1. Introduction :

L'augmentation croissante du trafic de données a posé de grands défis aux opérateurs mobiles pour augmenter leur capacité de traitement des données, ce qui entraîne une consommation d'énergie et des coûts de déploiement importants. Avec l'émergence de l'architecture Cloud Radio Access Network (C-RAN), les unités de traitement des données peuvent désormais être centralisées dans les centres de données et partagées entre les stations de base. En mappant un cluster de stations de base avec des schémas de trafic complémentaires à une unité de traitement de données, l'unité de traitement peut être pleinement utilisée à différentes périodes de temps, et la capacité requise à déployer devrait être inférieure à la somme des capacités d'une seule base stations. Cependant, étant donné que les schémas de trafic des stations de base sont très dynamiques à différents moments et endroits, il est difficile de prévoir et de caractériser les schémas de trafic à l'avance pour réaliser des schémas de regroupement optimaux. Dans cet article, nous abordons ces problèmes en proposant un cadre d'optimisation C-RAN basé sur l'apprentissage en profondeur. Premièrement, nous exploitons un modèle de mémoire à long terme et multivariée à long terme (MuLSTM) pour apprendre la dépendance temporelle et la corrélation spatiale entre les modèles de trafic de la station de base et faire des prévisions de trafic précises pour une période future. Ensuite, nous construisons un graphique pondéré pour modéliser la complémentarité des stations de base en fonction de leurs modèles de trafic et proposons un algorithme DCCA (Distance-Constrained Complementarity-Aware) pour trouver des schémas de clustering de stations de base optimaux dans le but d'optimiser l'utilité des capacités et les coûts de déploiement. . Nous évaluons les performances de notre framework en utilisant des données en deux mois à partir de réseaux mobiles réels à Milan et Trentin, en Italie. Les résultats montrent que notre méthode augmente efficacement l'utilité moyenne de la capacité à 83,4% et 76,7%, et réduit le coût de déploiement global à 48,4% et 51,7% de l'architecture RAN traditionnelle dans les deux ensembles de données, respectivement, ce qui surpasse constamment l'état de la méthodes de référence de pointe.

C-RAN et Problèmes de clustering :

Heureusement, avec l'évolution rapide des architectures de réseaux mobiles, l'émergence du Cloud Radio Access Network (C-RAN) (C. M. R. Institute, 2011) a offert de nouvelles opportunités pour relever les défis ci-dessus. Dans C-RAN, une station de base traditionnelle est divisée en deux composants: une tête radio distante (RRH) pour la communication radio et une unité de bande de base (BBU) pour le traitement mobile des données. Les BBU sont en outre détachées des RRH et hébergées dans des pools de BBU centralisés (Checko et al., 2015). Les pools RRH et BBU sont généralement connectés via une fibre optique à haute vitesse (Checko et al., 2015). En regroupant les RRH avec des schémas de trafic complémentaires vers une BBU, la capacité de traitement des données de la BBU peut être partagée entre les RRH à différentes périodes, augmentant ainsi l'utilité de la capacité de la BBU (Bhaumik et al., 2012). De plus, la capacité requise de la BBU devrait être inférieure à la somme des capacités des stations de base uniques, ce qui entraînera une baisse des coûts de déploiement. Par exemple, sur la figure 1, si nous regroupons les RRH dans le quartier des affaires (bleu) et dans la zone résidentielle (rouge) en une BBU, le modèle de trafic agrégé deviendra relativement stable et la BBU aura une utilité de capacité supérieure (Fig.1b). Pendant ce temps, la capacité requise pour le BBU peut être réduite de la somme des deux pics (1,50 = 0,65 + 0,85) à une valeur agrégée inférieure (1,10). En résumé, en regroupant les BBU de plusieurs stations de base dans un pool de BBU centralisé, le gain de multiplexage statistique (Checko et al., 2015) peut être obtenu dans l'architecture C-RAN (C.M.R Institute, 2011).
Afin de libérer la puissance de l'architecture C-RAN, il est très important de caractériser les modèles de trafic des RRH et de regrouper les RRH complémentaires en un ensemble de BBU (Bhaumik et al., 2012; Chen et al., 2016a), afin de maximiser l'utilité des capacités et de minimiser les coûts de déploiement. Cependant, étant donné que le trafic de données généré dans les RRH est très dynamique sur différents moments et emplacements, il est très difficile de prévoir et de caractériser à l'avance les modèles de trafic RRH, ce qui entrave l'optimisation du clustering RRH et du mappage BBU. Plus précisément, étant donné un ensemble de RRH dans une ville, nous devons prévoir avec précision leurs modèles de trafic de données dans une période future (par exemple, un jour), et trouver des schémas optimaux pour regrouper les RRH avec des modèles de trafic complémentaires, et les mapper(distribuer) à un ensemble de BBU pour cette période de temps. Pour atteindre ces objectifs, nous devons résoudre les problèmes suivants:
I-- Comment prévoir le trafic RRH pour une période future?
Le trafic de données dans chaque RRH peut varier considérablement, selon les impacts des contextes temporels (par exemple, les jours de semaine ou les week-ends), la mobilité humaine et les événements sociaux, etc. En outre, le trafic de données des RRH situés dans des zones fonctionnelles similaires peut démontrer un potentiel de cor - relations. Par exemple, pendant les jours de la semaine, les RRH situés dans les quartiers d'affaires observent généralement des pics de trafic de données pendant les heures de travail et de faibles volumes de trafic de données la nuit. La capture de la dépendance temporelle cachée et de la corrélation spatiale entre les modèles de trafic RRH n'est pas triviale à l'aide de modèles de séries chronologiques de pointe, tels que ARIMA (Hamilton, 1994) ou les réseaux de neurones (Zhang, 2003). Par conséquent, nous devons favoriser des techniques plus efficaces pour la prévision précise du modèle de trafic RRH.

II--Comment mesurer la complémentarité entre RRH?
Afin de partager et de réutiliser efficacement la capacité d'une BBU mappée à un cluster de RRH, les pics de trafic des RRH du cluster doivent être dispersés temporellement (c'est-à-dire qu'ils se produisent à différentes heures). Pendant ce temps, pour tirer pleinement parti de la BBU mappée sur un cluster et éviter la surcharge de la BBU, le trafic de cluster agrégé doit être proche de la capacité de la BBU dans une mesure maximale, tout en ne dépassant pas trop la capacité de la BBU. Par conséquent, nous devons prendre en compte les deux aspects, c'est-à-dire la distribution des pics et l'utilité de la capacité, pour concevoir une métrique efficace pour mesurer la complémentarité des RRH.

III--Comment regrouper de manière optimale les RRH complémentaires en BBU?
Compte tenu des prévisions de trafic et des mesures de complémentarité des RRH, il existe potentiellement un nombre énorme de schémas pour regrouper ces RRH et les mapper aux BBU dans un pool. Le schéma optimal doit non seulement maximiser l'utilité moyenne de la capacité BBU, mais également minimiser le coût de déploiement global. De plus, afin de prendre en charge le transfert rapide et le déchargement de contenu entre les RRH voisins (Checko et al., 2015; Zhao et al., 2016), les distances entre un cluster de RRH doivent être limitées dans une fourchette raisonnable. Par conséquent, nous devons concevoir un algorithme efficace pour trouver le schéma de regroupement RRH optimal sous la contrainte de distance.

2. Related work :

L'un des principaux problèmes de l'architecture C-RAN est de concevoir un schéma de clustering RRH optimal et de les connecter au pool BBU. Un schéma optimal devrait faciliter l'utilitaire de capacité BBU dans le pool, réduire le coût de déploiement et également empêcher le délai de propagation entre les RRH et le pool BBU (Checko et al., 2015). À cette fin, Bhaumik et al. (2012) ont proposé CloudIQ, un cadre pour partitionner un ensemble de RRH en groupes et traiter les signaux dans un centre de données partagé. Étant donné que la distance entre les centres de données et les RRH peut entraîner un retard potentiel entre les RRH distants et le centre de données (Checko et al., 2015). Lee et al. (2013) ont proposé un schéma de coopération RRH avec un regroupement dynamique en C-RAN, mais l'objectif de la coopération est de dériver le signal sur brouillage pour l'évaluation RRH. L'une des idées très pertinentes pour notre travail a été illustrée dans (Zheng et al., 2016), qui a exploré des approches pour intégrer l'analyse des mégadonnées avec l'optimisation du réseau dans la 5G, notamment en exploitant les données historiques pour optimiser l'allocation des ressources dans les BBU centralisées en C- RAN.nn
I--Time series forecasting models

Au cours des dernières décennies, la modélisation et la prévision des séries chronologiques ont été largement étudiées dans la littérature (Hamilton, 1994; Dorffner, 1996; Zhang, 2003). Nous examinons deux des approches de pointe dans l'analyse des séries chronologiques et discutons de leurs inconvénients à résoudre notre problème.

Modèles auto-régressifs à moyenne mobile intégrée (ARIMA): dans l'analyse des séries chronologiques, les modèles ARIMA sont couramment utilisés pour ajuster les données d'une série chronologique et pour prévoir les variations futures de la série. Les modèles ARIMA extraient explicitement d'une série chronologique trois fonctionnalités intuitives, à savoir l'auto-régression, la mobilité moyenne et l'intégration.

La partie d'auto-régression (AR) indique que la variable évolutive d'une série chronologique est régressée sur ses propres valeurs décalées.

La partie moyenne mobile (MA) indique que l'erreur de régression peut être représentée comme une combinaison linéaire de termes d'erreur dépendant des valeurs du passé.

La partie intégration (I) est appliquée au modèle de régression pour représenter des séries chronologiques non stationnaires (c'est-à-dire que la variable dans la série chronologique montre une tendance à l'augmentation ou à la diminution).

Les modèles ARIMA sont capables de s'adapter rapidement aux changements soudains de tendance, et il s'est avéré efficace dans de nombreux problèmes de prévision à court terme (Sang et Li, 2002). Cependant, pour les problèmes de prévision à long terme qui impliquent de prévoir plusieurs étapes futures, l'erreur des modèles ARIMA s'accumule de manière significative et la confiance dans les prévisions diminue rapidement à mesure que l'étape de prévision se développe (Box et al., 2015). Dans notre problème, nous devons prévoir avec précision le trafic RRH pendant plusieurs heures pour prévoir les modèles de trafic à l'avenir pour le cluster RRH, ce qui pose de grands défis pour les modèles ARIMA.

Modèles de réseau de neurones artificiels (ANN): Récemment, les modèles ANN sont largement utilisés pour comprendre les séries chronologiques et prévoir la tendance future en s'appuyant sur une technique basée sur les fenêtres coulissantes (Dorffner, 1996), qui peut être nommée windowed-ANN ou WANN. Plus précisément, cette technique découpe d'abord une série chronologique en plusieurs fenêtres de longueur égale, puis introduit ces fenêtres dans un modèle ANN en tant qu'entités. Le résultat du modèle est la prévision des valeurs futures de la série chronologique, qui peuvent être des résultats à court ou à long terme, selon le scénario d'application. Les modèles WANN ont été appliqués dans divers domaines, tels que le marché financier (Azoff, 1994) et la recherche opérationnelle (Zhang et Qi, 2005). Cependant, l'un des plus gros problèmes du modèle WANN est son incapacité à modéliser la dépendance temporelle entre les éléments dans chaque fenêtre de série chronologique. En fait, les éléments d'une fenêtre sont traités également comme des fonctions d'entrée et l'ordre séquentiel des éléments est donc ignoré. En conséquence, le modèle WANN peut faire des prévisions fluctuantes et incohérentes qui ne sont pas souhaitées dans notre problème.

Dans ce travail, nous proposons une architecture d'apprentissage profond (LeCun et al., 2015) pour modéliser la dépendance temporelle du trafic RRH et les corrélations spatiales entre RRH dans un cadre unifié. Ce type de cadre d'apprentissage profond spatio-temporel a été largement utilisé dans la prédiction du trafic IP et des réseaux de transport (Nie et al., 2016; Zhang et al., 2016), la compréhension des dossiers de santé électroniques (Rajkomar et al., 1801), et l'analyse du comportement des réseaux sociaux (Zhang et al., 2017).

II--Mobile data analytics

Avec l'émergence de diagrammes de détection et de calcul omniprésents (Zhang et al., 2011), un nombre massif de données mobiles peuvent désormais être collectées soit par des paradigmes de mobile crowdsensing (Wang et al., 2016, 2017; Guo et al., 2015) ou à partir des infrastructures des opérateurs. Ces mégadonnées mobiles hétérogènes font l'objet d'une analyse approfondie dans la littérature afin de récupérer des informations intéressantes et informatives (Chen et al., 2014, 2016b; Yang et al., 2015; Tan et al., 2016). Par exemple, Barlacchi et al. (2015) a publié un ensemble de données à grande échelle Call Detail Records (CDR) de Telecom Italia, contenant deux mois d'appels, de SMS et de données de trafic réseau de la ville de Milan et du Trentin, en Italie.

Sur la base de l'ensemble de données, Furno et al. (2016) ont proposé un cadre d'analyse de données pour construire des profils de la demande de trafic à l'échelle de la ville et identifier des situations inhabituelles dans les usages du réseau, visant à faciliter la conception et la mise en œuvre de réseaux cognitifs cellulaires.

Cici et al. (2015) ont étudié la décomposition des séries d'activités sur les téléphones cellulaires et connectent les séries décomposées aux activités socio-économiques, telles que les schémas de travail réguliers et les événements opportunistes (Chen et al., 2017b).

Cependant, l'application des données du réseau mobile du monde réel à l'optimisation C-RAN n'a pas encore été largement étudiée dans la littérature, car les travaux précédents se concentrent principalement sur des approches basées sur la simulation pour modéliser le trafic réseau (Zhan et Niyato, 2017; Zhang et al ., 2016).

Dans ce travail, nous exploitons des ensembles de données ouverts à grande échelle d'opérateurs de réseaux mobiles du monde réel pour comprendre les modèles de trafic dans des réseaux réels, puis menons des études d'optimisation C-RAN sur la base des connaissances découvertes à partir de ces ensembles de données mobiles.

3. Préliminaires et framework

I--Préliminaires

Dans les architectures de réseaux mobiles, un ensemble de stations de base est déployé sur des zones géographiques appelées cellules (Tse et Viswanath, 2005). Chaque station de base fournit à la cellule la couverture réseau qui peut être utilisée pour la transmission de la voix et des données. Avec l'émergence récente des smartphones et des tablettes, le trafic de données généré par les utilisateurs connectés aux RRH augmente rapidement (Cisco, 2016; J. Research, 2011).

Afin de comparer la capacité de traitement des données des stations de base, de nombreux opérateurs ont collecté à grande échelle des données statistiques de trafic RRH et les ont rendues publiques (Zheng et al., 2016). Dans cet article, nous exploitons l'ensemble de données publié par Telecom Italia pour l'initiative Big Data Challenge (Barlacchi et al., 2015). Nous extrayons deux mois de données de trafic réseau du 11/01/2013 au 31/12/2013 dans la ville de Milan, Italie et la province de Trentino, Italie. Nous collectons également les emplacements des stations de base actives à

Milan et au Trentin pendant les deux mois à partir de CellMapper.net 1 et déduisons le volume de trafic de chaque station de base pendant les deux mois sur une base horaire. Les étapes de prétraitement des données de trafic seront détaillées dans la section évaluation.

Dans ce travail, nous considérons une architecture C-RAN avec un pool BBU pour le réseau mobile à l'échelle de la ville. Les avantages de l'adoption d'une telle piscine centralisée sont doubles.
Premièrement, le coût de déploiement et la consommation d'énergie peuvent être considérablement réduits en utilisant des technologies de virtualisation des centres de données (Qian et al., 2015).
Deuxièmement, le transfert de transfert et le déchargement de contenu entre les RRH peuvent être traités en interne dans le pool, ce qui réduit considérablement les retards et augmente le débit (Checko et al., 2015). Les BBU du pool sont implémentées en tant que machines virtuelles avec des capacités prédéfinies spécifiques. Dans ce travail, à des fins de comparaison et de simplicité, nous supposons que la capacité des BBU est fixe et égale aux BBU sur site dans l'architecture traditionnelle. Nous discutons des détails de l'outil dans la section d'évaluation.

II—Framework

Nous proposons un framework en deux phases pour regrouper dynamiquement les RRH complémentaires en un ensemble de BBU, de sorte que l'utilitaire de capacité de BBU et le coût de déploiement de l'ensemble du réseau puissent être optimisés.
Dans la phase de profilage dynamique des RRH, étant donné un ensemble de RRH à un moment donné, nous proposons d'abord une approche basée sur l'apprentissage en profondeur pour prévoir les modèles de trafic des RRH dans une période future en fonction de leur données de trafic historiques, puis calculer la complémentarité des RRH en utilisant une métrique basée sur l'entropie proposée.
Dans la phase de regroupement dynamique des RRH, nous construisons d'abord un modèle graphique pour représenter la complémentarité entre les RRH, puis proposons un algorithme de regroupement à distance limitée pour regrouper les RRH avec des modèles de trafic complémentaires. Nous expliquons les détails de ce cadre dans les sections suivantes.

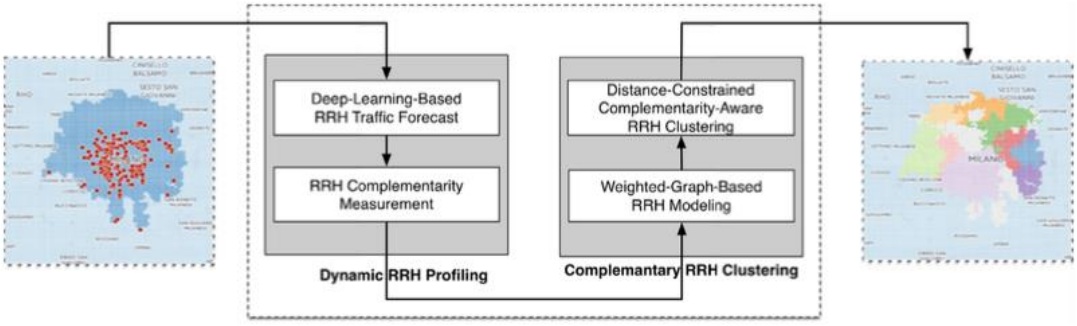

Fig. 2. Framework overview.

A--phase de profilage dynamique des RRH

Afin de regrouper les RRH avec des modèles de trafic complémentaires à une BBU, nous devons être en mesure de prévoir le modèle de trafic de chaque RRH pour une période de temps future. Étant donné que le trafic des RRH varie considérablement et présente des corrélations spatiales, nous proposons une approche basée sur l'apprentissage profond pour modéliser la dynamique spatio-temporelle et prévoir avec précision le futur modèle de trafic.
Sur la base des prévisions de trafic, nous caractérisons dynamiquement la complémentarité des RRH, en nous nous concentrons sur la distribution de pointe et l'utilité de la capacité d'un cluster de RRH, et concevons une métrique basée sur l'énergie pour caractériser leur complémentarité.

A.I--Prévision du trafic RRH

Sur la base des données de trafic historiques, nous observons que les modèles de trafic des RRH sont très dynamiques dans différents contextes temporels. Par exemple, la figure 3 montre les modèles de trafic de deux RRH situés dans deux quartiers d'affaires de Milan pendant une semaine, respectivement. Nous observons des pics de trafic importants pendant les heures de travail en semaine et des services publics de faible capacité pendant les heures de repos. De plus, nous observons que les modèles de trafic des RRH situés dans des zones fonctionnelles similaires montrent généralement des tendances similaires. Par exemple, sur la figure 3, les modèles de trafic dans les deux quartiers d'affaires de Milan montrent des modèles de semaine-week-end similaires.

a--Idée de base

Afin de prévoir avec précision les modèles de trafic des RRH dans une période de temps future, nous devons être en mesure de capturer efficacement leur dépendance temporelle et leur corrélation spatiale. Cependant, ce n'est pas tripartite en utilisant les techniques de pointe.
Dans ce travail, nous proposons une approche basée sur l'apprentissage profond pour notre problème. Plus précisément, nous exploitons le réseau de neurones récurrents (RNN) pour capturer automatiquement la dépendance temporelle intrinsèque dans nos données de trafic. Un RNN est un type spécial de réseau de neurones conçu pour les problèmes d'exploration de motifs séquentiels (Sutskever et al., 2014). Construit sur l'architecture fenêtre-ANN, un RNN comporte des boucles supplémentaires aux neurones dans les couches du réseau neuronal. Chaque neurone peut transmettre son signal latéralement en plus de transmettre à la couche suivante, et par conséquent, la sortie du réseau pour une fenêtre peut renvoyer en tant qu'entrée au réseau pour la fenêtre suivante. De telles connexions récurrentes ajoutent de l'état ou de la mémoire à l'architecture window-ANN et lui permettent d'apprendre et d'exploiter la dépendance temporelle intrinsèque dans la série temporelle.
Malheureusement, la formation efficace d'un RNN est techniquement difficile en raison du problème de gradient qui disparaît ou explose, c'est-à-dire que les poids dans la procédure de formation sont rapidement devenus si petits qu'ils n'ont aucun effet (gradients de disparition) ou si grands qu'ils entraînent de très grands changements (explosion des gradients). Pour surmonter ce problème, les chercheurs ont proposé le modèle LSTM (Long Short-Term Memory Network) (Gers et al., 2002), qui présente les concepts de cellules de mémoire et oublie les portes pour générer un flux de données cohérent entre les couches du réseau et garder les poids stables (Hochreiter et Schmidhuber, 1997).
Dans ce travail, nous exploitons le modèle LSTM pour apprendre efficacement la dépendance temporelle de nos données de trafic.
L'autre défi consiste à modéliser la corrélation spatiale entre les RRH dans le réseau. Les approches mentionnées ci-dessus modélisent généralement le trafic de chaque RRH comme une série temporelle distincte, ce qui rend difficile la capture de la corrélation entre les RRH.
Dans ce travail, nous proposons une approche multivariée à long terme du réseau de mémoire à court terme (MuLSTM) pour modéliser le trafic RRH dans une ville dans un modèle unifié, en mettant chaque trafic RRH comme une séquence pour la formation et la prévision, et par conséquent apprendre la corrélation spatiale entre RRH.

b--Le modèle MuLSTM

Avant d'introduire le modèle MuLSTM, nous définissons plusieurs terminologies importantes comme suit:
Définition 1. Remote Radio Head (RRH):
Les RRH d'un réseau mobile à l'échelle de la ville peuvent être décrits comme un ensemble de points désignés par le triplet suivant:
{r, r = (rid, lat, lng)}
où rid, lat, lng sont l'ID, la latitude et la longitude uniques du RRH.

Définition 2. Trafic du RRH:
Le trafic de données mobiles collecté à partir de chaque RRH peut être désigné par un ensemble de séquences de longueur fixe:
{f , fi = [ui (1), ..., ui (t), ..., ui (Nt)]}
où ui (t) est le volume de trafic de RRH i dans l'intervalle de temps t ($1 \leq t \leq Nt$). Dans ce travail, nous utilisons une durée d'une heure.
Avec les données de trafic collectées, nous organisons d'abord le trafic RRH collecté dans une matrice FRNt × RNr, où Nt est le nombre d'intervalles de temps, et Nr désigne le nombre de RRH dans le réseau. Nous désignons le trafic de RRH que nous avons observé jusqu'au temps t comme F ([0, t], :), et le trafic de RRH que nous aimerions prévoir dans une période future Δt comme F ([t, t + Δt], :).
Dans ce travail, pour simplifier l'implémentation, nous utilisons une plage horaire d'une heure, et Δt = 24 h avec t mod 24 = 0, c'est-à-dire que nous prévoyons le trafic horaire des RRH pour le lendemain à la fin de chaque journée, et on met dynamiquement à jour le schéma de clustering RRH en fonction des prévisions. Sur cette base, nous générons un ensemble de trafics instantanés à partir de la matrice de trafic, qui est définie comme suit.
Définition 3. RRH Traffic Snapshot(Trafic instantané du RRH):
Un Traffic Snapshot est défini comme une matrice Fi, qui correspond au trafic de tous les RRH pendant une période de temps donnée Δt, c'est-à-dire F = {Fi, Fi = F ([(i − 1) * Δt, i * Δt], :), i = 1,2, …}

Afin de faire des prévisions de trafic, nous apprenons sur un modèle séquence à séquence (Sutskever et al., 2014) en utilisant un modèle LSTM multivarié unifié. Lors de chaque prévision, le modèle accepte Fi en entrée et sort Fi+1. Notez qu'un tel modèle est appelé un modèle séquentiel plusieurs à plusieurs car l'entrée et la sortie contiennent toutes deux des intervalles de temps Δt, et l'ordre des intervalles de temps joue un rôle important dans la mise en forme de la structure interne du modèle. De plus, le trafic des RRH est entré dans le modèle en tant qu'entités multivariées simultanément, ce qui permet au modèle d'apprendre la corrélation spatiale entre les RRH.
Enfin, nous développons la conception de la structure du réseau MuLSTM. En général, le modèle MuLSTM suit la structure codeur-décodeur en empilant deux couches LSTM L1 et L2.
--Le codeur L1 accepte un snapshot de taille [Δt, Nr], apprend les structures temporelles et spatiales dans le snapshot et transmet les séquences codées au décodeur.
--Le décodeur fait alors des prévisions pour un futur snapshot de taille [Δt, Nr] sur la base des structures apprises. Le modèle est formé à l'aide de l'algorithme populaire de rétropropagation dans le temps (BPTT) pour plusieurs itérations. Nous élaborons les détails des paramètres du modèle dans la section évaluation.

A.II--Mesure de complémentarité RRH
Une fois que nous avons la prévision du trafic pour le lendemain, nous sommes en mesure d'évaluer la complémentarité des RRH dans ce contexte et de regrouper les RRH complémentaires en une BBU. Nous considérons les deux aspects suivants pour concevoir une métrique de complémentarité efficace des RRH.
a--Distribution maximale
Le volume de trafic atteignant le pic d'un ensemble de RRH regroupés dans la même BBU doit être dispersé dans différents contextes temporels, afin que la capacité de la BBU puisse être partagée entre ces RRH. À cette fin, nous concevons une métrique basée sur l'entropie pour mesurer la distribution de crête d'un ensemble de RRH. Plus précisément, étant donné un ensemble de RRH groupés C = {r1, ..., rn}, nous trouvons d'abord les heures de pointe dans leurs profils de trafic, respectivement, c'est-à-dire, T (ri) = {ti1, ti2, ..., tim}, $1 \leq im \leq 24$ (1) où tim désigne le mième temps de pointe de ri (le mieme pic). Ensuite, nous calculons l'entropie de Shannon (Lin, 1991) des heures de pointe de l'ensemble des RRH groupées T(C)=U T(ri) comme suit: $H(C)=-\Sigma p_k \log(p_k)$ $k=1,\ldots, K$
où K = | T (C) | correspond à la quantité totale de pics en C, et pk est la probabilité d'observer l'heure de pointe correspondante dans l'ensemble T(C). Une plus grande valeur d'entropie d'un cluster RRH indique que les RRH sont plus complémentaires dans le cluster w.r.t. modèles de trafic.
b-- Utilitaire de capacité
Pour utiliser pleinement le BBU mappé à un cluster C, l'agrégat du trafic du cluster devrait être proche de la capacité BBU à différentes heures de la journée. Pendant ce temps, pour éviter la surcharge de la BBU, le trafic de cluster agrégé ne doit pas dépasser trop la capacité de la BBU. À cette fin, nous concevons la métrique suivante pour mesurer quantitativement l'utilité de la capacité d'un BBU B mappé sur un cluster C:

$$U(C) = \left(\frac{\text{mean}\mathbf{f}(C)}{|B|}\right)^{-\ln \frac{\text{mean}\mathbf{f}(C)}{|B|}}$$

où f (C) = ∑n f (ri) désigne le profil de trafic agrégé du i = 1
Cluster RRH et | B | est la capacité BBU fixe mesurée en volume de trafic. La figure 4 montre la courbe de la fonction d'utilité de capacité, qui atteint son maximum lorsque le volume de trafic agrégé moyen est égal à la capacité BBU.
Enfin, nous calculons la complémentarité du cluster RRH C comme suit:

$$M(C) = U(C) * H(C) = -\left(\frac{\text{mean}\,\mathbf{f}(C)}{|B|}\right)^{-\ln \frac{\text{mean}\,\mathbf{f}(C)}{|B|}} \sum_{k=1}^{K} p_k \log p_k$$

B--Clustering RRH complémentaire
Dans cette phase, notre objectif est de regrouper les RRH avec des schémas de trafic complémentaires à un ensemble de BBU dans un pool.
Une méthode intuitive consiste à rechercher de manière exhaustive les RRH avec des modèles de trafic complémentaires et à les regrouper de manière itérative. Cependant, étant donné qu'il existe un nombre énorme de schémas de clustering, une telle méthode peut être exploitable par ordinateur à mesure que l'échelle du réseau augmente. De plus, la distance entre les RRH et le pool de BBU devrait également être limitée dans une plage, car le délai de propagation entre les RRH et le pool de BBU peut dépasser les exigences de qualité de service à mesure que la distance augmente, et nous devons également permettre les communications machine à machine entre RRH comme le transfert (Tekinay et Jabbari, 1991) dans le réseau mobile.
Par conséquent, nous proposons un algorithme basé sur un modèle de graphe pour regrouper efficacement les RRH voisins à la même BBU sous contraintes de distance.
Premièrement, nous construisons un modèle de graphe pondéré pour représenter la relation des RRH, en exploitant des liens de graphe pour exprimer les contraintes de distance RRH, et des poids de lien pour caractériser la mesure de complémentarité RRH.
Ensuite, nous proposons un algorithme basé sur la détection communautaire pour regrouper de manière itérative les RRH en grappes, de sorte que la complémentarité des RRH soit maximisée au sein de chaque grappe et minimisée sur différentes grappes.

B.I--Modélisation RRH basée sur un graphe pondéré

Nous modélisons la complémentarité entre les RRH sous la forme d'un graphique pondéré non dirigé G = (V, E), où V = {r1, ..., rN} désigne l'ensemble des N RRHs et E désigne l'ensemble des liens entre deux RRH.

Nous définissons ensuite la matrice d'adjacence A du graphe G, qui est une matrice symétrique N × N avec les entrées ai, j = 1 lorsqu'il existe un lien entre RRH ri et RRH rj, et ai,j = 0 dans le cas contraire (i, j = 1, .. ., N) .Utilisez la distance géographique de deux RRH pour déterminer s'ils sont adjacents ou non. Plus précisément, pour RRH ri et RRH rj, nous définissons : $a_{i,j} = \begin{cases} 1, \text{ si } dist(ri,rj) \leq T \\ 0, \text{ sinon} \end{cases}$

où dist (ri, rj) est la distance géographique entre les deux RRH, et T est un seuil de voisinage contrôlant la distance géographique des RRH voisins.

Étant donné deux RRH voisins, nous utilisons leurs mesures de complémentarité pour déterminer leurs poids de liaison, c'est-à-dire :

$w(r_i, r_j) = M(\{r_i, r_j\}) * a_{i,j}$

Nous considérons le cas des poids positifs symétriques normalisés (w (ri, rj) ∈ [0, 1]) sans boucles (w (ri, ri) = 0). On note que w (ri, rj) = 0 lorsqu'il n'y a pas de lien entre ri et rj (ai, j = 0).

B.II--Regroupement RRH à contrainte de distance

Dans cette étape, nous devons regrouper les RRH en une BBU, de sorte que chaque cluster se compose de RRH voisins avec des modèles de trafic complémentaires. Comme le poids de liaison du graphique G code la complémentarité des RRH, nous devons regrouper les RRH avec des poids de liaison élevés, ce qui peut être identifié comme un problème de détection communautaire (Newman et Girvan, 2004).

Problème : Étant donné le graphique G = (V, E), nous définissons d'abord un ensemble de clusters P= {C1, ..., CK}, où ∪Ck = V et ∩Ck = ∅.

Puis, étant donné un RRH v, nous définissons la connectivité de v à un cluster C comme somme des poids de liaison entre v et les RRH dans le cluster C : con (v, C) = $\sum w_{v,v'}$.

Enfin, nous définissons les grappes adjacentes CC(v) de v comme CC(v) = {C, con (v, C)> 0, C∈P} Regroupement RRH à contrainte de distance

Avec la définition ci-dessus, notre objectif est de trouver un ensemble optimal de clusters P, de sorte que la connectivité interne au sein d'un cluster soit supérieure à la connectivité inter-cluster, c'est-à-dire,

∀ v ∈ Ck, con (v, Ck) ≥ max {con (v, Cl), Cl ∈ P} Nous devons également délimiter la plage de distance d'un cluster dans le seuil de voisinage, c-à-d : ∀ v, v′∈ Ck, dist (v, v ′) ≤ T

Solution : Sur la base du concept de propagation des étiquettes (Chen et al., 2016a; Raghavan et al., 2007), nous proposons un algorithme DCCA (Distance-Constrained Complementarity-Aware) pour regrouper les RRH. L'idée de base de DCCA est l'attribution itérative des RRH aux clusters adjacents, où le gain d'affectation de RRH v au cluster C est évalué itérativement par une fonction de valeur comme suit :

$$value(v, C) = con(v, C) \times \log\left(\frac{\tau}{\max\{dist(v, v')\}}\right)$$

L'algorithme DCCA attribue goulûment(greedily) les RRH au cluster adjacent avec la valeur la plus élevée jusqu'à ce qu'aucun des RRH ne soit déplacé parmi les clusters (Raghavan et al., 2007). Comme la convergence d'une telle approche gourmande est difficile à prouver, nous fixons un nombre d'itération maximum max _iter pour nous assurer que l'algorithme s'arrêtera.

Algorithme : l'algorithme DCCA est initialisé en affectant chaque RRH du graphique à une étiquette de cluster unique. Dans chaque itération, nous remplissons au hasard une liste de RRH L, et parcourons la liste pour mettre à jour l'étiquette de cluster de chaque RRH. Le processus de mise à jour des étiquettes est le suivant.

Tout d'abord, nous supprimons le RRH de son cluster actuel et trouvons l'ensemble des clusters adjacents au RRH actuel.

Ensuite, nous calculons la fonction de valeur pour tous les clusters adjacents et attribuons le RRH au cluster avec la valeur la plus élevée. Nous marquons le RRH comme déplacé parmi les clusters si son nouveau label de cluster est différent de l'ancien. Après avoir terminé l'itération sur la liste RRH, nous décidons d'effectuer une autre itération ou de terminer l'algorithme sur la base des critères d'arrêt suivants : (1) le nombre d'itérations maximal spécifié max_iter est atteint, ou (2) aucun des RRH n'est déplacé parmi les grappes.

4. Evaluation :

Dans cette section, sur la base d'un ensemble de données de trafic de réseau mobile du monde réel, nous évaluons les performances de notre infrastructure en évaluant sa capacité à réduire les coûts de déploiement et la consommation d'énergie. Nous décrivons d'abord les paramètres de l'expérience, puis présentons les résultats de l'évaluation et les études de cas.

Paramètres du test

Ensembles de données : L'ensemble de données Telecom Italia Big Data Challenge contient deux mois de données de trafic réseau du 11/01/2013 au 31/12/2013 à Milan et Trentin, en Italie, respectivement. La ville de Milan est divisée en grilles de 100 × 100 avec une taille de grille d'environ 235 × 235 mètres carrés, tandis que la province du Trentin est divisée en 117 × 98 grilles avec une taille de grille d'environ 1 000 × 1 000 mètres carrés. Dans chaque grille, le volume de trafic est enregistré sur une base horaire. Nous compilons un ensemble de données de station de base à partir de CellMapper.net, qui comprend les emplacements et les zones de couverture des stations de base actives observés au cours des deux mois. En fonction de l'emplacement et de la couverture de chaque station de base, nous trouvons les grilles couvertes correspondantes et calculons leur volume de trafic. Enfin, nous normalisons les volumes de trafic de chaque station de base à la plage [0, 1] pour la commodité de l'analyse. Les détails de ces deux ensembles de données sont répertoriés dans le tableau 1.

Capacité BBU : Nous déterminons la capacité BBU en fonction du volume de trafic normalisé. Pour l'architecture traditionnelle, nous supposons que chaque RRH est équipé d'une BBU sur site d'une capacité d'un volume de trafic normalisé. De cette façon, le trafic dans chaque RRH peut être couvert par le BBU. Nous définissons la capacité du site comme une unité de capacité. Pour l'architecture C-RAN, nous supposons que les BBU du pool (pool BBU) sont de la même taille et que la capacité est de Q (Q = 1, 2, ...) unité de capacité, de sorte que le trafic de un cluster de trafic RRH peut être traité dans une BBU sans provoquer de surcharge importante. Dans ce travail, basé sur une série d'expériences empiriques, nous choisissons Q = 8 pour la ville de Milan et Q = 10 pour la province du Trentin, respectivement.

Plan d'évaluation : Sur la base des ensembles de données collectés, nous mappons les grilles aux zones de couverture des RRH et agrégons les données de trafic aux RRH correspondantes sur une base horaire. Nous générons ensuite un ensemble de 61 snapshots quotidiens de trafic F, contenant chacun le trafic de 24 h pour l'ensemble des 182 RRH. Nous utilisons les snapshots des premiers 70% comme ensemble d'entraînement Ftrain, et les snapshot des 30% restants comme ensemble de test Ftest. Pour l'ensemble de test, nous calculons la complémentarité des RRH sur la base des prévisions de trafic et construisons un graphique de 182 nœuds avec la structure de lien correspondante basée sur les métriques de

complémentarité. Enfin, nous effectuons l'algorithme DCCA pour regrouper les RRH complémentaires à un ensemble de BBU dans un pool centralisé.

Spécification du modèle : Nous construisons un modèle MuLSTM avec deux couches LSTM empilées. La couche de codeur L1 contient des unités de mémoire Nencoder, qui acceptent un snapshot de trafic de forme [24, 182] en entrée, et sort une séquence codée pour le décodeur. Le décodeur contient des unités de mémoire Ndecoder, qui acceptent la séquence codée en entrée et émettent les prévisions de snapshot du trafic. Nous formons le réseau avec l'ensemble de formation Ftrain pour les itérations Niter afin de nous assurer que le réseau apprend les structures temporelles et spatiales potentielles.

Formation au modèle : Nous utilisons la bibliothèque populaire Tensorflow (Abadi et al.,) Pour construire notre modèle d'apprentissage en profondeur. Sur la base d'une série d'expériences empiriques, nous choisissons le Nencoder = Ndecoder = 32 et le Niter = 10 000 optimaux. Le modèle est formé sur un serveur 64 bits avec une carte graphique NIVIDA GeForce GTX 1080 et 16 Go de RAM. Chaque itération d'entraînement prend environ 3 s et l'ensemble du processus prend 8,3 h.

Mesures d'évaluation : Nous concevons les mesures d'évaluation suivantes pour évaluer respectivement la phase de prévision du trafic RRH et la phase de mise en cluster RRH.

(1) Pour la phase de prévision du trafic RRH, nous comparons la prévision d'instantané de trafic F^i avec les données de vérité terrain Fi dans l'ensemble de test, et calculons l'erreur absolue moyenne (MAE) pour chaque snapshot:

(2) Pour la phase de clustering RRH, nous mesurons quantitativement le gain de multiplexage statistique sous deux aspects, à savoir l'augmentation de l'utilité de la capacité moyenne et la diminution du coût de déploiement global, par rapport aux BBU sur site dans l'architecture traditionnelle. Afin de mesurer l'utilité de capacité d'un schéma de clustering P= {C1, ..., CK}, nous dérivons la métrique suivante basée sur l'équation

(3), c'est-à-dire,
Utilité (P) = mean_{Ck} U(Ck)

sur cette base, nous calculons l'utilité moyenne de la capacité de l'ensemble de test. Afin de mesurer le coût de déploiement global, nous résumons le total des unités de capacité BBU requises dans le pool pour un schéma de clustering i.e., c'est-à-dire:
Coût(P) = $\sum${Ck}
Nous utilisons la quantité maximale d'unités de capacité mesurée dans l'ensemble de test comme coût de déploiement global requis dans le pool.

Méthodes de référence : Nous concevons les méthodes de référence suivantes pour comparer avec la méthode proposée.
• Traditionnel : Dans l'architecture traditionnelle, un RRH est équipé d'un BBU sur site avec une unité de capacité. Les prévisions de trafic et le clustering RRH ne sont pas nécessaires et ne sont donc pas effectués.
• ARIMA-DCCA : cette méthode de base utilise le modèle ARIMA traditionnel pour la prévision du trafic RRH, un RRH à la fois, puis utilise l'algorithme GCLP proposé pour le clustering RRH.
• WANN-DCCA : cette méthode de référence utilise un modèle ANN fenêtré pour la prévision du trafic RRH, qui entre un instantané du trafic pour une journée et génère un instantané du trafic pour le lendemain. L'algorithme de clustering RRH est le même que la méthode proposée.
• MuLSTM-DC : cette méthode de base utilise le modèle MuLSTM proposé pour la prévision du trafic RRH, puis utilise un algorithme de clustering à contrainte de distance (DC) qui regroupe les RRH voisins sans tenir compte de leur complémentarité de trafic. Les étapes de regroupement sont similaires à la méthode DCCA proposée.

Résultats de l'évaluation

Résultats globaux : Le tableau 2 montre les résultats globaux de l'évaluation de la méthode proposée ainsi que les méthodes de référence. Pour la précision des prévisions de trafic RRH, nous pouvons voir que le modèle Mu-LSTM proposé atteint le score d'erreur absolu moyen le plus bas (0,074 à Milan et 0,083 à Trentin) par rapport aux deux lignes de base (ARIMA et WANN), validant sa capacité à modéliser le la dépendance temporelle et la corrélation spatiale du trafic RRH et faire des prévisions précises. En revanche, la méthode ARIMA ne capture pas la corrélation spatiale entre les RRH, tandis que la méthode WANN n'est pas capable de modéliser la dépendance temporelle des modèles de trafic RRH. Par conséquent, les deux niveaux de référence ont un taux d'erreur de prévision plus élevé dans les deux ensembles de données.
Pour les résultats du cluster RRH, la méthode proposée atteint systématiquement la capacité de service moyenne la plus élevée (83,4% à Milan et 76,7% à Trentino), ainsi que le coût de déploiement global le plus bas (88 unités de capacité à Milan et 270 unités de capacité à Trentino). Par rapport à l'architecture traditionnelle avec BBU sur site, les schémas de clustering font passer le taux d'utilité moyen des capacités de 38,8% à 83,4% et réduisent le coût de déploiement global de 182 unités de capacité à 88 unités de capacité (48,4% du coût d'origine) en Milan, validant la possibilité d'obtenir un gain de multiplexage statistique significatif grâce à l'optimisation C-RAN. En comparaison, la ligne de base de clustering à contraintes de distance (MuLSTM-DC) ne tient pas compte de la complémentarité du trafic RRH dans le processus d'optimisation, et n'est donc pas en mesure d'augmenter l'utilité de la capacité et de réduire les coûts de déploiement aussi efficaces que la méthode proposée. En raison de résultats de prévisions de trafic inexacts, les méthodes de référence ARIMA-DCCA et WANN-DCCA ont tendance à produire des schémas de regroupement sous-optimaux et donc à obtenir un gain multiple statistique inférieur.
Nous notons également que notre méthode fonctionne mieux dans la ville de Milan que dans la province du Trentin, ce qui peut s'expliquer par la caractéristique géographique du Trentin. Plus précisément, le Trentin est une région montagneuse où les villes et les villages se dispersent entre les vallées. Les RRH sont dispersés à distance, ce qui rend difficile la formation de grappes RRH complémentaires dans leurs quartiers. En revanche, les zones métropolitaines de Milan est plus grand, plus concentré et plus peuplé, ce qui facilite la formation de clusters complémentaires pour l'optimisation C-RAN.

Études de cas: Nous menons quelques études de cas à Milan pour montrer l'efficacité de notre méthode. Pour la prévision du trafic RRH, la Fig. 5 montre un exemple illustratif des résultats de prévision en utilisant la méthode MuLSTM proposée ainsi que les méthodes de référence ARIMA et WANN. Nous pouvons voir que notre méthode prévoit avec précision les modèles de trafic en semaine et le week-end en fonction de la dépendance temporelle et de la corrélation spatiale qu'elle apprend de l'ensemble d'entraînement. Au lieu de cela, la méthode ARIMA ne parvient pas à apprendre les modèles de dépendance temporelle hybride et génère les prévisions de trafic moyennes. La méthode WANN est capable d'apprendre une certaine dépendance temporelle cachée à partir des données RRH uniques mais n'est pas stable (par exemple, vendredi et samedi).
La figure 6 montre le schéma de clustering RRH avec la méthode proposée le 25/11/2013 (lundi) à Milan. En général, nous obtenons 12 grappes RRH, chacune connectée à une BBU dans le pool centralisé. Sur la figure 6a, nous pouvons voir que de nombreux clusters (par exemple, les clusters A, B et C) sont composés d'une partie urbaine et d'une partie suburbaine, indiquant que les modèles de trafic dans ces zones sont potentiellement complémentaires pendant un jour de semaine typique. Nous notons également que le cluster D est concentré dans une zone relativement petite, ce qui indique la diversité des modèles de trafic dans cette zone (figure 6b). La raison est probablement due aux fonctions hybrides de cette zone, qui se compose d'un grand quartier résidentiel (le quartier de Washington), de plusieurs musées et théâtres nationaux (par exemple, Museo Nazionale

Scienza e Tecnologia Leonardo da Vinci et Teatro Nazionale CheBanca), et d'un carrefour de transport composé de plusieurs gares et stations de métro (par exemple, Milano Porta Genova et Milano Cadorna). L'algorithme est capable d'identifier les RRH avec des schémas de trafic complémentaires pendant la journée et de les regrouper efficacement en une BBU pour obtenir un gain de multiplexage statistique.