计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200700093

基于动态附加布隆过滤器的 RFID 数据冗余处理算法

段 雯 周 良
南京航空航天大学计算机科学与技术学院 南京 210016
(duanwen076@163.com)

摘 要 针对 RFID 设备在读取标签信息时产生的高度冗余会造成实时传输压力、存储空间浪费和上层应用分析结果不可靠等问题,提出一种动态附加布隆过滤器算法(Dynamic-Additional Bloom Filter,DATRBF)来清除 RFID 冗余数据。首先结合 RFID 动态数据流特点,利用时间和阅读器因素的影响设计了基础布隆过滤器(Time-Reader Bloom Filter,TRBF),然后根据定时间区间内数据量变化动态决定是否调整或附加额外的 TRBF,通过附加 TRBF 从而扩充数组的方式将误判率控制在阈值内,最后结合两个过滤器对数据是否冗余进行综合判断。实验证明,在过滤 RFID 实时动态数据流中的冗余数据时,DATRBF 算法相比传统布隆过滤器(Bloom Filter,BF)和时空布隆过滤器(Temporal-Spatial Bloom Filter,TSBF)有明显的优势,在数据量随机波动时 DATRBF 的误判率平均约为 TSBF 的 49%,且 DATRBF 算法能够在数据量持续上升时保持平稳的低误判率。

关键词:布隆过滤器;RFID;冗余数据;动态附加;误判率
中图法分类号 TP391

Redundant RFID Data Removing Algorithm Based on Dynamic-additional Bloom Filter

DUAN Wen and ZHOU Liang
School of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 210016,China

Abstract The high redundancy generated by RFID devices in reading tag information will result in pressure of real-time transmission,waste of storage space and unreliable analysis results of upper application. To slove these problems,a dynamic-additional Bloom filter algorithm (DATRBF) is proposed to remove redundant RFID data. Firstly,combining the characteristics of RFID data and considering the influence of time and reader,the basic Bloom filter (TRBF) is designed. Then,it is decided whether to adjust or add additional TRBF dynamically according to the change of data amount in a fixed time interval,and the misjudgment rate is controlled within the threshold by expanding bit array with additional TRBF. Finally,combining the two filters to judge whether the data is redundant or not. The experiment proves that the DATRBF algorithm has obvious advantages over the traditional Bloom filter (BF) and temporal-spatial Bloom filter (TSBF) when filtering the redundant data stream of RFID. When the data amount fluctuates randomly,the misjudgment rate of DATRBF is about 49% of that of TSBF on average,and the DATRBF algorithm can maintain a stable and low misjudgment rate when the data amount continues to rise.

Keywords Bloom filter,RFID,Redundant data,Dynamic additional,Misjudgment rate

1 引言

RFID 技术作为下一代制造系统中关键的先进制造技术,被认为是可提高可视性和效率的最有前景的技术创新之一[1],作为一种非接触式采集处理信息的自动识别技术,它能够有效感知制造车间的动态信息,如监测车间生产加工状态、优化物流仓储控制以及智能定位跟踪货物等。随着物联网技术的发展,在离散制造过程中应用 RFID 技术成为制造业发展的必然趋势[2]。

传感器和 RFID 采集的数据是车间生产环境监控、运行设备状态、车间产品物流监测等相关的信息,这些数据通常有严格的实时性要求。为了保证采集的准确,传感器和 RFID 的采集频率非常高,且为了防止标签移动太快造成数据丢失,会在同一区域部署多个阅读器,这就导致采集的数据规模巨大且质量低下。同时,由于 RFID 的工作方式,同一个标签长时间停留在某一固定读写器的读写范围内会被多次读取,因此产生了大量重复、无效的数据。由于射频干扰以及上述读取原理等多种因素,RFID 原始数据的正确率大约是 60%~70%,这些数据被采集后需要进行实时传输、存储并送到上层应用进行计算分析,如此,大量的冗余数据会造成存储空间的浪费,进而影响上层程序的分析和判断,给企业造成无法预计的后果。

为了过滤 RFID 冗余数据,文献[3-4]从冗余阅读器的层面来考虑。Kamaludin 等[3]采用单布隆过滤器实现了对冗余阅读器的过滤,从而有效消除了冗余数据并避免了因大量 hash 函数导致处理效率低下的问题。Ma 等[4]提出分布式冗

到稿日期:2020-07-14 返修日期:2020-09-21 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
通信作者:周良(betazllj@nuaa.edu.cn)

余阅读器消除算法，根据期望的标签覆盖范围定义的阈值序列来选择有效的阅读器，从而消除由冗余阅读器造成的重复数据。基于滑动窗口的过滤方法只需要维护一个小规模窗口大小的数据就能对流数据进行实时处理。文献[5]针对 RFID 数据流的移动性、流特性和不确定性，设计了一个基于块滑动窗口模型的有概率特性的三级滤波框架对数据进行过滤。文献[6]结合了自适应滑动窗口和欧氏距离的优点对 RFID 不可靠数据进行过滤，并对滤波后的数据进行分流。有学者基于机器学习的背景提出了提高 RFID 数据质量的方法。文献[7]提出了一种利用变分推理技术对 RFID 数据进行清洗的新方法，利用先验知识所采用的应用程序物理约束来提高数据质量。文献[8]提出了一种基于静态贝叶斯网络和动态贝叶斯网络的传感器数据的冗余预处理方法，但是该算法对历史数据的依赖性较强，冗余决策速度较慢。

由于 RFID 数据具有实时性、流式等特点，对数据的过滤必须在有限的空间和时间内进行，因此，使用 BF 算法对 RFID 数据进行过滤受到研究者的广泛关注[9]。Bloom 在 1970 年提出了布隆过滤器(Bloom Filter，BF)算法[10]，BF 由长度为 m 的比特数组和 k 个独立的哈希函数组成，通过 k 个独立的哈希函数将数据映射到比特数组 k 个不同的位置，如果数据 k 个值均为 1，则以一定的误判概率认为该数据曾出现在集合中，若 k 个位置存在一个或多个值为 0，则认为该数据没有出现在集合中，最后将映射的 k 个位置置 1。BF 的误判是假阳性错误，即某个数据没有在集合中出现过也会被判定为冗余数据。近年来，有很多学者对 BF 进行了改进，文献[11]利用改进 BF 的时间和空间高效性来处理访问控制问题。文献[12]在 BF 算法和 Cuckoo filters 算法的基础上提出一种改进的过滤器算法，该算法不仅能够减小插入操作的复杂性，而且支持删除操作。文献[13]基于 BF 改进的 OHBF(One-Hashing Bloom Filter)仅需要一个基哈希函数加上几个简单的模运算就能实现一个过滤器，降低了哈希函数计算的开销。文献[14]提出的自缩放布隆过滤器解决了传统 BF 在可扩展性方面的主要困难。在过滤 RFID 数据方面，文献[15]提出了时间布隆过滤器(Time Bloom Filter，TBF)，通过 RFID 数据的时间属性进行冗余判断，但其仅适用于静态数据，当数据动态变化时，其适用性下降。在 TBF 的基础上，文献[15]又提出了时间间隔布隆过滤器(Time Interval Bloom Filter，TIBF)，该过滤器采用二维数组结构分别存储数据被读取的开始时间和结束时间，通过哈希映射后对应的过滤单元里的开始时间到结束时间这一时间段内是否存在相交时间，来判断冗余数据，如果不存在相交时间，则判断为非冗余数据，否则为冗余数据。文献[16]提出了时空布隆过滤器(Temporal-Spatial Bloom Filter，TSBF)，它能够处理动态 RFID 数据流，消除了假阳性错误，存在少量假阴性错误，但其仅考虑时间对数据的影响。文献[17]基于 TSBF 提出了冗余清洗模型 R-TSBF，将 RSSI 引入冗余判断规则中，在动态处理 RFID 数据流的同时，进一步降低误判率，并保留数据最大强度。文献[18]提出一种名为时间距离布隆过滤器(Time-Distance Bloom Filter，TDBF)的算法，利用时间和 RSSI 两个因素来判断冗余，该算法结合监控场景需求，能够对动态场景中的移动标签从时间和空间两个方面进行冗余过滤。

本文提出了动态附加布隆过滤器算法(Dynamic-Additional Bloom Filter，DATRBF)算法，该算法首先利用时间和分区阅读器对数据冗余的影响在 TSBF 的基础上设计了一个基础 TRBF，并根据设置的时间区间内数据量的变化动态附加 TRBF 单元或调整已有的附加 TRBF，将误判率控制在一定范围内，并在数据量较小时避免空间浪费。

2 DATRBF 算法的设计

2.1 相关定义

综合影响 RFID 实时数据质量的多维因素，下面给出 RFID 数据流冗余的相关定义。

定义 1(RFID 数据流) 用 S 表示实时 RFID 数据流，S 为 n 个元素组成的序列 $\{s_1, s_2, \cdots, s_n\}$，其中每一个元素 s_i ($1 \leqslant i \leqslant n$) 都是一个三元组 $\langle tagid, time, readerid \rangle$，其中 $tagid$ 是标签的唯一标识，$time$ 是读取 RFID 标签的时间，$readerid$ 表示读取标签的阅读器标识。

定义 2(时间冗余) 判断数据 x 为时间冗余，当且仅当存在 y，使得 $x.tagid = y.tagid$，$x.time - y.time \leqslant \tau$ 且 $x.time > y.time$，τ 为设定的时间阈值。

定义 3(阅读器冗余) 判断数据 x 是阅读器冗余，当且仅当存在 y，使得 $x.readerid$ 与 $y.readerid$ 为属于同一划分区域的阅读器编号。

定义 4(误判率) 误判率的计算公式为 $N_f/(N_t+N_f)$，其中 N_t 为正确判断输出的 RFID 数据，N_f 是错误判断输出的 RFID 数据。

2.2 DATRBF 的数据结构

实际应用中，RFID 数据流是持续不断的，随着数据量的增大，传统 BF 中比特数组为 1 的部分越来越多，导致误判率上升，因此传统 BF 不能长时间处理大量 RFID 流数据。文献[16]对传统 BF 进行了改进，提出了 TSBF 算法。TSBF 的数据结构如图 1 所示，其使用二维数组代替比特数组，二维数组分别保存标签号(tagid)和读取时间(time)，以代替传统 BF 比特数组的 0 或 1。二维数组初始化为 0，当新数据 x 到达时，经过 k 个 hash 函数映射后，检查 k 个位置，如果存在 i 使 $x.tagid = TSBF[hi(x.tagid)].tagid$，则表明数据 x 已经出现过，但还不足以表明该数据是冗余的；接着，如果 $x.time$ 与 $TSBF[hi(x.time)].time$ 的差值小于设置的时间阈值，则表示该数据为冗余数据。不管 x 是否为冗余，都将相应的 tagid 和 time 单元更新为 $x.tagid$ 和 $x.time$。数组的内容随着新数据的到来不断更新，而 TSBF 算法能够长时间处理动态 RFID 数据流，且消除了假阳性错误。但 TSBF 存在将冗余的数据误判为非冗余数据的假阴性错误。

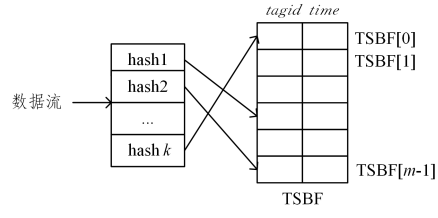

图 1 TSBF 数据结构

Fig. 1 Data structure of TSBF

TSBF 未考虑读取标签的阅读器编号(readerid)对数据冗余的影响，且不论数据量大小，数组的空间都是固定的。DATRBF 算法在 TSBF 的基础上考虑了阅读器编号对冗余数据的影响，对密集分布的阅读器进行区域划分，设计了基础 TRBF。受动态 BF[19] 和可扩展 BF[20] 结构的启发，在设置基础 TRBF 的数组后，我们根据实时数据量的变化动态附加或撤销额外的数组空间，在数据量大的时区内通过附加过滤器来保证误判率在阈值之下，在数据量小的时区内撤销附加过滤器以避免空间浪费。若存在附加 TRBF，则最终结果由两个过滤器综合判断。

DATRBF 的具体数据结构如图 2 所示。基础 TRBF 由一组数量为 k 的哈希函数和一个大小为 m 的三维数组组成。数组初始化为 0，第一维存储标签标识 tagid，第二维存储读取标签的时间 time，第三维存储读取标签的阅读器编号 reader-id，第 i 个单元的数据表示为 $TRBFa[tagid][time][readerid]$。附加 TRBF 的结构与基础 TRBF 一致，第 i 个单元的数据表示为 $TRBFb[tagid][time][readerid]$。其数组大小由超出阈值的数据大小决定。

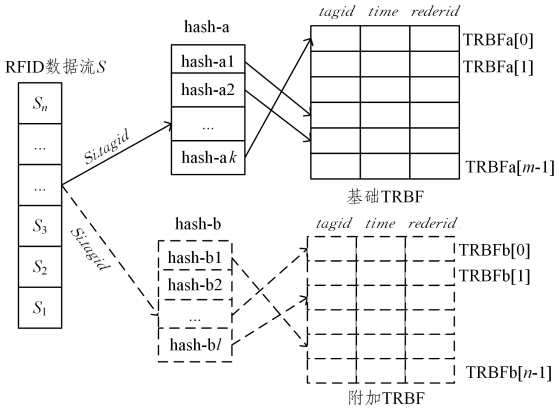

图 2　DATRBF 数据结构

Fig. 2　Data structure of DATRBF

2.3　DATRBF 的判断流程

当某个新的时间区间开始时，RFID 数据 x 到达，冗余判断流程如下：

(1)根据上一个时间区间内统计的数据总量判断是否超过设定的阈值，若超过，则计算出合适的哈希函数个数 l 和数组大小 n，设置附加 TRBF；若不超过，则只使用基础 TRBF，并撤销上一轮时间区间的附加 TRBF。

(2)将 x 的 tagid 进行 k 次独立哈希映射到基础 TRBF 的数组中。

(3)如果基础 TRBF 存在一个单元 i 使得 $TRBFa[hi(x.tagid)].tagid=x.tagid$，$x.time-TRBFa[hi(x.tagid)].time\leqslant\tau(x.time>TRBFa[hi(x.tagid)].time)$，$x.readerid$ 与 $TRBFa[hi(x.tagid)].readerid$ 属于同一个划分区域，则说明数据 x 为冗余数据，删除该数据；否则，如果不存在附加 TRBF，则直接判断数据 x 为非冗余数据，更新基础 TRBF 数组的 k 个单元的标签信息、时间信息和阅读器信息，即 $TRBFa[tagid]=x.tagid$，$TRBFa[time]=x.time$，$TRBFa[readerid]=x.readerid$。

(4)若存在附加 TRBF，则将数据 x 的 tagid 进行 l 次独立哈希映射到附加 TRBF 的数组中。

(5)附加 TRBF 的判断方法与基础 TRBF 一致，如果附加 TRBF 判定数据 x 为冗余数据，则删除该数据；否则，结合基础 TRBF 的判断结果，若两个过滤器均判为非冗余数据，则最终判定数据 x 非冗余。最后更新基础 TRBF 数组 k 个单元的标签信息、时间信息和阅读器信息，即 $TRBFb[tagid]=x.tagid$，$TRBFb[time]=x.time$，$TRBFb[readerid]=x.readerid$。

(6)在这一轮时间区间结束后，根据本轮数据总量判断下一轮是增加或调节过滤器还是撤销已有的附加过滤器。

结合冗余定义和冗余判断流程，对数据 x 判断是否冗余的伪代码如算法 1 所示。

算法 1　DATRBF 算法

```
输入：RFID 数据 x：x.tagid，x.time，x.readerid.
输出：x 是否为冗余数据
1. init flaga=1;
2. init flagb=1;
3. if(TRBFb is not null)
4.    for(i=1;i<=l;i++)
5.       if(TRBFbh[i][tagid]=x.tagid and x.time-TRBFbh[i]
         [Time]<=τand x.time>TRBFbh[i][Time] and x.readerid.
         aera=TRBFbh[i][readerid].aera)
6.            then delete x;
7.            flagb=0;
8.            break;
9.         end if;
10.   end for;
11. end if;
12. for(i=1;i<=k;i++)
13.   if(TRBFah[i][tagid]=x.tagid and x.time-TRBFah[i]
      [Time]<=τ and x.time>TRBFah[i][Time] and x.readerid.
      aera=TRBFah[i][readerid].aera)
14.       then delete x;
15.       flaga=0;
16.       break;
17.    end if;
18.  end for;
19. if(flaga&&flagb=1)
20.     then send x to the event;
21. update TRBFa(x.tagid,x.time,x.readerid);
22. update TRBFb(x.tagid,x.time,x.readerid).
```

2.4　误判率估计

2.4.1　TRBF 误判率估计

由于 TRBF 判断数据冗余的机制是，存在一个单元 i 的数据与待判断的数据同时满足标签号相同、时间冗余和阅读器冗余 3 个条件。TRBF 同 TSBF 一样不会出现假阳性错误，但是会出现某个存在过的数据映射的 k 个位置均被其他标签号改变的情况，导致下次出现同一个数据时被判为非冗

余数据，出现假阴性错误。误判率计算过程如下：

假设过滤器中的每个数据经过哈希函数都能独立地等概率地映射到 m 个比特位中的任何一个，则某一特定单元在一个哈希函数映射时没有被映射到的概率为：

$$1-\frac{1}{m} \tag{1}$$

则 k 个哈希函数都没有映射到此特定位置的概率为：

$$\left(1-\frac{1}{m}\right)^{k} \tag{2}$$

如果 τ 时间内插入了 n 个元素，n 个元素里有 n' 个元素为非冗余数据，只有这些非冗余数据到特定位置才会改变映射位置的标签号，所以 τ 时间内，所有的元素输入都未将一个特定位置标签号改变的概率为：

$$\left(1-\frac{1}{m}\right)^{kn'} \tag{3}$$

因此，τ 时间内插入了 n 个元素，将特定数组位置标签号改变的概率为：

$$1-\left(1-\frac{1}{m}\right)^{kn'} \tag{4}$$

当某个待判断数据的 k 个 hash 映射的位置标签号都被改变，且此数据被判定为非冗余时，出现误判，因此可得出误判率公式为：

$$P(TRBF)=\left(1-\left(1-\frac{1}{m}\right)^{kn'}\right)^{k} \tag{5}$$

2.4.2 DATRBF 误判率估计

TRBF 的误判率会随着系统非冗余数据量的增多而提高，为了将其控制在理想状态，DATRBF 算法根据当前的数据变化，当数据超过设定的阈值时，通过启动并自动调节附加 TRBF 参数来降低误判率。下面介绍附加 TRBF 相关参数的计算。

由于式(5)中的 m 可以视作趋向于无穷，则公式简化为：

$$P(TRBF)=(1-e^{\frac{-kn'}{m}})^{k} \tag{6}$$

如果按照最优条件：$k=\ln 2*\frac{m}{n'}=\ln 2*b$，则式(6)还可以简化为：

$$P(TRBF)=0.6185^{b} \tag{7}$$

其中，$b=\frac{m}{n'}$，表示每一个非冗余数据占的位空间大小，且误判率和 b 成正比。

我们设定系统最大误判率和初始参数如下：

$$P_0(TRBF)=0.6185^{\frac{m_0}{n_0'}}=0.6185^{b_0} \tag{8}$$

其中，m_0 为系统保证最大误判率的位空间大小；n_0' 为系统最大误判率容许的最大非冗余数据大小，也就是非冗余数据阈值。

根据以上公式，如果非冗余数据数量上升，超过阈值 n_0'，那么 b 会减小，从而误判率升高。为了将误判率控制在一定范围内，设置一组附加 TRBF。

假设非冗余数据增长了 n'_{add}，那么此时误判率表示如下：

$$P_1(TRBF)=0.6185^{\frac{m_0}{n_0'+n'_{add}}}=0.6185^{\frac{m_0}{n_1'}}=0.6185^{b_1} \tag{9}$$

其中，n_1' 为数据增长后的非冗余数据数量。

由于附加 TRBF 的 hash 函数完全独立，那么 DATRBF 的误判率计算如下：

$$P_1(DATRBF)=0.6185^{b_1}*0.6185^{b_{add}}=0.6185^{b_1+b_{add}} \tag{10}$$

$$b_{add}=\frac{m_{add}}{n_1'} \tag{11}$$

其中，m_{add} 为附加 TRBF 数组空间大小。

为了使 $P_1(DATRBF)$ 误判率不超过阈值，其值应该等于最大误判率 $P_0(TRBF)$，由式(8)、式(10)、式(11)可知满足下式即可：

$$b_1+b_{add}=b_0 \tag{12}$$

进一步计算可知，附加 TRBF 的数组大小为：

$$m_{add}=\frac{m_0}{n_0'}n_1'-m_0 \tag{13}$$

3 实验结果与分析

本实验采用的硬件环境是 IntelCore i5-6200U CPU，8GB 内存，500GB 硬盘；软件环境为 Microsoft Windows 10，Visual Studio 2017。实验采用的数据为模拟 RFID 三元组(〈*tagid*，*time*，*readerid*〉)数据集，其中每组实验的数据集都设定 50% 的冗余数据，分 5 个时间区间来进行实验，DATRBF 算法的误判率最大阈值设置为 0.1%，取数据量 $n'=40000$。对误报率计算公式(见式(6))求导可知，当 $k=9$，$m=560000$ 时，系统误判率最低，但在实际实验中，hash 函数个数 k 过大会导致系统运算速度过慢，最后综合时空开销选取更合适的 k 值和 m 值，即 $k=6$，$m=631314$，并在实验(1)、实验(2)、实验(4)中均使用此参数。实验还引入传统 BF 算法和 TSBF 算法作为对比方法。

由 DATRBF 的数据结构以及上述公式可知，影响 DATRBF 误判率的因素主要有：三维数组单元长度 m、哈希函数个数 k、数据总量 n 以及定时间区间内的数据量 n'。在实验中，通过改变不同的因素，分析 3 种过滤器误判率的变化情况。

由于 BF 会将非冗余数据误判为冗余数据，冗余数据中存在错误数据，而 TSBF 和 DATRBF 算法会将冗余数据误判为非冗余数据，非冗余数据中存在错误数据。因此，将在相应的实验表格中列出 BF 算法判断出的冗余数据，以及 TSBF 和 DATRBF 算法判断出的非冗余数据。

(1)数据量变化对误判率的影响

为了验证 DATRBF 的有效性，对数据流大小对误判率的影响进行实验。本组实验中，固定数组大小为 631364，哈希函数个数 k 为 6。不同数据流大小时 3 种过滤器判断出的非冗余数量和冗余数量情况如表 1 所列。对应的误判率情况如图 3 所示，其中 BF 的误判率最高，并且随着数据量的增长呈上升趋势。这是由于随着数据流的增大，越来越多的单元被置为 1，因此产生高误判率。TSBF 的误判率相对 BF 小很多，但它仍然有随着数据量的增大而呈上升的趋势。DATRBF 的误判率在所有算法中最低并且随数据量增加的波动最小，

保持较为平稳的趋势。这是由于 DATRBF 算法能够根据数据量变化自适应地将误判率控制在阈值内。该实验证明，DATRBF 在数据量增大时也只有很低的错误率，适合长时间处理 RFID 数据流。

表 1　不同数据总量时的判断情况

Table 1　Judgment of different data amount

数据总量($\times10^5$)	BF 冗余	TSBF 非冗余	DATRBF 非冗余
4	217949	200307	200307
5	288074	250978	250422
6	365390	302367	300646
7	448810	354842	351183
8	537145	409104	401991

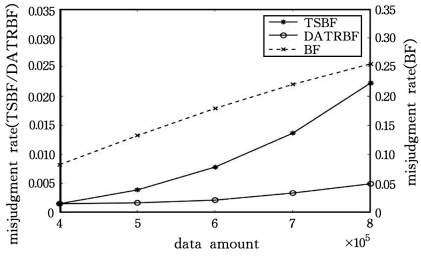

图 3　不同数据量时误判率的比较

Fig. 3　Comparison of misjudgment rate with different data amount

(2)数组大小的变化对误判率的影响。

本组实验中，固定数据量大小为 6×10^5，哈希函数个数 k 为 6。表 2 列出了随数组大小增长时判断出的冗余数量和非冗余数量。图 4 给出了不同数组大小下误判率的变化情况。从图 4 可以看出，当数组较小时，3 种算法的误判率都较高，其中传统 BF 的误判率达到 35%以上，这是由于数据量过大使得数组负载因子过大，误判的概率较大，这种趋势随着数组的增大而下降。通过对比发现，随着数组增大，DATRBF 的误判率能够趋近于 0。

表 2　不同数组大小时的判断情况

Table 2　Judgment of different bit array sizes

数组大小($\times10^5$)	BF 冗余	TSBF 非冗余	DATRBF 非冗余
3	465181	330101	315160
4	426697	312219	304929
5	395598	305637	301797
6	371415	302771	300793
7	353448	301548	300370

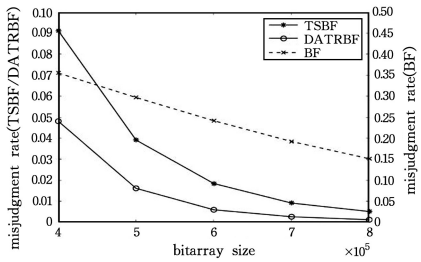

图 4　不同数组大小时误判率的比较

Fig. 4　Comparison of misjudgment rate with different bit array size

(3)哈希函数个数变化对误判率的影响

本组实验中，固定数据量大小为 6×10^4，数组大小为 631364。表 3 列出了哈希函数个数变化时 3 种算法判断出的冗余数量和非冗余数量。图 5 给出了哈希函数个数由少变多时误判率的变化情况。从图 5 可以看出，DATRBF 算法和 TSBF 算法的误判率均比传统 BF 低，随着哈希函数个数的增加，DATRBF 与 TSBF 的区别越来越小，这是由于误判率随着 hash 函数个数的增多而降低，非冗余数据和阈值之间的差值越来越小，附加 TRBF 的效果没有在 hash 函数较少时明显。

表 3　不同的 hash 函数个数的判断情况

Table 3　Judgment of different number of hash functions

hash 个数	BF 冗余	TSBF 非冗余	DATRBF 非冗余
2	228513	254050	235665
3	217129	214666	213986
4	215279	203965	203906
5	215941	201101	201089
6	215039	200307	200287

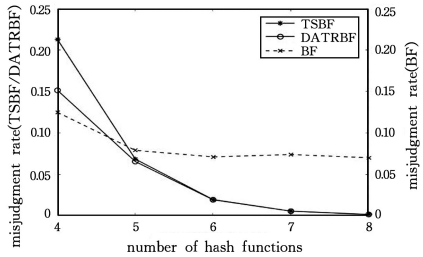

图 5　不同 hash 函数个数时误判率的比较

Fig. 5　Comparison of misjudgment rate with different number of hash functions

(4)数据量波动时误判率的比较

设置 4 个数据量波动的实验场景，对比 DATRBF 和 TSBF 的误判率情况。其中，场景一每轮时间区间内数据量是递增的，场景二每轮时间区间内数据量是递减的，场景三数据量先增加后减小，场景四数据量先减小后增加。如图 6 所示，在数据量波动的情况下，DATRBF 的误判率均比 TSBF 低。这是由于在数据量变化时，DATRBF 能够根据时间区间内数据量的变化自动添加或调节附加 TRBF，避免误判率过大。

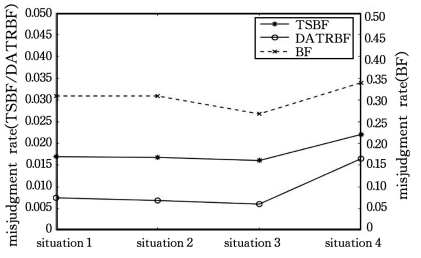

图 6　数据量波动时误判率的比较

Fig. 6　Comparison of misjudgment rate when data amount changes

结束语　实际应用中，RFID 数据流持续到达，短时间内数据量巨大，DATRBF 算法能够快速实时过滤 RFID 数据流中的冗余数据。DATRBF 算法考虑到阅读器对冗余数据的影响，并在数据量较大时动态增加或调整附加 TRBF 以保持低误判率，在数据量减小时撤销附加的 TRBF 以避免空间浪费。实验结果表明，DATRBF 算法具有较好的过滤冗余数据的效果，提高了 RFID 数据的准确性，同时减轻了数据传输和上层应用分析的压力。

本文提出的 DATRBF 算法还存在一些不足。由于

DATRBF 算法在数据量变化时实时调整附加 TRBF 的参数，需要耗费额外的时间，因此如何继续优化 DATRBF 的数据结构和判断流程以提高算法的执行效率是进一步研究的方向。

参 考 文 献

[1] CAO W, JIANG P Y, LU P, et al. Real-time data-driven monitoring in job-shop floor based on radio frequency identification[J]. International Journal of Advanced Manufacturing Technology, 2017, 92(5/6/7/8): 2099-2120.

[2] CAO W, JIANG P Y, JIANG K Y, et al. Radio frequency identification-based real-time data collecting and visual monitoring for discrete manufacturing workshop[J]. Computer Integrated Manufacturing Systems, 2017, 23(2): 273-284.

[3] KAMALUDIN H, MAHDIN H, ABAWWAJY J H, et al. Filtering Redundant Data from RFID Data Streams[J]. Journal of Sensors, 2015, 2016: 1-7.

[4] MA M, WANG P, CHU C H. Redundant Reader Elimination in Large-Scale Distributed RFID Networks[J]. IEEE Internet of Things Journal, 2018, 5(2): 884-894.

[5] LIAO G Q, ZHOU J, HUI N, et al. Approximately Filtering Redundant Data for Uncertain RFID Data Streams[C]// IEEE International Conference on Mobile Data Management. IEEE, 2017.

[6] LIU L L, YUAN Z L, LIU X W, et al. RFID unreliable data filtering by integrating adaptive sliding window and Euclidean distance[J]. Advances in Manufacturing, 2014, 2(2): 121-129.

[7] YOUSIF A, KAFAFY A, ABDLKADER H M. Reducing RFID Data Uncertainty using Mean Field Variational Inference[C]// 2018 14th International Computer Engineering Conference (ICENCO). 2018: 131-136.

[8] LIN Q M, XIAO Y, YE N, et al. A method of cleaning RFID data streams based on Naive Bayes classifier[J]. International Journal of Ad Hoc & Ubiquitous Computing, 2016, 21(4): 237.

[9] MAHDIN H. A Review on Bloom Filter Based Approaches for RFID Data Cleaning[C]// First International Conference on Advanced Data and Information Engineering. 2014: 79-86.

[10] BLOOM B H. Space/time trade-offs in hash coding with allowable errors[J]. IPSJ Magazine, 1970, 12(7): 422-426.

[11] MOUSAVI N, TRIPUNITARA M. Constructing Cascade Bloom Filters for Efficient Access Enforcement[J]. Computers & Security, 2019, 81: 1-14.

[12] REVIRIEGO P, MARTINEZ J, PONTARELLI S. CFBF: Reducing the Insertion Time of Cuckoo Filters with an Integrated Bloom Filter[J]. IEEE Communications Letters, 2019, 23(10): 1857-1861.

[13] LU J, YANG T, WANG Y, et al. Low Computational Cost Bloom Filters[J]. IEEE/ACM Transactions on Networking, 2018, 26(5): 2254-2267.

[14] KLEYKO D, RAHIMI A, GAYLER R W, et al. Autoscaling Bloom Filter: Controlling Trade-off Between True and False Positives[J]. Neural Computing & Applications, 2020, 32: 3675-3684.

[15] LEE C H, CHUNG C W. An approximate duplicate elimination in RFID data streams[J]. Data & Knowledge Engineering, 2011, 70(12): 1070-1087.

[16] WANG Y L, WANG C, JIANG X H, et al. RFID duplicate removing algorithm based on temporal-spatial Bloom filter[J]. Journal of Nanjing University of Science and Technology, 2015 (3): 253-259.

[17] ZHU W L. Warehouse Package Longitudinal Positioning Based on RFID Data Redundancy Cleaning and Clustering[D]. Chongqing: Chongqing University, 2017.

[18] HUANG W Q, ZHANG Y F, CAO Z W, et al. Redundant RFID Data Filtering Algorithm Research Based on Bloom Filter[J]. Journal of Cyber Security, 2019, 4(3): 93-105.

[19] GUO D, WU J, CHEN H, et al. Theory and Network Applications of Dynamic Bloom Filters[C]// Proceedings IEEE INFOCOM 2006, 25th IEEE International Conference on Computer Communications. IEEE, 2006: 2849-2860.

[20] XIE K, MIN Y H, ZHANG D F, et al. A Scalable Bloom Filter for Membership Queries[C]// IEEE Global Telecommunications Conference. IEEE, 2007: 543-547.

DUAN Wen, born in 1996, postgraduate, is a student member of China Computer Federation. Her main research interests include information system integration and so on.

ZHOU Liang, born in 1966, Ph.D, associate professor, master supervisor. His main research interests include information system integration and knowledge engineering.