremiere', 10), ('#redcarpet', 10), ('#igerslondon', 8), ('#igerspinoy', 7)]

Curtain Up Exhibition

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms [('term 1', frequency), ('term 2', frequency). . .]
25	21	128	14	7	[('CURTAIN', 11), ('UP', 11), ('OF', 11), ('exhibited', 10), ('CELEBRATING', 9)]
25	22	128	12	7	[('Victoria', 20), ('Albert', 20), ('Museum', 20), ('U', 10), ('K', 10)]
27	16	128	10	7	[('Victoria', 9), ('Albert', 8), ('Museum', 8), ('museo', 2), ('manzana', 2)]

Craft Beer Rising 2016

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms [('term 1', frequency), ('term 2', frequency). . .]
26	13	12	17	14	[('Drinking', 8), ('Craft', 8), ('Beer', 8), ('Rising', 8), ('#photo', 4)]
26	14	12	18	14	[('Drinking', 11), ('Craft', 9), ('Beer', 8), ('Rising', 8), ('@trumanbrewery', 4)]
26	15	12	16	14	[('Drinking', 12), ('Craft', 9), ('Beer', 9), ('Rising', 9), ('Ale', 4)]
26	16	12	18	14	[('Drinking', 12), ('Beer', 12), ('Craft', 10), ('Rising', 10), ('@trumanbrewery', 3)]
26	18	12	17	14	[('Craft', 5), ('Beer', 5), ('Rising', 5), ('#photo', 4), ('Drinking', 4)]
26	19	12	20	14	[('Drinking', 10), ('@trumanbrewery', 8), ('Rising', 6), ('Craft', 5), ('Beer', 5)]
26	22	12	20	14	[('Drinking', 14), ('Craft', 12), ('Beer', 12), ('Rising', 12), ('#photo', 12)]
26	23	12	18	14	[('Drinking', 10), ('London', 7), ('Beer', 7), ('Craft', 6), ('Rising', 6)]
27	12	12	25	14	[('Drinking', 16), ('Beer', 15), ('Craft', 14), ('Rising', 14), ('London', 9)]
27	13	12	24	14	[('Beer', 18), ('Craft', 17), ('Rising', 16), ('Drinking', 14), ('London', 6)]
27	14	12	18	14	[('Drinking', 17), ('Craft', 11), ('Beer', 11), ('Rising', 11), ('@trumanbrewery', 5)]

Stop Trident Demo

Day	Hour	Cluster	Tweets/hour	Threshold	Top 5 terms [('term 1', frequency), ('term 2', frequency). . .]
27	14	13	8	7	[('Square', 4), ('National', 3), ('Gallery', 3), ('Trafalgar', 3), ('3', 2)]
27	15	13	9	7	[('Trafalgar', 5), ('Square', 5), ('London', 3), ('#stoptrident', 2), ("I'm", 1)]
27	16	13	14	7	[('Square', 6), ('Trafalgar', 4), ('think', 3), ("I'm", 2), ('London', 2)]

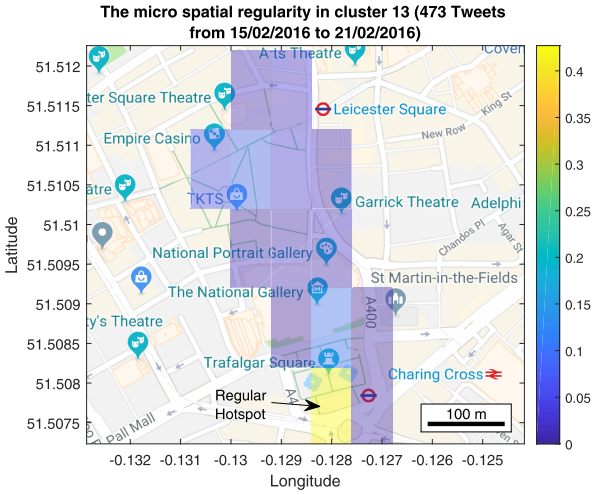
is positively correlated to cellular traffic, the Tweets irregularity becomes the alarm for upcoming high traffic demands. Furthermore, the detected most-frequently used words match the topic keywords in the event calendar.

According to the results of detection, we can forecast the changes of hotspots by highlighting the event location as the new high-traffic region. For example, Fig. 7 (a) and (c) present the hotspots of regularity and on the 'irregular' day of the first event 'Grimsby World Premiere'. The traffic ranking of the pixels has changed and the hottest pixel altered from the Trafalgar Square to the Leicester Square. In our algorithm, once the outlier is detected, the event-detection algorithm automatically allocates the event pixel to be the first position in the hotspots ranking (as visualised in Fig. 7 (b)). This forecasting pattern alerts the network of upcoming high traffic demand at least one hour ahead. Besides, the traffic pattern changed back to regular conditions after the end of the 'World Premiere' (Fig. 7 (d)).

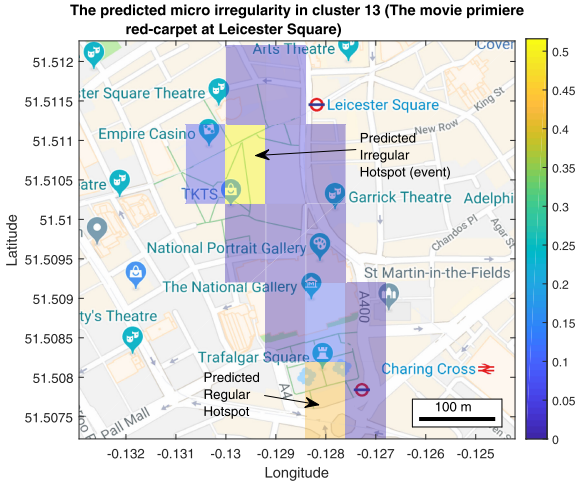
The 3-stage data-analytics also determine the relationship between cells and hotspots according to the hotspots locations and ranks. For example, the hotspots can be in the high-load cell or on the cell edge. We need the fuzzy rules to decide the strategy for each condition. Therefore, we summarise the corresponding relations as follows:

- The cell is predicted to be high-load with event hotspot. (output 1)
- The cell is predicted to be the nearest neighbouring cell to the event hotspot. (output 1)
- The cell is predicted to be the neighbouring cell to the event hotspot but not the nearest one. (output 0)
- Other conditions. (output -1)

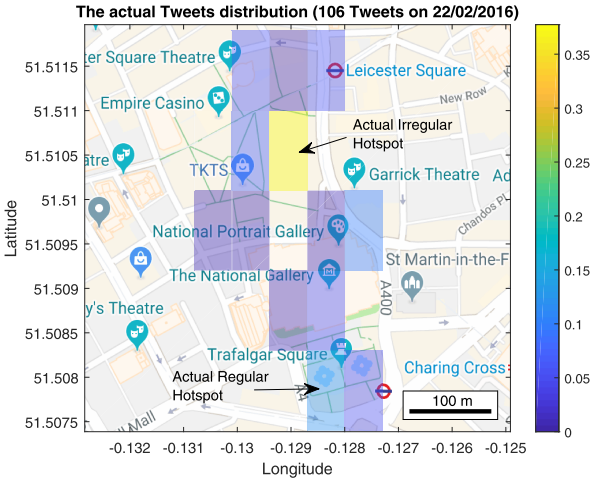
In the first and second conditions, the event hotspots influence the cells most so that they are the cells to be optimised (output 1). The output 0 indicates that the cells are still close to the event hotspot but not directly involved. Moreover, the output 0 denotes that the cells become far from the event hotspot. The next section will provide the strategy of associating proactive optimisation focusing on the predicted hotspots.



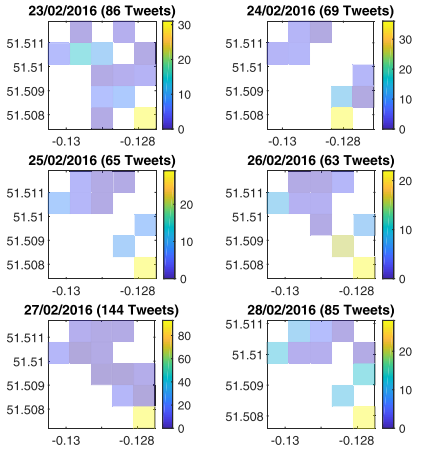
(a) The hotspots in the training data (15/02-21/02).



(b) Predicted hotspots locations and ranking before event.



(c) The actual hotspots on the event day (22/02/2016).



(d) The hotspots recover to regularity after the event (23/02-28/02).

FIGURE 7. The actual and predicted traffic patterns before and after the event. We transfer the Tweets distribution in cluster 13 into a histogram that describes the Tweets occurrence in each pixel. The pixel with more Tweets is regarded as a hotspot. The regular hotspot is usually attractive for users. In contrast, the irregular hotspot brings a sudden burst of Tweets and disappears after the event.

V. PROACTIVE OPTIMISATION WITH CONTEXT-AWARENESS: LOAD BALANCING USE CASE

A. OPTIMISATION FRAMEWORK

The framework in Fig. 8 optimises network configurations based on the forecasting of hotspots from the 3-stage data-analytics. It consists of four major functions: irregularity check, network model, decision maker, and activation time. The following parts will introduce them in detail.

1) IRREGULARITY CHECK

The irregularity check function takes the responsibility of checking and avoiding the following conditions:

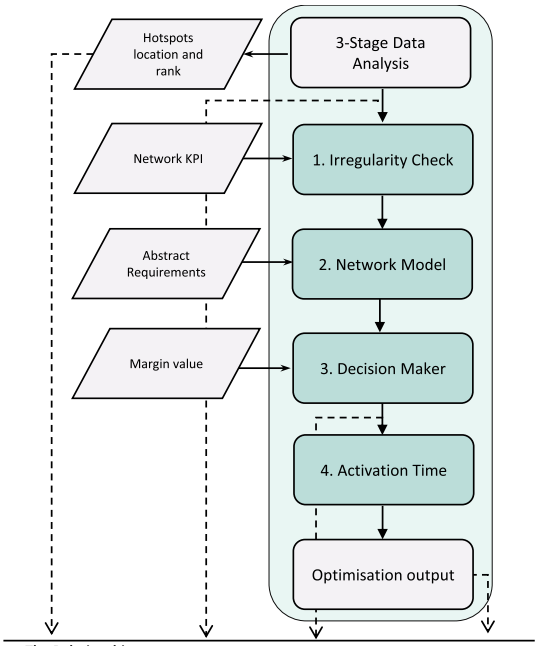
- Errors of events detection.
- The network irregularity lasts for a short duration.
- The detected events do not come with the expected high traffic increase.

In that case, it needs two checking items. Firstly, the web-keywords check will compare the term frequency with the topic keywords on the event calendar. Secondly, for the involved BS itself, the cell traffic should be estimated and justified that the cell indeed has a load increase caused by the events. Accordingly, this function aims to minimise the probability of wrong alarms.

2) NETWORK MODEL

The network model is required to forecast the network performance according to the forecasting of network context. As the context comes from real-world data-analytics, the network model should satisfy the abstract requirements of the actual network settings. For example, the filtered area in London is an urban area, so the small cell density (dense deployment) is 80 $cells/km^2$ [3]. There are several requirements to abstract the network model from the dataset:

- Urban scenario: Different scenarios decide the parameter setting of networks, such as the BS density and frequency band. The dataset in this work is from London, an urban area, so we consider an ultra-dense



The Relationships between Cell and Hotspots	Analyser Output	Decision Maker Output	Optimisation Detail (example)
If the cell is predicted to be with hotspot in or the nearest neighbouring cell	1	Apply new margin value to handover some active UEs between the two BSs	The high-load cell connects UE if >150 kb/s, handoff UE if <100 kb/s
If not, the cells are neighbours of a high-load cell	0	No variation is applied to current margin	No variation
Otherwise	-1	Cancel current margin value and return to original	Connect if > 50kb/s

FIGURE 8. The framework of proactively making decisions of load balancing based on the urban-area anomaly detection and forecasting the hotspots' changes.

heterogeneous network model (downlink) in [3] associated with orthogonal frequency division multiple access (OFDMA). The network consists $\mathcal{B}$ BSs and N subcarriers, $i \in \mathcal{B}$ $n \in N$, and the region served by all BSs is $\mathcal{K} \subset \mathbb{R}^2$, the cell area $x \in \mathcal{K}$.

- Time scale: The time scale varies in different scenarios from seconds to hours, so it is required to determine a proper scale to suit both 3-stage data-analytics and network optimisation. In the simulation, there does not exist an exact time scale to describe the fluctuation of traffic, so a discrete time normalisation is used. In this study, the time is normalised to integer units, and we perform static simulation of network performance at different time snapshots $t \in \mathbb{Z}^+$ and compare the performance between the conditions with and without context-awareness. Therefore, the transmit power and channel gain of BS i to UE in x using channel n at time t are $P_i^n(x,t)$ and $g_i^n(x,t)$.
- Channel model: Such a mathematical representation describes the effect that wireless signals propagate through a wireless channel. It is an essential element in modelling the network. In this study, the channel model does not influence the comparison between with and without the context-awareness. Therefore, the path loss channel model is free-space for computational convenience, so the channel gain $g_i^n(x,t) = G_t G_r (\frac{c}{4\pi d_i(x,t) f})^2$, where $[G_t \quad G_r]$ are values of transmit and receive antenna gain, $[c \quad f]$ are the speed of light and frequency, and $d_i(x,t)$ is the distance between UE and BS i at time t.
- UE association: The load balancing requires to change the UE association with different BSs for transferring the extra load to idle cells. In this work, we bring channel association variable $\rho_i^n(x,t)$ and BS association variable $u_i^n(x,t)$ in the model, if the value is 1 which means the UE is associated with channel n or BS i at time t. The power of noise is $\sigma^2(x,t)$.

Based on the above parameter settings, the signal-to-interference-plus-noise ratio (SINR):

$$\gamma_i^n(x,t) = \frac{P_i^n(x,t) g_i^n(x,t) \rho_i^n(x,t)}{\sum_{j \in \mathcal{B}, j \neq i} \int_{\mathcal{K}} P_j^n(y,t) g_j^n(y,t) dy + \sigma^2(x,t)} \tag{2}$$

Then, according to Shannon's theory, the transmit rate on subscriber n is $R_i^n(x,t) = W log_2(1 + \gamma_i^n(x,t))$, where W is the channel bandwidth. And the average data rate for all the UEs in BS i is

$$\bar{R}_i(x,t) = \frac{\sum_{n=1}^{N} R_i^n(x,t) u_i^n(x,t)}{\sum_{n=1}^{N} u_i^n(x,t)}, \quad \forall i, x, t \tag{3}$$

The goal of load balancing is to handoff extra UEs (with worse data rate) to their nearest neighbour BS. In other words, this process is changing current BS association variable $u_i^n(x,t) = 1 \rightarrow 0$ and the nearest neighbour BS association $u_j^n(y,t) = 0 \rightarrow 1$. In that case, the $u_i^n(x,t)$ value determination is the next aim.

As the edge UEs naturally have higher interference (lower SINR) and occupy channels, they are the objects to be unloaded in the predicted high-load cell (BS i). To do this, we consider setting SINR margin $Margin(i,j)$ to preserve the UEs in BS i with better SINR when $\gamma_i^n(x,t) > Margin(i,j)$, and handoff the UEs with worse performance to the nearest neighbour BS j if $\gamma_i^n(x,t) < Margin(j,i)$. Moreover, the margins should be subjected to $Margin(i,j) > Margin(j,i)$ for avoiding ping-pong effect.

The above network model can simulate the network performance but require to decide the strategy of load balancing according to the context-awareness. The next part introduces this process.

3) DECISION MAKER

This function follows fuzzy rules which are also used in [10]. According to the 3-stage data-analytics, the decision maker will decide the load balancing strategy (see Fig. 8 for the table of fuzzy rules):

- 1: Apply new margin value to handover some active UEs from the high-load cell to the idle cell nearest to the event hotspot.
- 0: No variation will be applied to the current margin.
- -1: Cancel the current margin.

The simulation based on the above strategy results in the average UE data rate $\bar{R}_i(x, t)$ with and without load balancing, and visualises the advantages when proactively trigger the optimisation (at t_1). Moreover, the passive load balancing starts at t_2, $t_1 < t_2$ because the proactive algorithm predicts the peak of traffic while the passive algorithm reacts to it. Therefore, it is important to optimise the strategy to determine an activation time t_1 with maximum profit.

4) ACTIVATION TIME

The activation time of proactive optimisation influences the profit improvement because the forecasting will have more errors if the optimisation needs to be activated earlier. However, if the network is not configured earlier for the upcoming traffic changes, it will need some time to recover and have a poor performance period like in [10]. To balance the trade-off between the prediction errors and the poor performance period, a mechanism needs to be designed for particular scenarios. We provide an example in the next part.

B. AN EXAMPLE OF PROACTIVE LOAD BALANCING IN THE LONDON URBAN SCENARIO

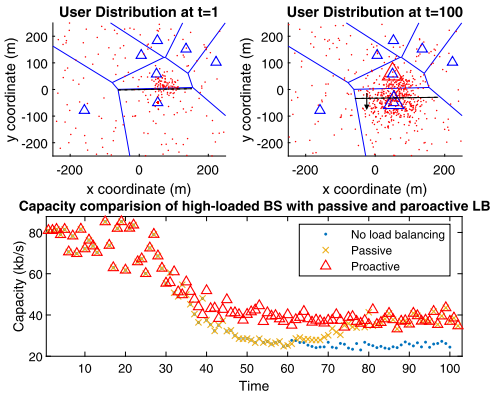
In the simulation, there is a 500 $m \times 500$ m square covered by $\mathcal{B}$ cells generated through Poisson Point Process. The BSs have a medium-density deployment $\lambda = 40$ $cells/km^2$ in an area with radius $R = 250$ m. We assume that one event happens in a cell (which will have a high load in future) located close to the centre of the area. The event will bring a growing hotspot with the number of users increasing from t_1 (10% low load) to t_2 (high load) as a Sigmoid function in an area with a 100 m radius. Therefore, the number of UEs in the event hotspot at $t_1 = 10$ is 18 ($10\% \times 6000\ UE/km^2 \times 0.01\ km^2 \times \pi$), and at $t_2 = 70$ it grows to 377 ($12000\ UE/km^2 \times 0.01\ km^2 \times \pi$). The event hotspot is generated according to a normal distribution with a mean value $\mu = 0.01$ and a standard deviation $\sigma = 0.2$. There is also a regular hotspot (e.g., commercial area) close to the event hotspot. This hotspot has a 50 m radius and a medium load of UE density, so it has 78 UEs generated from another normal distribution ($\mu = 0.01, \sigma = 0.1$). The detailed simulation parameters of communication environment setting (such as SINR, capacity, power, interference, and noise et al.) are shown in Table 3.

We perform a Monte Carlo simulation through repeating random sampling for 100 times. All users are assumed to have equal demands, and the channel allocation can satisfy them equally with allocating proper resource blocks. In that case, when the BSs own equivalent resource, the associated users with fewer competitors will experience owning more resources and better networks. Moreover, both the context-aware and traditional load balancing schemes use the same controllers to ensure that the convergence ability is the same.

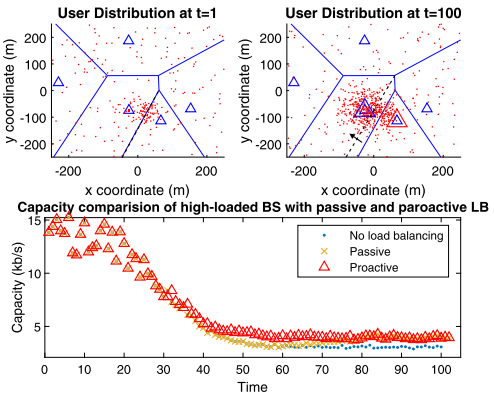
TABLE 3. Simulation Parameters.

Parameter Name	Value
Simulation Environment	Urban Microcell
BS Transmit Power	41 dBm
Frequency Carrier	5 GHz
Channel Bandwidth	5 MHz
Resource Block Bandwidth	180 kHz
Thermal Noise Level	-174 dBm/Hz
UE Antenna Gain	0 dBi
Small Cell Antenna Gain	17 dBi
Simulation Area Radius	250 m
Small Cell Density Dense Deployment	80 cells/km^2
Small Cell Density Medium Deployment	40 cells/km^2
Small Cell Density Sparse Deployment	20 cells/km^2
UE Density Hotspot High Load	12000 UEs/km^2
UE Density Hotspot Medium Load	10000 UEs/km^2
UE Density Hotspot Low Load	6000 UEs/km^2
UE Density Non-Hotspot	800 UEs/km^2
Event Hotspot Area Radius	100 m
Event Hotspot UE Number	18(10% Low) - 377 (High)
Regular Hotspot Area Radius	50 m
Regular Hotspot UE Number	78 (Medium Load)
Scattered UE Number	157 (Non-Hotspot)
UE Start/End Increasing Time	10-70
Time of Proactive Optimization Starts*	30
Time of Passive Optimization Starts*	60
Simulation Loops	100
Handoff Threshold*	50kb/s

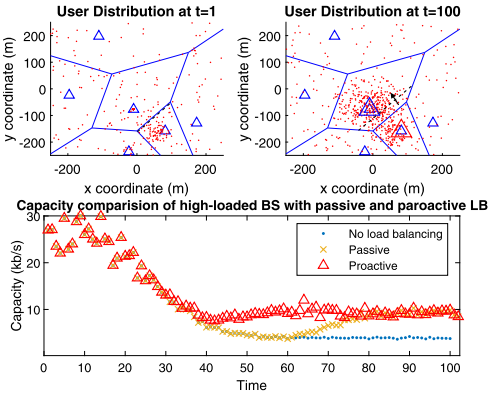
The Fig. 9a,9b,9c proposed the cell layout (blue triangles are BSs) with user distributions (red dots) at the beginning time ($t = 1$) as well as the end time ($t = 100$). The cell margins are visualised through using the Voronoi diagram. They shrank in the event cells (nested blue triangles) and moved as shown as dash lines. The corresponding network performances are displayed (in the bottom line charts separately) as average capacity (data rate) changing along with time. For the first example, the network performance drops to around 25 kb/s at $t = 50$, and it stays in the condition with poor performance if there is no load balancing algorithm (blue dash line). The yellow line indicates the condition with load balancing, but it is reactive to the event (cold-start with no prior knowledge about traffic increase). We denote it as a passive (without context-awareness) load balancing because they are not able to absorb the burst of traffic. It starts at $t = 60$ and finally converges to around 40 kb/s but causes a poor-performance period from $t = 40$ to $t = 80$. That provides an evaluation of the negative effect of the cold-start problem. In contrast, with forecasting upcoming events traffic, the proactive load balancing (red triangle line) experience almost no poor-performance as it starts at $t = 30$ (before the traffic peak arrives). After the Monte Carlo simulation, Fig. 9d displays the average capacity. The network indeed experiences low capacity from $t = 40$ to $t = 80$, which is combined with the time of reaction and the time of convergence. Such a cold start problem is not avoidable unless it



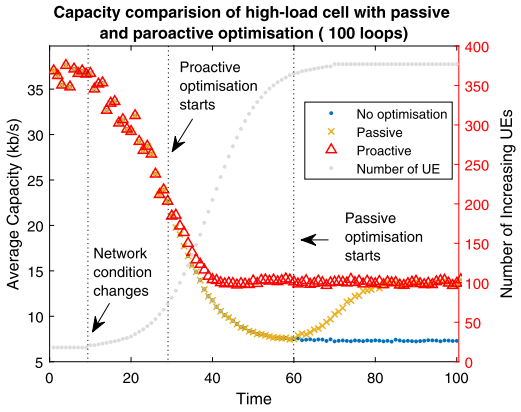
(a) Example 1: Regular and event hotspots are in the same cell.



(b) Example 2: Regular hotspot is on edge.



(c) Example 3: Regular hotspot is in the neighbour cell.



(d) Capacity comparison after the Monte Carlo simulation.

FIGURE 9. **The layouts and results of capacity comparison between proactive (with context-awareness) and passive (without context-awareness) optimisation. There are three random examples in the 100 loops.**

is benefited by context-awareness. In the above simulation, the activation time of the proactive optimisation is assumed according to our experience. The next problem is to optimise the strategy in this area by automatically determining the best start (activation) time.

The activation time of load balancing determines the width of the poor-performance period which looks like a 'pit' dug by the burst traffic (see the yellow crosses lower than red triangles in Fig. 9d). Moreover, the depth of 'pit' is the difference of capacity between with and without optimisation. We manually trigger the load balancing at each discrete time from $t = 0$ to $t = 80$ and display the change of 'pit' width and depth in Fig. 10. As shown by the blue dot line, the bottom of the 'pit' drops fast after it appears ($13.2\ kb/s$ at $t = 29$) until approximately $7.36\ kb/s$ at $t = 60$. Next, the width of poor-performance (black cross line) shows that with different activation times, the degradation 'pit' firstly does not occur (from $t = 0$ to $t = 28$), then increases non-linearly (from $t = 29$ to $t = 38$), finally increases linearly (from $t = 39$ to $t = 80$). The best time to trigger the proactive load balancing is at the end of the 'first period' ($t = 28$) because it has no poor-performance 'pit' while reserving the maximum time for the prediction calculation. Therefore, it is valuable to find this time point.

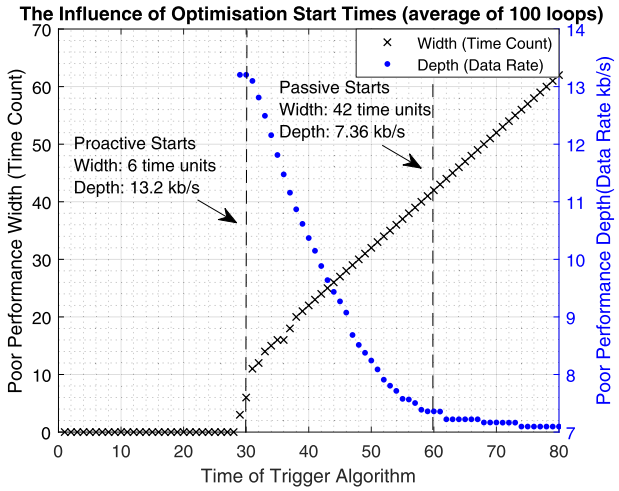


FIGURE 10. **This result indicates how wide and deep the poor performance 'pit' is when we choose different trigger times.**

However, currently, we have to simulate all the discrete time points to visualise the phenomenon. It is inefficient and impossible in real-world conditions. In that case, we propose a feasible design to take advantage of the 'constant-nonlinear-linear' characteristic for reducing the complexity of simulation as much as possible.

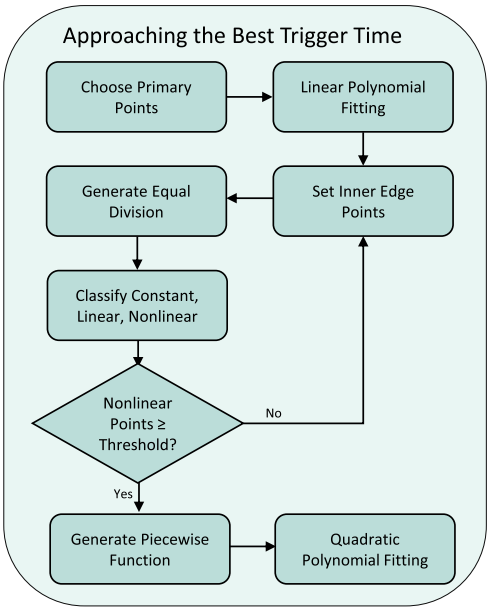


FIGURE 11. The process of modelling the poor performance width for approaching the best trigger time for proactive load balancing.

The design flowchart of approximating the best trigger time is shown in Fig. 11. It follows the following steps:

- Choose primary points: this step chooses two time points at the very beginning t_{b1} and t_{b2}, then simulates two time points at the end t_{e1} and t_{e2}.
- Linear polynomial fitting: The corresponding width value for t_{b1} and t_{b2} is zero, and the t_{e1} and t_{e2} will generate a linear model through linear polynomial fitting.
- Set inner edge points: the points t_{b2} and t_{e1} are currently the closest points to the best triggering time, they are denoted as inner edge points.
- Generate equal division: the period between the two inner edge points is equally divided into n_d segments that there are $n_d - 1$ new time points.
- Classify constant, linear, and nonlinear: the system simulates the 'pit' width of these three time points, then classifies them to constant (0) or linear model. Otherwise, the time points belong to the non-linear part. We need to count the nonlinear time points $n_{nonlinear}$.
- Check number of nonlinear points: we manually set a threshold $\theta_{nonlinear}$ to check if there are enough nonlinear points for quadratic polynomial fitting ($n_{nonlinear} \geq \theta_{nonlinear}$).
- If no ($n_{nonlinear} < \theta_{nonlinear}$): The system needs more iterations by setting new inner edge point, classifying, and checking nonlinear again. This process will approach the best triggering time.
- If yes ($n_{nonlinear} \geq \theta_{nonlinear}$): the system will generate piecewise function (constant, nonlinear function, linear function). The nonlinear part is modelled by the quadratic polynomial fitting.
- Finally, the best time for activating proactive optimisation is at the edge between the constant function and nonlinear function.

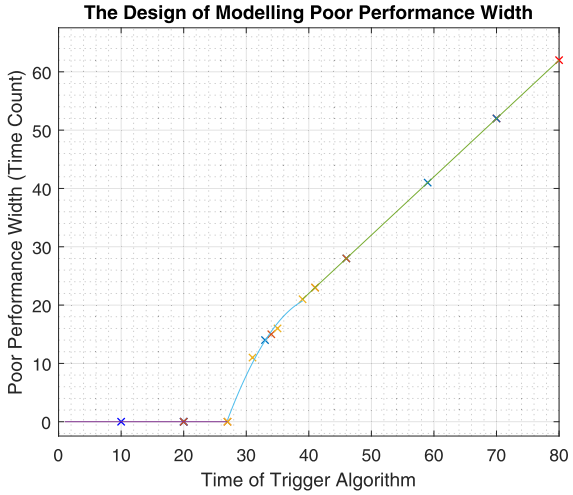


FIGURE 12. The results of applying the design to model the poor-performance width and approach the best time for activating proactive load balancing. The time point $t = 27$ is the edge between the constant function and the nonlinear function, so it is the expected triggering time.

The above design is tested through setting $t_{b1} = 10$, $t_{b2} = 20$, $t_{e1} = 70$, $t_{e2} = 80$, $n_d = 4$, and $\theta_{nonlinear} = 3$. We present the result in Fig. 12. The line indicates the generated piecewise function, and the thirteen crosses represent the total simulation attempts. This result concludes that the design is feasible to find the best activation time of proactive load balancing ($t = 27$ in the current case study) with reducing the simulation times form 80 to 13.

VI. CONCLUSION

In general, this paper has three main contributions, the first one is designing the context-aware module about event (irregularity) detection associated with hotspots in urban regions, the second contribution is coupling the context-awareness and the proactive load balancing in urban regions, and the third one is optimising the proactive schemes about forecasting the best activation time. Our design for approaching the best triggering time satisfies the requirements of reducing time complexity.

Moreover, the proposed frameworks benefit not only the proactive load balancing but also other optimisations, such as proactive caching and interference management. These optimisation algorithms also require a traffic pattern for resource allocation. In the future, prediction granularity (the distinguishable scale of detail in the context) and prediction error should be addressed. In particular, different optimisations require various granularity (e.g., the small-cell level for deployment, and inter-cell level for load balancing). Also, the privacy-free data set takes a little part in the overall data, so using only part of them can lose dimensions and generate detection errors. Even though private data faces sensitive privacy problems, it is still necessary to find a privacy-free way to improve context accuracy by using this data.

REFERENCES

[1] A. Imran and A. Zoha, "Challenges in 5G: How to empower SON with big data for enabling 5G," *IEEE Netw.*, vol. 28, no. 6, pp. 27–33, Nov. 2014.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[3] Y. Li, Y. Zhang, K. Luo, T. Jiang, Z. Li, and W. Peng, "Ultra-dense HetNets meet big data: Green frameworks, techniques, and approaches," *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 56–63, Jun. 2018.

[4] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2392–2431, 4th Quart., 2017.

[5] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, "Load balancing in downlink LTE self-optimizing networks," in *Proc. IEEE 71st Veh. Technol. Conf.*, 2010, pp. 1–5.

[6] I. Viering, M. Dottling, and A. Lobinger, "A mathematical perspective of self-optimizing wireless networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–6.

[7] H. Hu, J. Zhang, X. Zheng, Y. Yang, and P. Wu, "Self-configuration and self-optimization for LTE networks," *IEEE Commun. Mag.*, vol. 48, no. 2, pp. 94–100, Feb. 2010.

[8] J. M. Ruiz-Avilés, S. Luna-Ramírez, M. Toril, and F. Ruiz, "Traffic steering by self-tuning controllers in enterprise LTE femtocells," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 337, Dec. 2012.

[9] J. Rodriguez, I. De La Bandera, P. Munoz, and R. Barco, "Load balancing in a realistic urban scenario for LTE networks," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, May 2011, pp. 1–5.

[10] A. Aguilar-Garcia, S. Fortes, A. F. Duran, and R. Barco, "Context-aware self-optimization: Evolution based on the use case of load balancing in small-cell networks," *IEEE Veh. Technol. Mag.*, vol. 11, no. 1, pp. 86–95, Mar. 2016.

[11] C. A. S. Franco and J. R. B. De Marca, "Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach," in *Proc. 7th IEEE Latin-Amer. Conf. Commun. (LATINCOM)*, Nov. 2015, pp. 1–5.

[12] S. S. Mwanje and A. Mitschele-Thiel, "A Q-Learning strategy for LTE mobility Load Balancing," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 2154–2158.

[13] J. Xu, L. Tang, Q. Chen, and L. Yi, "Study on based reinforcement Q-learning for mobile load balancing techniques in LTE-a HetNets," in *Proc. IEEE 17th Int. Conf. Comput. Sci. Eng.*, Dec. 2014, pp. 1766–1771.

[14] S. Tomforde, A. Ostrovsky, and J. Hähner, "Load-aware reconfiguration of LTE-antennas dynamic cell-phone network adaptation using organic network control," in *Proc. 11th Int. Conf. Inform. Control, Autom. Robot.*, vol. 1, Sep. 2014, pp. 236–243.

[15] R. Joyce and L. Zhang, "Self organising network techniques to maximise traffic offload onto a 3G/WCDMA small cell network using MDT UE measurement reports," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2212–2217.

[16] N. P. Kuruvatti, A. Klein, and H. D. Schotten, "Prediction of dynamic crowd formation in cellular networks for activating small cells," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015.

[17] N. P. Kuruvatti, J. F. S. Molano, and H. D. Schotten, "Mobility context awareness to improve quality of experience in traffic dense cellular networks," in *Proc. 24th Int. Conf. Telecommun. (ICT)*, May 2017.

[18] F. Botta, H. S. Moat, and T. Preis, "Quantifying crowd size with mobile phone and Twitter data," *Roy. Soc. Open Sci.*, vol. 2, no. 5, May 2015, Art. no. 150162.

[19] B. Yang, W. Guo, B. Chen, G. Yang, and J. Zhang, "Estimating mobile traffic demand using Twitter," *IEEE Wireless Commun. Lett.*, vol. 5, no. 4, pp. 380–383, Aug. 2016.

[20] H. Klessig, H. Kuntzschmann, L. Scheuvens, B. Almeroth, P. Schulz, and G. Fettweis, "Twitter as a source for spatial traffic information in big data-enabled self-organizing networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–5.

[21] H. Becker, D. Iter, M. Naaman, and L. Gravano, "Identifying content for planned events across social media sites," in *Proc. WSDM*, 2012, pp. 533–542.

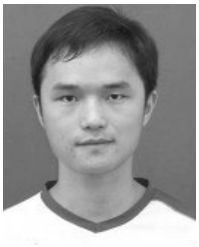
[22] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection," in *Proc. 2nd ACM SIGSPATIAL Int. Workshop Location Based Social Netw. (LBSN)*, New York, NY, USA, 2010, p. 1.

[23] K. Watanabe, M. Ochi, M. Okabe, and R. Onai, "Jasmine: A real-time local-event detection system based on geolocation information propagated to microblogs," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, New York, NY, USA, 2011, p. 2541.

[24] S. Fortes, D. Palacios, I. Serrano, and R. Barco, "Applying social event data for the management of cellular networks," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 36–43, Nov. 2018.

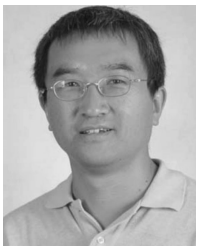
[25] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, "A density based algorithm for discovering density varied clusters in large spatial databases," *Int. J. Comput. Appl.*, vol. 3, no. 6, pp. 1–4, Jun. 2010.

BO MA received the bachelor's degree in automatic control engineering from Southeast University, China, in 2014, and the M.Sc. degree (Hons.) in communication and signal processing from Newcastle University, U.K., in 2016. He is currently pursuing the Ph.D. degree in wireless systems with the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, U.K. His main research interests include proactive network optimization, the application of data analysis, and machine learning algorithms in 5G networks.



BOWEI YANG received the B.S. and Ph.D. degrees in computer science from Zhejiang University, China, in 2006 and 2011, respectively. In 2016, he started the AI assisted wireless communications researches in The University of Sheffield, U.K., and was promoted to the Associate Professor at the School of Aeronautics and Astronautics, Zhejiang University, China. His research interests include software-defined networking, satellite networking, and distributed computing and storage.

YUNPENG ZHU received the B.S. degree in mechanical engineering and automation and the M.S. degree in mechanical manufacturing and automation from Northeastern University, Shenyang, China, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Department of Automatic Control and System Engineering, The University of Sheffield, Sheffield, U.K. His current research interest includes the analysis and design of nonlinear systems.



JIE ZHANG received the Ph.D. degree in automatic control and electronic engineering from the East China University of Science and Technology, Shanghai, China, in 1995. From January 2002 to January 2011, he was with the University of Bedfordshire, where he became a Lecturer, a Reader, and a Professor, in 2002, 2005, and 2006, respectively. He has been a Full Professor and the Chair in wireless systems with the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, U.K., since January 2011. He is currently a Visiting Professor with the Chongqing University of Posts and Telecommunications and the East China University of Science and Technology. With his students/colleagues, he has pioneered research in femto/small cell and HetNets and published some of the earliest and/or most cited publications in these topics. Since 2005, he has also been awarded more than 20 grants by the Engineering and Physical Sciences Research Council, the EC FP6/FP7/H2020, and industry, including some of world's earliest research projects on femtocell/HetNets. He has cofounded Ranplan Wireless Network Design Ltd., Cambridge, U.K., which produces a suite of world-leading in-building distributed antenna systems, indoor - outdoor small cell/HetNet network planning, and optimization tools iBuildNet that have been used by Ericsson, Huawei, and Cisco.