



(12)发明专利申请

(10)申请公布号 CN 111698695 A

(43)申请公布日 2020.09.22

(21)申请号 201910348004.1

(22)申请日 2019.04.28

(71)申请人 南京恩瑞特实业有限公司

地址 210039 江苏省南京市江宁区将军大道39号

(72)发明人 马建婷 余锦军 王昊 王恒

(74)专利代理机构 南京知识律师事务所 32207

代理人 高骄阳

(51)Int.Cl.

H04W 16/22(2009.01)

H04W 64/00(2009.01)

权利要求书2页 说明书5页 附图4页

(54)发明名称

一种基于神经网络的LTE指纹式定位方法

(57)摘要

本发明公开了一种基于神经网络的LTE指纹式定位方法,通过LTE系统并结合神经网络算法处理的室内指纹定位来实现更精确的定位策略。本发明克服了基于指纹定位的定位点仅为离散点的弊端,同时也改善了传统三点定位因传输时间等造成的定位误差。包括一下步骤:(1)数据收集阶段:使用USRP硬件设备收集数据;(2)神经网络训练数据阶段:根据神经网络模型,建立数据训练网络并进行训练,建立数据指纹库;(3)在线匹配阶段:根据用户实际数据向量特征,利用神经网络进行预测。

1. 一种基于神经网络的LTE指纹式定位方法,其特征在于,包括以下内容:

步骤1:数据收集阶段:使用USRP硬件设备收集数据;

步骤2:神经网络训练数据阶段:根据神经网络模型,建立数据训练网络并进行训练,建立数据指纹库;

步骤3:匹配阶段:根据用户实际数据向量特征,利用神经网络进行预测。

2. 根据权利要求1所述的一种基于神经网络的LTE指纹式定位方法,其特征在于,所述步骤1对数据的收集处理,分为一下步骤:

步骤1-1:利用USRP硬件设备搭建SDR LTE平台;

步骤1-2:先对LTE的下行链路进行解码获取LTE的上行信号;

步骤1-3:将需要定位的区域等间隔划分为多个小区域,并将小区域依次编号为1,2,3...N,这个编号可以使我们检索到每一个区域的位置;使用USRP信号源在每一个小区域中多次采集信号,记录信号强度,并与相应的位置即小区域编号构成向量数据为:(信号强度<sub>1</sub>,信号强度<sub>2</sub>,...,信号强度<sub>N</sub>,小区域编号);将所采集到的向量数据存入MySQL数据库中。

3. 根据权利要求1所述的一种基于神经网络的LTE指纹式定位方法,其特征在于,所述步骤2对网络的训练,分为一下步骤:

步骤2-1:首先是搭建神经网络:使用Python语言以及TensorFlow机器学习支持库搭建一个8层的神经网络结构;神经网络包含1个数据输入层与1个数据输出层,中间的6层隐藏层;为每一层均构建需要的激励函数为sigmoid函数,选择损失函数为欧式距离函数,选择Adam作为训练中的优化器;

步骤2-2:为训练神经网络做准备:将每一层网络的系数 $\theta$ 随机初始化;将预先采集到的数据进行正则化处理,将数据的存储顺序进行随机打乱,并取百分之八十的数据作为训练数据,百分之二十的数据作为测试数据;

步骤2-3:神经网络的训练:将预先采集到的向量数据输入到构建好的神经网络结构中训练;训练时使用分批优化,每一批训练量batch大小可根据总的训练数据量进行调节,这里设置为100;预设训练步骤为5000步,那么训练结束后,将得到一个完整调谐的神经网络结构。

4. 根据权利要求1所述的一种基于神经网络的LTE指纹式定位方法,其特征在于,所述步骤3对用户位置的指纹匹配,分为一下步骤:

步骤3-1:收集测试用户的相关指纹信息,并进行训练神经网络时的数据正则化处理,并将数据存储顺序进行随机打乱;

步骤3-2:将处理好的测试数据输入到步骤2训练好的神经网络中,神经网络将会计算出这个数据的位置标签,也就是对应到某一个小区域的编号;由于在采集阶段将每一个小区域进行了编号,那么现在也能够根据编号得出输入数据具体在哪一个小区域中,也就得出了其位置信息。

5. 根据权利要求3所述的一种基于神经网络的LTE指纹式定位方法,其特征在于,所述步骤2-2具体包括以下内容:

系统开始训练时,随机地初始化参数 $\theta$ ,根据输入数据 $x$ , $x$ 的正确的标签为 $y$ ,神经网络能够得到一个输出 $\hat{y}$ 。在此时系统并未拟合数据,导致 $\hat{y}$ 与 $y$ 之间存在差值,将其定义为损失

函数 $L(y, f(x, \theta))$ ;

损失函数可以有多种选择,这里使用了平方损失函数,其定义为下式: $L(y, \hat{y}) = (y - \hat{y})^2 = (y - f(x, \theta))^2$ ,其中 $f(\cdot)$ 称为激励函数,采用 $\tanh(x) = 2\text{sigmoid}(2x) - 1$ ,及 $f(x) = \frac{1}{1+e^{-x}}$ ,其中 $e$ 为自然对数;

系统通过损失函数进行多次迭代并更新参数 $\theta$ ,系统参数 $\theta$ 的更新式为: $\theta_{t+1} = \theta_t - \lambda \Delta f(\theta)$ ,其中 $\lambda$ 是参数更新的步长因子, $t$ 代表第 $t$ 次训练, $\theta_t$ 代表第 $t$ 次训练的 $\theta$ 值, $\theta_{t+1}$ 则代表 $\theta_t$ 经过一步更新之后的值,即第 $(t+1)$ 步的 $\theta$ 值, $\Delta f(\theta)$ 表示函数 $f$ 在 $\theta$ 处的导数。步长因子 $\lambda$ 影响着训练过程;因此在机器学习中采用自适应调整学习,即 $\theta_{t+1} = \theta_t - \frac{\lambda}{\sqrt{\sum_{\tau=1}^t g_{\tau}^2}} g_{\tau}$ ,其中 $g_{\tau}$ 是第 $\tau$ 次的学习率,即 $g_{\tau} = \Delta f(\theta_{\tau})$ 。

## 一种基于神经网络的LTE指纹式定位方法

### 技术领域

[0001] 本发明公开了一种基于神经网络的LTE指纹式定位方法,主要用于解决在室内环境下结合LTE系统与指纹方式对用户进行定位的问题。本发明涉及指纹匹配和神经网络技术领域,是一种在室内环境中利用LTE信号进行较高精确的定位的方法。

### 背景技术

[0002] 无线用户的位置信息对于通信服务有着重要的影响,因而定位成为第四代通信系统(4G)与第五代通信系统(5G)的重要研究内容,例如LBS(基于位置信息的服务)、5G毫米波波束赋形(5G BeamForming)、公共安全的紧急服务等。根据用户所处的环境,定位又被分为室外定位与室内定位。在第四代通信系统(4G)中主要适用全球卫星定位系统(GPS)来提供位置服务,然而这种系统由于室内环境下无法接收到GPS信号而失效。为了解决GPS定位系统的这一问题,人们研究了许多本地化定位技术作为其补充,这些技术包括基于RSS(接收信号强度)、CSI(信道状态指示)、AoA(信号到达角度)、ToF(信号到达时间)、TDOA(信号到达时差)等。在这些方法中,TDOA与AOA主要依赖于来自目标的LOS(视距)信号,因而在复杂传播路径的情况系统往往无法避免较大的误差,尤其是在室内环境中。同时,TDOA对定位系统的时间精确度要求很高,例如某个测量设备的时间误差达到1us,则计算信号从移动设备到接收设备的距离为飞行时间乘以光速,误差将达到300m,这将几乎使得定位结果失效。同样的,AOA的方法需要测量方设计复杂的天线,制作与控制复杂度均较高。这些技术之中,基于无线信号的接收强度(RSS)的定位无需复杂的硬件支持,以及在信号非视线传播下良好的性能而成为具有优势的一种室内定位技术。

[0003] 基于RSS的定位技术一般又分为两类,包括基于信道衰落模型的定位与基于指纹数据匹配的定位。由于无线信号的传播受到距离变化与障碍物的影响而不断衰减和变化,RSS值与信号发射端(移动设备)到接收端之间的距离具有很强的相关性,所以RSS可以用来估计发射端和接收端之间的距离。在以信道衰落模型的定位中,建模无线信号的信道衰弱模型,然后估计出发射端和接收端之间的距离,然后利用三角定位方法实现定位。这种方法的局限性是在室内环境下难以构建准确的信道衰落模型而引入误差,当室内环境往往不相同,无法使用统一的模型建模多个地点,不具有通用性并且模型构建复杂。指纹式定位方法则利用数据匹配的思想,通过已知位置的数据来计算未知节点的位置。这里,指纹被定义为与无线信号及其信号源位置。指纹式定位主要包括两个步骤,一是离线阶段,二是在线阶段。离线阶段中采集多个地点的位置信号,存储起来,在在线阶段进行数据匹配从而得到未知信号的信号源位置。数据匹配采用的传统方法是计算离线数据与在线数据的相似程度(如向量距离等)或使用最大似然估计来计算,这些方法往往容易受到数据噪声或测量误差的影响而无法达到较好的精度要求。近年来机器学习方法的发展为这个问题带来了新的解决方案,机器学习方法选取数据的一些特征(例如接收信号强度)训练数据模型,在系统要计算未知用户的位置时,将其特征信号输入到模型中就可以得到其位置信息。在指纹式定位的数据匹配过程中,通常使用的匹配算法有神经网络算法(NN)、最近邻算法(KNN)、支持向

量机 (SVM) 等。这些算法中,神经网络能够提取数据的各方面特征,并进行更好的数据匹配,训练得到的模型能一定程度上模拟信道衰落模型。

## 发明内容

[0004] 本发明是使用LTE信号并结合神经网络模型进行室内用户定位的方法,主要解决了基于指纹定位的定位点仅为离散点的问题,同时也改善了传统三点式定位方法由于时间敏感性高等原因造成的定位误差,提高了定位精度。

[0005] 为了解决以上问题本发明采用了如下技术方案:一种基于神经网络的LTE指纹式定位方法,其特征在于,包括以下内容:

[0006] 步骤1:数据收集阶段:使用USRP硬件设备收集数据;

[0007] 步骤2:神经网络训练数据阶段:根据神经网络模型,建立数据训练网络并进行训练,建立数据指纹库;

[0008] 步骤3:匹配阶段:根据用户实际数据向量特征,利用神经网络进行预测。

[0009] 所述步骤1对数据的收集处理,分为一下步骤:

[0010] 步骤1-1:利用USRP硬件设备搭建SDR LTE平台;

[0011] 步骤1-2:先对LTE的下行链路进行解码获取LTE的上行信号;

[0012] 步骤1-3:将需要定位的区域等间隔划分为多个小区域,并将小区域依次编号为1, 2, 3... (这个编号可以使我们检索到每一个区域的位置)。使用USRP信号源在每一个小区域中多次采集信号,记录信号强度,并与相应的位置(这里是小区域的编号)构成向量数据为:(信号强度<sub>1</sub>, 信号强度<sub>2</sub>, ..., 信号强度<sub>N</sub>, 小区域编号)。将所采集到的向量数据存入 MySQL数据库中。

[0013] 所述步骤2对网络的训练,分为一下步骤:

[0014] 步骤2-1:首先是搭建神经网络。使用Python语言(一种编程语言)以及TensorFlow (TensorFlow是当前主流的一种用于机器学习的支持库)搭建一个8层的神经网络结构。神经网络包含1个数据输入层与1个数据输出层,中间的6层隐藏层。为每一层均构建需要的激励函数为sigmoid函数,选择损失函数为欧式距离函数,选择Adam作为训练中的优化器。

[0015] 步骤2-2:为训练神经网络做准备。选择Sigmoid函数作为各层的激励函数,并将每一层网络的系数 $\theta$ 随机初始化;将预先采集到的数据进行正则化处理,将数据存储顺序做随机打乱,并取百分之八十的数据作为训练数据,百分之二十的数据作为测试数据。

[0016] 步骤2-3:神经网络的训练。将预先采集到的向量数据输入到构建好的神经网络结构中训练。训练时使用分批优化, batch大小(每一批训练量)可根据总的训练数据量进行调节,这里设置为100。预设训练步骤为5000步,那么训练结束后,将得到一个完整调谐的神经网络结构。

[0017] 所述步骤3对用户位置的指纹匹配,分为一下步骤:

[0018] 步骤3-1:收集测试用户的无线信号强度信息,并做与训练阶段相同的数据预处理,这里是对数据进行正则化处理。

[0019] 步骤3-2:将处理好的测试数据输入到(步骤2-3中)训练好的神经网络中,神经网络将会计算出这个数据的位置标签,也就是对应到某一个小区域的编号。由于在采集阶段将每一个小区域进行了编号,那么现在也能够根据编号得出输入数据具体在哪一个小区域

中,也就得出了其位置信息。

[0020] 所述步骤2-2具体包括以下内容:

[0021] 系统开始训练时,随机地初始化参数 $\theta$ ,根据输入数据 $x$ , $x$ 的正确的标签为 $y$ ,神经网络能够得到一个输出 $\hat{y}$ 。在此时系统并未拟合数据,导致 $\hat{y}$ 与 $y$ 之间存在差值,将其定义为损失函数 $L(y, f(x, \theta))$ ;损失函数可以有多种选择,这里使用了平方损失函数,其定义为下式: $L(y, \hat{y}) = (y - \hat{y})^2 = (y - f(x, \theta))^2$ ,其中 $f(\cdot)$ 称为激励函数,采用 $\tanh(x) = 2\text{sigmoid}(2x) - 1$ ,及 $f(x) = \frac{1}{1+e^{-x}}$ ( $e$ 为自然对数);系统通过损失函数进行多次迭代并更新参数 $\theta$ ,系统参数 $\theta$ 的更新式为: $\theta_{t+1} = \theta_t - \lambda \Delta f(\theta)$ ,其中 $\lambda$ 是参数更新的步长因子, $t$ 代表第 $t$ 次训练, $\theta_t$ 代表第 $t$ 次训练的 $\theta$ 值, $\theta_{t+1}$ 则代表 $\theta_t$ 经过一步更新之后的值,即第 $(t+1)$ 步的 $\theta$ 值, $\Delta f(\theta)$ 表示函数 $f$ 在 $\theta$ 处的导数。步长因子 $\lambda$ 影响着训练过程;因此在机器学习中采用自适应调整学习,即 $\theta_{t+1} = \theta_t - \frac{\lambda}{\sqrt{\sum_{\tau=1}^t g_{\tau}^2}} g_{\tau}$ ,其中 $g_{\tau}$ 是第 $\tau$ 次的学习率,即 $g_{\tau} = \Delta f(\theta_{\tau})$ 。

[0022] 有益效果:由于本发明采用了基于神经网络的LTE用户定位系统,起到了通过预先采集数据预测用户位置作用,从而产生了提高用户定位精度的效果。

## 附图说明

[0023] 图1本发明方法的系统流程图;

[0024] 图2 USRP工作流程图;

[0025] 图3链路解码模型;

[0026] 图4指纹定位系统框架;

[0027] 图5神经网络结构框图;

[0028] 图6定位结果模拟图。

## 具体实施方式

[0029] 如图1至6所示,发明实验时采用MATLAB进行数字信号处理,使用USRP(Universal Software Radio Peripheral,通用软件无线电外设)来采集LTE无线电信号,使用官方提供的UHD来实现MATLAB与USRP的连接。

[0030] SDR LTE搭建

[0031] 本发明搭建了SDR LTE平台,所使用的硬件是USRP。USRP是一款可以让计算机像无线电设备一样接收和发送信号的外设,其能实现无线电通讯系统的数字基带和中频的功能。USRP 包含母板和子板,母板上主要有FPGA(Field-Programmable Gate Array,现场可编程门阵列)以及A/D转换器和D/A转换器,母板主要的作用是将采集到的数据发送给计算机,附图 2展示了USRP的工作过程。子板接收信号,子板主要包括低噪声放大器以及接收天线和发送天线,子板接收到空气中的信号之后先经过低噪声放大器进行放大然后通过一次下变频到中频发送给母板,之后母板再通过一次下变频到基带信号发送给计算机。类似的,计算机产生信号数据然后发送给母板,母板进行一次上变频到中频再发送给子板,子板将数据再上变频到射频,然后通过天线发射出去。

[0032] 在LTE系统中,用户要发送数据需要经过基站的预先控制信息,基站的控制信息会

指定用户发送数据的时隙位置以及频率位置。因此,为了获取LTE的上行信号,需要先对LTE的下行链路进行解码。LTE协议中,基站对与用户的资源分配的控制信息是在控制信道发送的,这个控制信道信息并未加密,因而观测者可以利用无线电设备分析基站的控制信息,进而根据解析得到的控制信息来捕获用户的上行信号。信道解码模型如附图3所示。首先使用USRP 进行下行链路解码,可以获得C-RNTI(用户物理层临时标识)和DCI(下行控制信息)以及 SFN(子帧号),利用这些信息可以获取某一特定用户的上行发送信号时间,进而捕获上行链路信号,解码得到每个用户发送的上行链路信号,然后对信号进行分析处理并且计算接收信号强度。

#### [0033] 数据收集

[0034] 本发明实验时采用四台USRP来探测用户的信号,使用MATLAB计算每台USRP的接收信号强度并且记录用户在当前的位置(如附图4所示,图中为了方便采集数据,使用MATLAB和 USRP发送上行链路信号,然后用探测器来接收信号并且采集数据),然后将接收信号强度和位置写入数据库以便从数据库中读取。

#### [0035] 神经网络构建

[0036] 1.使用Python语言(一种编程语言)以及TensorFlow(TensorFlow是当前主流的一种用于机器学习的支持库)搭建一个8层的神经网络结构。神经网络包含1个数据输入层与1个数据输出层,中间的6层隐藏层。为每一层均构建需要的激励函数为sigmoid 函数,选择损失函数为欧式距离函数,选择Adam作为训练中的优化器。

[0037] 2.为训练神经网络做准备。将每一层网络的系数 $\theta$ 随机初始化;将预先采集到的数据进行正则化处理,将数据的存储顺序进行随机打乱,并取百分之八十的数据作为训练数据,百分之二十的数据作为测试数据。

[0038] (1)系统开始训练时,随机地初始化参数 $\theta$ ,根据输入数据 $x$ , $x$ 的正确的标签为 $y$ ,神经网络能够得到一个输出 $\hat{y}$ 。在此时系统并未拟合数据,导致 $\hat{y}$ 与 $y$ 之间存在差值,将其定义为损失函数 $L(y, f(x, \theta))$ ;

[0039] (2)损失函数可以有多种选择,这里使用了平方损失函数,其定义为下式: $L(y, \hat{y}) = (y - \hat{y})^2 = (y - f(x, \theta))^2$ ,其中 $f(\cdot)$ 称为激励函数,采用 $\tanh(x) = 2\text{sigmoid}(2x) - 1$ ,及 $f(x) = \frac{1}{1+e^{-x}}$ ( $e$ 为自然对数);

[0040] (3)系统通过损失函数进行多次迭代并更新参数 $\theta$ ,系统参数 $\theta$ 的更新式为: $\theta_{t+1} = \theta_t - \lambda \Delta f(\theta)$ ,其中 $\lambda$ 是参数更新的步长因子, $t$ 代表第 $t$ 次训练, $\theta_t$ 代表第 $t$ 次训练的 $\theta$ 值, $\theta_{t+1}$ 则代表 $\theta_t$ 经过一步更新之后的值,即第 $(t+1)$ 步的 $\theta$ 值, $\Delta f(\theta)$ 表示函数 $f$ 在 $\theta$ 处的导数。步长

因子 $\lambda$ 影响着训练过程;因此在机器学习中采用自适应调整学习,即 $\theta_{t+1} = \theta_t - \frac{\lambda}{\sqrt{\sum_{\tau=1}^t g_{\tau}^2}} g_{\tau}$ ,其中 $g_{\tau}$ 是第 $\tau$ 次的学习率,即 $g_{\tau} = \Delta f(\theta_{\tau})$ 。

#### [0041] 在线预测定位

[0042] 本次实验时,测试数据是在收集数据的过程中同一个点测试了10帧数据,将采集得到的样本分为训练样本和测试样本。在得到训练样本与测试样本之后,为了防止过拟合,将训练样本和测试样本带上标签之后全部打乱,然后使用训练样本对模型进行训练,以及将测试样本输入到模型中得到预测的结果。最后得到测试样本的预测位置集合,在对集合

中的每一个元素计算与之对应的实际位置坐标之间的欧式距离再对所有的结果取平均作为最终的模型误差。为说明实验效果,将定位模型预测得到的测试点的实际位置和预测位置全部绘制了一个高和宽的比例是3:2的矩形,然后在以实际位置为中心绘制了一个红色的矩形,以预测位置为中心绘制了一个蓝色圆形,最终定位效果如附图6所示。

[0043] 以上所述仅为本发明的优选实施例而已,并不限制于本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的权利要求范围之内。





图1

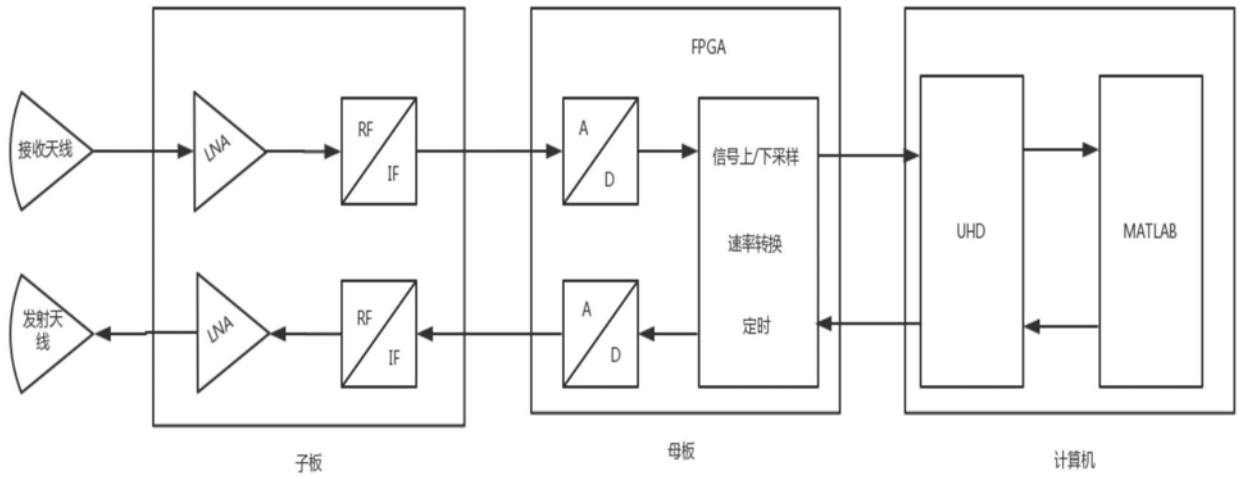


图2

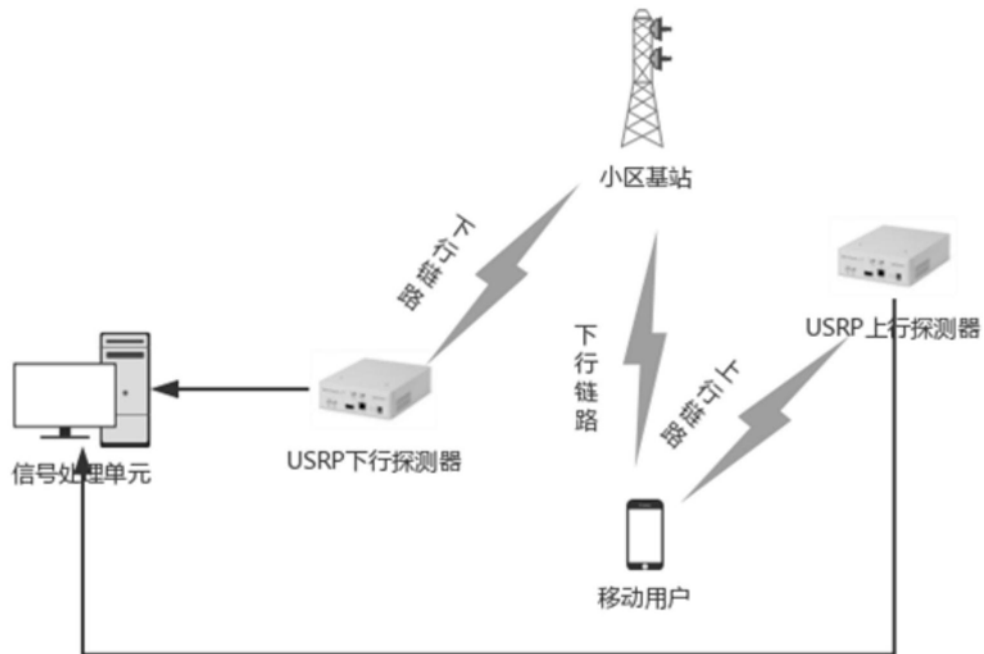


图3

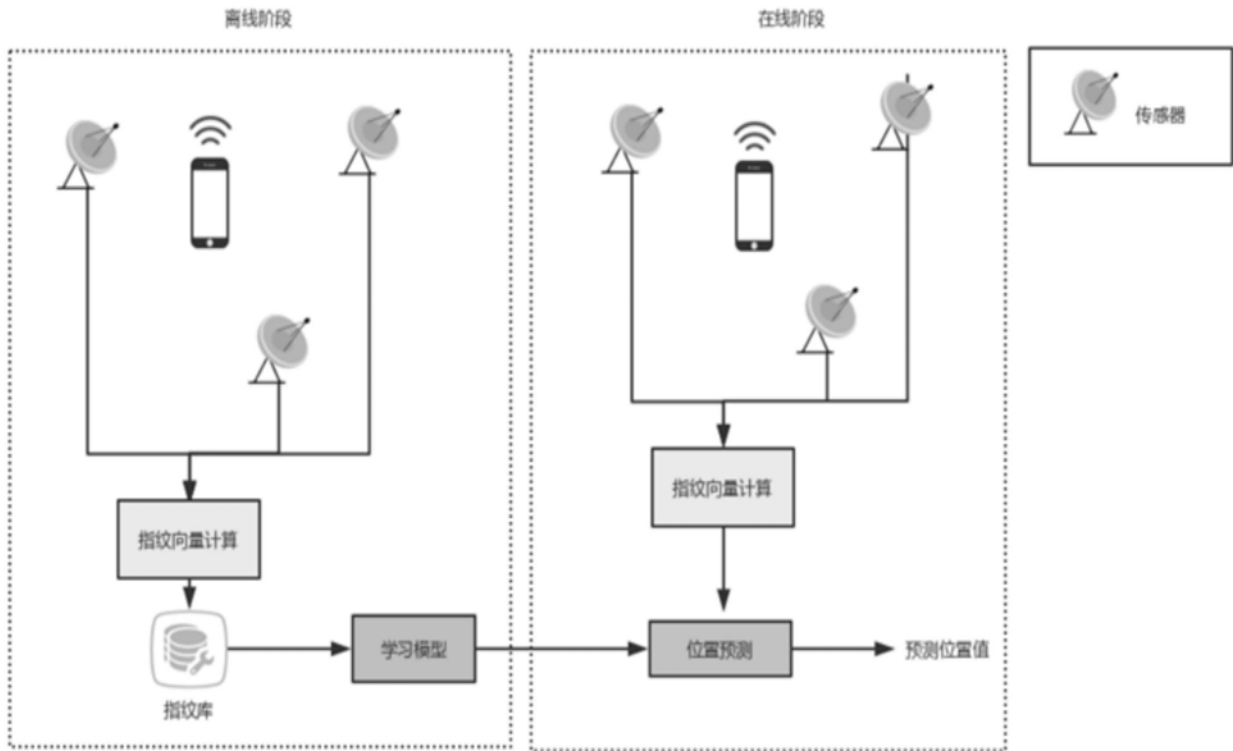


图4

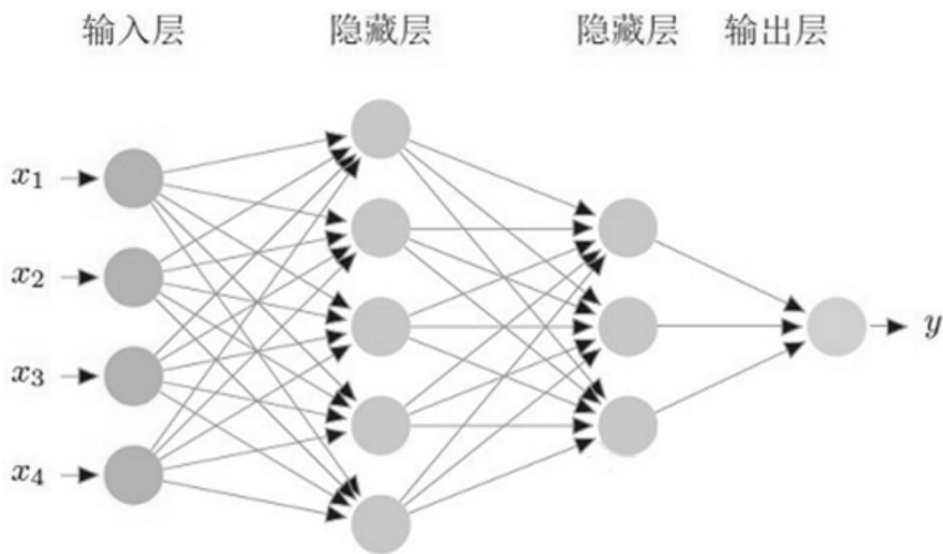


图5

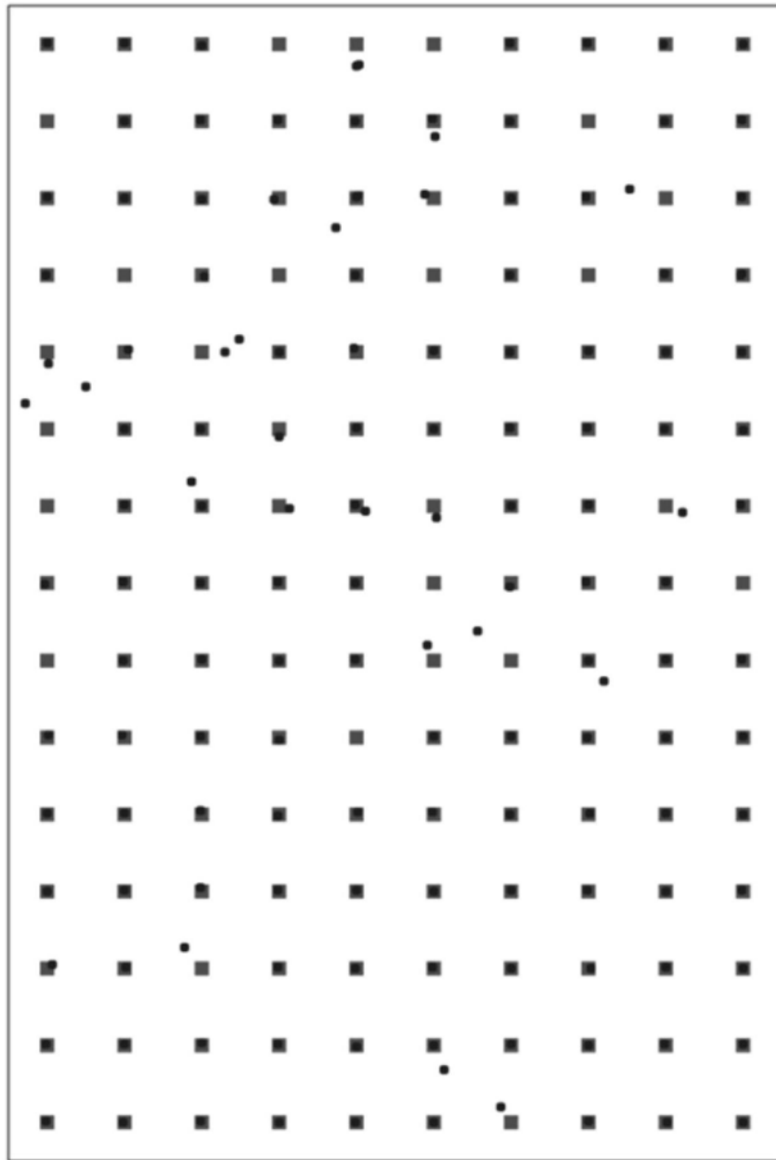


图6