

一种 WSN 中的三层多维事件协作检测算法

王浩云 刘俊俊 方贺贺 任守纲 徐焕良

(南京农业大学信息科技学院 南京 210095)

摘 要 提出了一种适用于无线传感器网络的三层多维事件协作检测算法。传感器节点通过计算均值向量序列的相似度发现异常,并通过投票机制确认事件发生。簇头节点根据边界向量序列的相似度,利用改进的 K 均值聚类算法对多维事件数据进行分类和合并。汇聚节点利用事件属性数据的概率分布,匹配检测出事件的类型。理论分析和仿真试验的结果表明:与传统集中式的事件检测算法相比,该算法能在噪声干扰下提高对多维事件的检测精度,降低算法的通信量和计算复杂度,延长网络的生存时间。

关键词 无线传感器网络,多维时间序列,事件检测,协作算法,类型匹配

中图法分类号 TP393 **文献标识码** A

3-layer Cooperative Detection Algorithm for Multi-dimensional Events in WSN

WANG Hao-yun LIU Jiao-jiao FANG He-he REN Shou-gang XU Huan-liang

(School of Information Science & Technology, Nanjing Agricultural University, Nanjing 210095, China)

Abstract A 3-layer cooperative detection algorithm for multi-dimensional events was proposed in wireless sensor networks. Sensors detect abnormality through the similarity of mean vector sequences and confirm the occurrence of events with the voting mechanism. On the basis of similarity of boundary vector sequences, cluster-headers analyze multi-dimensional events data using a modified K-means algorithm. Sinks match the types of known events using the distribution probability of event data. The results of theory analysis and simulation experiments indicate that compared with the traditional centralized algorithm, this 3-layer cooperative detection algorithm for multi-dimensional events can improve the detection precision under the interference of noise, reduce the data traffic and the computation complexity, and prolong the network lifetime in wireless sensor networks.

Keywords Wireless sensor networks (WSN), Multi-dimensional time series, Event detection, Cooperative algorithm, Type matching

1 引言

无线传感器网络(wireless sensor network, WSN)^[1]由随机部署在监测区域内的大量传感器节点(sensor)组成,通过无线通信方式形成多跳自组织网络。其目的是协作地感知、采集和处理网络覆盖区域中感知对象的信息,并发送给汇聚节点(sink)以便用户进行存储和查询。WSN 是以数据为中心的网络^[1],具有精度高、容错性好、可扩展性强以及动态自组织等特性,因此具有广阔的应用前景^[2]。

WSN 需要 sensor 节点能及时发现周围环境中的事件,并判断所发生事件的类型, sink 节点才能采取合适的策略,对周围的环境进行控制。传统 WSN 中事件检测^[3,4]是以集中式的方式在 sink 节点上进行的,但节点读数的误差以及通信过程中噪声的干扰会影响事件检测的精度,并且大多数算法

都是针对 sensor 节点所感知的一维数据进行检测的。同时,现有检测算法由于采取集中式的方式,能耗较高。本文将提出一种新的适用于 WSN 的三层多维事件协作检测方法。首先, sensor 节点相互协作检测多维事件的发生。 sensor 节点通过计算均值向量序列的相似度发现异常后,向周边的 sensor 节点进行问询,根据周边 sensor 节点投票的结果得到事件发生的可信程度。当可信度超过阈值后, sensor 节点将和簇头节点协作对多维事件进行聚类分析。簇头节点在接受到 sensor 节点上报的多维事件数据后,计算边界向量序列的相似度,并利用改进后的 K-means 算法对多维事件数据进行分类和合并。最后,簇头节点将和 sink 节点协作检测多维事件的类型。 sink 节点利用事件属性数据的概率分布,匹配检测出事件的类型。

到稿日期:2013-07-14 返修日期:2013-10-12 本文受中央高校基本科研业务费专项资金资助项目(KYZ201421),国家科技部重大科技支撑计划资助项目(2011BAK21B05),国家农业科技成果转化资金项目(2011GB2C100001),江苏省工业和信息产业转型升级专项引导资金项目(2011C1),中央高校基本科研业务费资助项目(Y0201100080),江苏省科技支撑计划资助项目(BE2011398, BE2011339),南京农业大学 2013 年国家大学生创新创业训练计划资助项目(201310307049)资助。

王浩云(1981—),男,博士,讲师,CCF 会员,主要研究方向为 P2P 网络、物联网、CPS 相关协议和机制优化设计等;刘俊俊(1990—),女,硕士生,主要研究方向为农业物联网关键技术;方贺贺(1994—),女,主要研究方向为 CPS 建模方法;任守纲(1977—),男,博士,副教授,主要研究方向为农业物联网应用技术;徐焕良(1963—),男,博士,教授,主要研究方向为物联网关键技术, E-mail: huanliangxu@njau.edu.cn(通信作者)。

2 相关研究

无线传感器网络已经在许多应用领域进行了部署。通常,WSN是由sink节点和一组sensor节点构成。sensor节点从物理环境中收集信息,而sink节点执行数据融合和分析功能。这种集中式架构存在数据冗余、功能复杂、计算量较大、鲁棒性较差等缺点,且靠近sink的sensor节点存在能量早衰的问题^[5]。文献[6]提出了WSN协作架构,使得sensor与sensor之间、sensor与簇头(cluster-header, CLH)节点之间以及CLH节点与sink节点之间能够协同工作,以实现整体应用目标。本文所提事件检测算法将利用WSN中节点间的协作关系,使sensor节点、CLH节点和sink节点协作完成事件检测任务。检测任务的能耗被分摊到WSN中的各类节点上,缓解了节点能量早衰的问题,延长了WSN的工作寿命。同时,原先在sink节点上执行的事件检测的计算量可由sensor节点和CLH节点进行分担,检测算法的设计难度也被降低。

大多WSN的事件检测算法中,sensor节点的数据在sink节点进行数据融合^[1,3,4]。如果个别sensor节点出现故障或被人为破坏,其错误读数将会影响sink节点的数据融合,从而得出错误的判断。文献[7]提出了一种通过对中心节点得到的数据集进行主成分分析的方法来得到事件的特征值,但计算复杂度较高。文献[8]提出了一种通过主动注入检测矢量,然后与原始数据交叉验证的方法来进行事件检测,但这种方法可能会改变事件区域的环境。文献[9]提出了一种基于局部投票的分布式的检测方法,其利用“少数服从多数”的规则过滤掉sensor节点错误的读数,从而保证数据的准确性。文献[10]同时分析了事件区域内的sensor节点、事件边界上的节点以及错误节点的读数在时空相关性上的区别。本文采用局部投票机制在避免错误节点读数干扰的同时,尽可能将事件区域中心的读数上传给sink节点,从而使得sink节点能准确地检测出事件的类型。

目前,WSN中的事件检测方法大致可以归纳为3类^[11]:基于阈值的检测方法、基于时空模型的检测方法以及基于模式的检测方法。如果传感器的读数超过预设的某个阈值,则基于阈值的检测方法将判断某个事件发生;基于时空模型的检测方法是利用传感器读数的时空相关性建模,对事件发生的可能性做出预测;基于模式的检测方法是利用模式识别的方法判断事件是否发生。前两类事件检测方法存在一定的局限性。基于阈值的方法需要利用长时间序列样本对机器学习算法进行训练^[11],从而得到优化的阈值,但这无法满足WSN实时部署的应用需求;基于时空模型的方法需要建立正确的预测模型,但这和WSN具体的应用领域相关联,需要人为调整预测模型的参数。在实际的WSN应用中,需要从感知数据中发现事件类型,从而制定相应的控制策略。因此,本文将采用基于模式的检测方法作为WSN中的事件检测方法。

WSN中的sensor节点的读数将形成时间序列。时间序列表示方法是时间序列数据挖掘的前提,直接关系到挖掘算法的运行效率和处理结果的准确度。目前,时间序列主要表示方法有:DWT(discrete wavelet transform)^[12]、PAA(piecewise aggregate approximate)^[13]、PLA(piecewise linear approximation)^[14]和SAX(symbolic aggregate approximation)^[15]等。其中,PAA方法使用一个固定宽度的滑动窗口

来界定序列的每个分段,并计算窗口内数据的平均值。PAA方法计算量较小,适于在能量受限的sensor节点上执行。并且,它能对时间序列进行有效降维,数据中变化比较大的部分特征能被有效保留,便于事件检测算法对其的后期处理和数据挖掘。文献[16]提出了基于边界的时间序列聚类算法。而文献[17]在其基础上进一步考虑了时间序列的上界和下界。通过计算序列的边界,在保留子序列均值信息的基础上,进一步补充了子序列的特征信息,记录了子序列数据波动的最大范围。本文将分别在sensor节点和CLH节点采用PAA和基于上下界的方法表示事件数据,并在此基础上进行事件检测。

3 概念和定义

设WSN监测区域是一个二维空间,记为 R 。在区域 R 内,存在 n_s 个sensor节点,且每个传感器节点 $s_i, i \in \{1, \dots, n_s\}$ 携带有 d_s 种传感器。当sensor节点分簇稳定后,区域内共有 n_c 个簇,不同的普通sensor节点隶属于不同的簇。sensor节点能够将其感知的数据通过相应的CLH节点 $c_k, k \in \{1, \dots, n_c\}$ 传递给sink节点。假设监测区域内潜在的事件类型有 D 种。

3.1 多维向量时间序列

区域 R 的属性值随时间动态变化。sensor节点上的多个传感器将周期性对区域 R 的属性值进行采样。时刻 t ,sensor节点 s_i 记录的属性向量记为 $\bar{m}_{s_i}(t) = (x_{s_i}^1(t), x_{s_i}^2(t), \dots, x_{s_i}^{d_s}(t))$ 。sensor节点在时刻 t_1 到时刻 t_j 间的 j 个连续采样时间点记录的数据将构成多维向量时间序列,记为 $\bar{m}_{s_i}(t_1, t_j) = \{\bar{m}_{s_i}(t_1), \bar{m}_{s_i}(t_2), \dots, \bar{m}_{s_i}(t_j)\}$ 。

为了从复杂的多维向量时间序列中发现事件及其类型,就需要对时间序列进行分割,然后对分割后的子序列进行聚类、分类、异常监测等数据挖掘^[18]工作。本文采用类似PAA的方法在sensor节点对所接收到的多维向量时间序列进行等长分割,划分的粒度为 g_s 。 g_s 值越大,则分割后的子序列对变化的敏感度越低;但其如果设置得过小,数据模式容易被破坏,且算法的能耗会增加。

考虑到WSN应用对实时性的要求,本文在sensor节点和CLH节点上使用滑动窗口以支持在线检测。滑动窗口的长度分别设为 W_s 和 W_c 。滑动窗口长度越大,历史数据在滑动窗口内的比重越大,最新数据的变化也越不敏感;但如果长度过小,则滑动窗口内数据的模式容易被忽略,且节点维护滑动窗口的能耗会增加。因此,正确地配置 g_s 和 W_s, W_c 的大小对WSN事件监测算法的准确性和能耗都会有一定的影响。在实际应用时,需要根据特定应用的需求调整 g_s 和 W_s, W_c 的设置。

3.2 均值向量序列

sensor节点 s_i 利用长度为 W_s 、步长为1的滑动窗口 $[t_1, t_{W_s}]$ 在时刻 t_{W_s} 得到一个多维向量时间序列 $\bar{m}_{s_i}(t_1, t_{W_s}) = \{\bar{m}_{s_i}(t_1), \bar{m}_{s_i}(t_2), \dots, \bar{m}_{s_i}(t_{W_s})\}$,其中, $\bar{m}_{s_i}(t_j, j \in \{1, 2, \dots, W_s\}) = (x_{s_i}^1(t_j), x_{s_i}^2(t_j), \dots, x_{s_i}^{d_s}(t_j))$ 。根据分割粒度 g_s ,序列 \bar{m}_{s_i} 将被等长分割为 $\theta_s = \frac{W_s}{g_s}$ 个子序列。根据PAA方法,各个子序列的均值将被计算出来,从而得到序列 $\bar{m}_{s_i}(t_1, t_{W_s})$ 的均值向量序列 $\bar{p}_{s_i}(t_{W_s}) = \{\bar{p}_{s_i,1}(t_{W_s}), \bar{p}_{s_i,2}(t_{W_s}), \dots, \bar{p}_{s_i,\theta_s}(t_{W_s})\}$ 。 $\bar{p}_{s_i}(t_{W_s})$ 中第 j 个向量 $\bar{p}_{s_i,j}(t_{W_s})$ 的计算方法如式(1)所示。

$$\bar{p}_{s_i,j}^d(t_{W_s}) = \frac{1}{g_s} \sum_{k=g_s(j-1)+1}^{g_s j} \bar{m}_{s_i}^d(t_k) \quad (1)$$

式中, $\bar{m}_{s_i}^d(t_k)$ 表示 $\bar{m}_{s_i}(t_k)$ 的第 d 维元素, $\bar{p}_{s_i}^d(t_{W_s})$ 表示 $\bar{p}_{s_i}(t_{W_s})$ 的第 d 维元素, 且 $\bar{p}_{s_i,j}(t_{W_s}) = (\bar{p}_{s_i,j}^1(t_{W_s}), \dots, \bar{p}_{s_i,j}^d(t_{W_s}), \dots, \bar{p}_{s_i,j}^{\theta_s}(t_{W_s}))$ 。

3.3 边界向量序列

当区域 R 有事件发生时, CLH 节点 c_i 将收到 sensor 节点 s_i 上传的实时边界向量序列 \bar{q}_{s_i} 。sensor 节点 s_i 首先利用滑动窗口 $[t_1, t_{W_s}]$ 在时刻 t_{W_s} 得到多维向量时间序列 $\bar{m}_{s_i}(t_1, t_{W_s})$, 然后将序列 \bar{m}_{s_i} 等长分割为 θ_s 个子序列, 并且各个子序列的上界和下界将根据式(2)被计算出来, 从而得到序列 $\bar{m}_{s_i}(t_1, t_{W_s})$ 的边界向量序列 $\bar{q}_{s_i}(t_{W_s}) = \{\bar{q}_{s_i,1}(t_{W_s}), \bar{q}_{s_i,2}(t_{W_s}), \dots, \bar{q}_{s_i,\theta_s}(t_{W_s})\}$ 。 $\bar{q}_{s_i}(t_{W_s})$ 中第 j 个向量 $\bar{q}_{s_i,j}(t_{W_s})$ 的计算公式为:

$$\bar{q}_{s_i,j}^d(t_{W_s}) = \bar{u}_{s_i,j}^d(t_{W_s}) + \bar{l}_{s_i,j}^d(t_{W_s}) \cdot i \quad (2)$$

$$\bar{u}_{s_i,j}^d(t_{W_s}) = \max\{\bar{m}_{s_i}^d(t_{g_s(j-1)+1}), \dots, \bar{m}_{s_i}^d(t_{g_s j})\} \quad (3)$$

$$\bar{l}_{s_i,j}^d(t_{W_s}) = \min\{\bar{m}_{s_i}^d(t_{g_s(j-1)+1}), \dots, \bar{m}_{s_i}^d(t_{g_s j})\} \quad (4)$$

式中, $\bar{q}_{s_i,j}^d(t_{W_s})$ 表示 $\bar{q}_{s_i,j}(t_{W_s})$ 的第 d 维元素, 且

$$\bar{q}_{s_i,j}(t_{W_s}) = (\bar{q}_{s_i,j}^1(t_{W_s}), \dots, \bar{q}_{s_i,j}^d(t_{W_s}), \dots, \bar{q}_{s_i,j}^{\theta_s}(t_{W_s}))$$

式中, $\bar{u}_{s_i,j}^d(t_{W_s})$ 表示子序列的上界, $\bar{l}_{s_i,j}^d(t_{W_s})$ 表示子序列的下界。

4 协作事件检测算法

4.1 sensor 协作检测算法

本文将提出一种新的适用于 WSN 的三层多维事件协作检测方法。首先, sensor 节点协作检测事件的发生。sensor 节点发现检测数据异常后, 向周边的 sensor 节点进行询问, 根据周边 sensor 节点投票的结果得出事件发生的可信程度。当确认事件发生后, sensor 节点将感知到的数据发送给簇头节点 CLH, 进一步检测事件类型。

本文所提的事件检测方法, 首先需要 sensor 节点进行协作, 以检测区域 R 是否有事件发生。在时刻 t_{W_s} , sensor 节点 s_i 根据滑动窗口得到多维向量时间序列 $\bar{m}_{s_i}(t_1, t_{W_s})$ 。节点 s_i 根据式(1)所示方法将 $\bar{m}_{s_i}(t_1, t_{W_s})$ 转换为对应的均值向量序列 $\bar{p}_{s_i}(t_{W_s})$ 。随着滑动窗口的移动, 节点 s_i 在相邻时刻得到

$$D_s(\bar{q}_{s_i}(t_m), \bar{q}_{s_j}(t_n)) = \|\bar{q}_{s_i}(t_m) - \bar{q}_{s_j}(t_n)\| =$$

$$\begin{cases} \frac{1}{\theta_s \cdot d_s} \sum_{l=1}^{\theta_s} \sum_{k=1}^{d_s} \frac{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}, \text{Re}(\bar{q}_{s_i,l}^k(t_m)) \geq \text{Re}(\bar{q}_{s_j,l}^k(t_n)) \cap \text{Im}(\bar{q}_{s_i,l}^k(t_m)) \leq \text{Im}(\bar{q}_{s_j,l}^k(t_n)) \\ \frac{1}{\theta_s \cdot d_s} \sum_{l=1}^{\frac{W_s}{g_s}} \sum_{k=1}^{d_s} \frac{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}, \text{Re}(\bar{q}_{s_i,l}^k(t_m)) \geq \text{Re}(\bar{q}_{s_j,l}^k(t_n)) \cap \text{Im}(\bar{q}_{s_i,l}^k(t_m)) \leq \text{Im}(\bar{q}_{s_j,l}^k(t_n)) \\ \frac{1}{\theta_s \cdot d_s} \sum_{l=1}^{\frac{W_s}{g_s}} \sum_{k=1}^{d_s} \frac{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}, \text{Re}(\bar{q}_{s_i,l}^k(t_m)) \geq \text{Re}(\bar{q}_{s_j,l}^k(t_n)) \cap \text{Im}(\bar{q}_{s_i,l}^k(t_m)) \leq \text{Im}(\bar{q}_{s_j,l}^k(t_n)) \leq \text{Re}(\bar{q}_{s_j,l}^k(t_n)) \\ \frac{1}{\theta_s \cdot d_s} \sum_{l=1}^{\frac{W_s}{g_s}} \sum_{k=1}^{d_s} \frac{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}{\text{Re}(\bar{q}_{s_i,l}^k(t_m)) - \text{Im}(\bar{q}_{s_i,l}^k(t_m))}, \text{Re}(\bar{q}_{s_i,l}^k(t_m)) \geq \text{Re}(\bar{q}_{s_j,l}^k(t_n)) \cap \text{Im}(\bar{q}_{s_i,l}^k(t_m)) \leq \text{Im}(\bar{q}_{s_j,l}^k(t_n)) \leq \text{Re}(\bar{q}_{s_j,l}^k(t_n)) \\ 0, \text{else} \end{cases} \quad (6)$$

如果两个边界向量序列相似度较高, 则可以合并为独立的边界向量序列。边界向量序列合并的计算方法如式(7)所示

$$U_c(\bar{q}_{s_i,k}^d(t_m), \bar{q}_{s_j,k}^d(t_n)) = \begin{cases} \text{Re}(\bar{q}_{s_i,k}^d(t_m)) + \text{Im}(\bar{q}_{s_i,k}^d(t_m)) \cdot i, \text{Re}(\bar{q}_{s_i,k}^d(t_m)) \geq \text{Re}(\bar{q}_{s_j,k}^d(t_n)) \cap \text{Im}(\bar{q}_{s_i,k}^d(t_m)) \leq \text{Im}(\bar{q}_{s_j,k}^d(t_n)) \\ \text{Re}(\bar{q}_{s_j,k}^d(t_n)) + \text{Im}(\bar{q}_{s_j,k}^d(t_n)) \cdot i, \text{Re}(\bar{q}_{s_j,k}^d(t_n)) \geq \text{Re}(\bar{q}_{s_i,k}^d(t_m)) \cap \text{Im}(\bar{q}_{s_j,k}^d(t_n)) \leq \text{Im}(\bar{q}_{s_i,k}^d(t_m)) \\ \text{Re}(\bar{q}_{s_i,k}^d(t_m)) + \text{Im}(\bar{q}_{s_i,k}^d(t_m)) \cdot i, \text{Re}(\bar{q}_{s_i,k}^d(t_m)) \geq \text{Re}(\bar{q}_{s_j,k}^d(t_n)) \cap \text{Im}(\bar{q}_{s_i,k}^d(t_m)) \leq \text{Im}(\bar{q}_{s_j,k}^d(t_n)) \leq \text{Re}(\bar{q}_{s_j,k}^d(t_n)) \\ \text{Re}(\bar{q}_{s_j,k}^d(t_n)) + \text{Im}(\bar{q}_{s_j,k}^d(t_n)) \cdot i, \text{Re}(\bar{q}_{s_j,k}^d(t_n)) \geq \text{Re}(\bar{q}_{s_i,k}^d(t_m)) \cap \text{Im}(\bar{q}_{s_j,k}^d(t_n)) \leq \text{Im}(\bar{q}_{s_i,k}^d(t_m)) \leq \text{Re}(\bar{q}_{s_i,k}^d(t_m)) \\ 0, \text{else} \end{cases} \quad (7)$$

$\bar{p}_{s_i}(t_{W_s})$ 和 $\bar{p}_{s_i}(t_{W_s+1})$ 。当这两个序列的相似度距离超过阈值 δ_s 时, 节点 s_i 认为可能有事件发生。两均值向量序列相似度的计算方法如式(5)所示。

$$\begin{aligned} D_s(\bar{p}_{s_i}(t_{W_s+1}), \bar{p}_{s_i}(t_{W_s})) \\ = \|\bar{p}_{s_i}(t_{W_s+1}) - \bar{p}_{s_i}(t_{W_s})\| \\ = \frac{1}{\theta_s \cdot d_s} \sum_{j=1}^{\theta_s} \sum_{k=1}^{d_s} |\bar{p}_{s_i,j}^k(t_{W_s+1}) - \bar{p}_{s_i,j}^k(t_{W_s})| \end{aligned} \quad (5)$$

节点 s_i 认为当前时刻可能有事件发生时, 则将请求所有邻居节点进行投票。邻居节点 s_j 如果同样认为当前时刻可能有事件发生, 则投票 1 表示支持, 否则投票 0 表示否定。当节点 s_i 收到所有邻居节点投票后, 如支持率 v_{s_i} 超过 δ_v , 则确认事件发生。 δ_v 通常设置要大于 0.5, 这是因为支持率如果仅大于 0, 说明节点 s_i 感知到的是孤立的事件, 很可能是由于误差或干扰所导致的; 而支持率如果等于 0.5, 则说明节点 s_i 可能处在事件区域的边缘, 所得数据并不能准确反映事件的状态。节点 s_i 在时刻 t_j 检测到有事件可能发生, 而在时刻 t_k 收到邻居节点的投票结果后则确认事件发生, 之后节点 s_i 还需要继续向所属的 CLH 节点 c_i 上报事件相关的边界向量序列 $\bar{q}_{s_i}(t_k)$ 。

4.2 CLH 协作检测算法

sensor 节点 s_i 确认有事件发生后, 将和所在簇的 CLH 节点 c_i 协作对事件数据进行分类。节点 c_i 将利用一个长度为 W_c 、步长为 W_s 的滑动窗口接收所在簇内的多个普通 sensor 节点实时上传的事件相关的边界向量序列, 并对滑动窗口内所得多维数据进行 K-means 聚类分析。

为了进行聚类分析, 簇头节点需要计算所收到的两个边界向量序列的相似度。CLH 节点 c_i 在时间区间 $[t_1, t_{W_c}]$ 内接收到 sensor 节点 s_i 和节点 s_j 上传的边界向量序列 $\bar{q}_{s_i}(t_{m,m \in \{1,2,\dots,W_c\}})$ 和 $\bar{q}_{s_j}(t_{n,n \in \{1,2,\dots,W_c\}})$ 。边界向量序列相似度的计算方法如式(6)所示。其中, $\text{Re}(\cdot)$ 和 $\text{Im}(\cdot)$ 分别表示复数的实部和虚部。

CLH 节点 c_i 将采用 K-means 算法,对利用滑动窗口在时刻 t 所收到的边界向量序列按照事件接收的先后顺序进行聚类。聚类算法中的 K 值为潜在的事件类型种类 D 。算法利用式(6)对新到的序列和已有分类的中心点 $\overline{M}_{c_i,j,j \in \{1,2,\dots,D\}}$ 进行相似度计算,而对各分类的中心点的更新则可利用式(7)来完成。当聚类算法完成后,簇头节点将得到 D 个分类 $C_{j,j \in \{1,2,\dots,D\}}$,每个分类对应着不同传感器所上传的边界向量序列集合,各个分类都有各自的中心点 $\overline{M}_{c_i,j,j \in \{1,2,\dots,D\}}(t) = \{\overline{M}_{c_i,j,1}(t), \overline{M}_{c_i,j,2}(t), \dots, \overline{M}_{c_i,j,\theta_j}(t)\}$,且 $\overline{M}_{c_i,j,k}(t)$ 是 $\overline{M}_{c_i,j}(t)$ 的第 k 个向量。

4.3 sink 协作检测算法

CLH 节点 c_i 完成对多维边界向量序列的聚类分析后,将分类的中心点上传给 sink 节点,进行事件类型匹配。由于事件区域 R 存在 D 种类型的事件, sink 节点将聚类后的各中心点 $\overline{M}_{c_i,j,j \in \{1,2,\dots,D\}}(t)$ 与 D 种事件属性的概率分布函数 $F_{k,k \in \{1,2,\dots,D\}}(x)$ 进行匹配,以确定中心点所在分类对应的事件类型。匹配方法如式(8)所示。

$$D_{\text{sink}}(\overline{M}_{c_i,j}(t), F_k) = \frac{1}{\theta_s \cdot d_s} \cdot \sum_{l=1}^{\theta_s} \sum_{d=1}^{d_s} F_k^d(\text{Re}(\overline{M}_{c_i,j,l}(t))) - F_k^d(\text{Im}(\overline{M}_{c_i,j,l}(t))) \quad (8)$$

式中, $F_k^d(x)$ 表示事件第 d 维属性的概率分布函数。

如果 $D_{\text{sink}}(\overline{M}_{c_i,j}(t), F_n) = \max_{k \in \{1,2,\dots,D\}} \{D_{\text{sink}}(\overline{M}_{c_i,j}(t), F_k)\}$,则可确认该中心点 $\overline{M}_{c_i,j}(t)$ 所在分类的事件类型为 n ,同时也能确定上报该类型事件的传感器节点集合。可能存在的一种情况,就是同一类型事件的概率分布函数与多个分类的中心点 $\overline{M}_{c_i,l,l \neq j}(t)$ 都匹配成功,则将利用式(7)将多个中心点进行合并。合并完成后, sink 节点将得到与事件类型 n 匹配的分类中心点 $\overline{M}_{c_i,n}(t)$ 。

5 性能分析

本文所提的多维事件协作检测方法分为 3 层: sensor 节点协作检测算法、CLH 节点协作检测算法及 sink 节点协作检测算法。通过 sensor 节点、CLH 节点和 sink 节点的协作, WSN 应用中的节点能够检测周边环境中所发生事件的类型。本文所提算法在实际部署时需要考虑 WSN 应用对事件检测算法的低复杂度的要求。

sensor 协作检测算法中, sensor 节点 s_i 在时刻 t_j 将利用式(1)计算得到 $\overline{p}_{s_i}(t_j)$,其计算复杂度为 $O(d_s \times W_s)$ 。在利用式(5)计算两个序列的相似度时,其计算复杂度为 $O(2 \times d_s \times \theta_s)$ 。如果节点 s_i 在时刻 t_k 需要向簇头节点 c_i 上报数据,还需要利用式(2)计算得到 $\overline{q}_{s_i}(t_k)$,其计算复杂度为 $O(d_s \times W_s)$ 。因此,算法中 sensor 节点 s_i 的最大计算复杂度为 $O(2 \times d_s \times W_s)$ 。可以看出,如果 W_s 设置过大,则 sensor 节点有较大的计算复杂度;而如果 W_s 设置过小,则 $\overline{p}_{s_i}(t_j)$ 包含的数据信息较少,难以发现有用的数据模式。

CLH 协作检测算法中,簇头节点 c_i 在时刻 t 将利用式(6)进行聚类,得到 D 个分类,其计算复杂度为 $O(3 \times \theta_s \times d_s \times D \times (n_s - n_c - D))$ 。在 K-means 聚类的同时,需要利用式(7)计算各个分类的中心点,其计算复杂度为 $O(3 \times \theta_s \times d_s \times (n_s - n_c - D))$ 。因此,算法中节点 c_i 在时刻 t 的最大计算复杂度为 $O(3 \times \theta_s \times d_s \times (D+1) \times (n_s - n_c - D))$ 。考虑到簇头节点上滑动窗口的步长为 W_s ,簇头节点的平均最大计算复杂

度为 $O(3 \times \theta_s \times d_s \times (D+1) \times (n_s - n_c - D) / g_s)$ 。可以看出,如果 g_s 设置较大,则簇头节点的平均最大计算复杂度将会降低,但分割后的子序列对变化的敏感度也会降低;而如果 g_s 设置过小,则子序列的数据模式容易被破坏,且算法的计算复杂度会增加。此外,为了使簇头节点能够工作更长时间,普通传感器节点将利用簇头选举算法轮流担当簇头节点。

sink 协作检测算法中, sink 节点将利用式(8)进行事件类型匹配,其计算复杂度为 $O((2m+1) \times \theta_s \times d_s \times D^2)$ 。其中, m 为事件属性的概率分布函数 $F_{k,k \in \{1,2,\dots,D\}}(x)$ 的计算复杂度。考虑到匹配成功后, sink 节点需要利用式(7)将相同事件类型的分类中心点进行合并,其计算复杂度为 $O(3 \times \theta_s \times d_s \times (D-1))$ 。因此,算法中 sink 节点在时刻 t 的最大计算复杂度为 $O(\theta_s \times d_s \times D \times (3 \times (n_s - n_c) + 2 \times m \times D))$ 。考虑到 sink 节点可利用滑动窗口同步接收簇头节点上传的数据,因此 sink 节点上滑动窗口的步长为 W_s ,则 sink 节点的平均最大计算复杂度为 $O((d_s \times D \times (3 \times (n_s - n_c) + 2 \times m \times D)) / g_s)$ 。

6 仿真分析

6.1 仿真环境

本文将利用 Matlab 搭建仿真环境来测试所设计的三层事件协作检测算法。仿真中,事件区域 R 为 $400\text{m} \times 400\text{m}$,其中有 441 个普通传感器、16 个簇头节点和 1 个 sink 节点。假设传感器节点的分簇阶段已完成,所有普通传感器和簇头节点在事件区域内均匀分布,且传感器节点根据地理位置隶属于相应的簇头节点。普通传感器节点上有 3 种传感器,其采样频率均为 1Hz。普通传感器上的滑动窗口长度 W_s 为 16,分割粒度 g_s 为 4。而簇头节点的滑动窗口长度 W_c 为 24。当普通传感器节点感知到事件发生后,簇头节点将所感知的事件数据进行分类,而 sink 节点将进行类型匹配以确定事件的类型。

事件区域 R 内存在 4 类事件,每类事件的 3 个维度属性的概率分布函数独立服从正态分布 $N(\mu, \sigma^2)$, μ 为期望值, σ^2 为标准方差。因此,存在 12 种独立的分布函数 $N(\mu_i^d, (\sigma_i^d)^2)$,事件类型 $i=1,2,3,4$,事件属性的维度 $d=1,2,3$,且 μ_i^d 和 σ_i^d 将分别在区间 $[1,10]$ 内随机分配。

仿真中,不同时刻 t 在事件区域 R 的不同位置将随机产生不同类型的事件,且传感器的感知半径 R 为 10m。传感器的读数 $\overline{m}_{s_i}(t)$ 随着传感器节点 s_i 和事件位置的距离 λ 而变化,且 $\overline{m}_{s_i}(t) = e^{-\lambda/2R} \cdot \overline{m}(t)$ 。其中, $\overline{m}(t)$ 为事件的实际属性值。如果 sink 节点正确检测出事件类型,则所发生的事件将消亡;否则所发生的事件将持续到仿真结束。考虑到传感器节点工作时存在误差,各节点的读数的误差将独立服从正态分布 $N(0,1)$ 。本文的仿真试验主要针对所提三层事件协作检测算法的有效性和通信量进行分析,同时分析算法中各参数对算法性能的影响。

考虑到目前 WSN 中的事件检测方法大多是集中式的,仿真试验将构造集中式的事件检测算法,将所有的感知多维事件数据通过簇头节点上传给 sink 节点进行事件检测,检测方法为常用的 K 均值聚类算法。通过两种事件检测方式的对比,进一步分析本文所提多维事件协作检测算法的性能。

6.2 仿真结果

6.2.1 有效性

仿真试验将考察本文所提事件协作检测算法的有效性,

包括算法的正确率和错误率。正确率为簇头节点检测事件类型正确的次数除以簇头节点上报事件的总数。错误率包括错误肯定率和错误否定率。错误肯定率为簇头检测出事件但类型检测错误的次数和实际没有事件发生但检测出事件类型的次数占簇头上报事件总数的比例；而错误否定率为实际有事件发生而没有上报的比例。仿真试验将评估不同参数 δ_s, δ_o 取值对算法有效性的影响。仿真结果如图 1 所示。

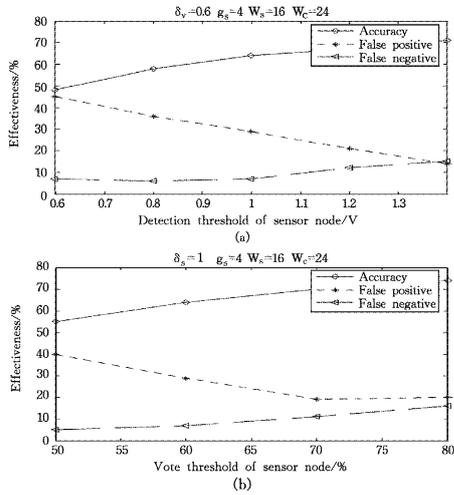


图 1 事件检测的有效性

由图 1(a)可以看出,当参数 δ_s 从 0.6 增至 1.4 时,算法的正确率从 50% 左右上升到 70% 左右。这是因为随着 δ_s 的增加,普通传感器节点能够感知到的事件在逐渐减少,sink 节点检测出的事件总数也会相应减少。同时,传感器的读数受噪声的影响逐渐减弱,sink 节点对事件类型正确检测的次数将会增加。这样,算法的正确性将会随着 δ_s 的增加而上升。图 1(a)中,算法的错误肯定率从 45% 左右下降至 15% 左右。这是因为当 δ_s 逐渐增加时,sink 节点检测出事件但类型错误的次数有所下降,且随着噪声对传感器读数影响的减弱,实际没有事件发生但检测出事件类型的次数也会有所下降。这样,算法的错误肯定率逐渐降低。仿真中,算法的错误否定率随着 δ_s 的增加,上升的幅度接近 10%。

由图 1(b)可以看出,当参数 δ_o 从 0.5 增至 0.8 时,算法的正确率从 55% 左右上升到 75% 左右。这是因为随着 δ_o 的增加,上报事件的普通传感器节点需要距离事件源更近,而距离对传感器读数衰减的影响也更小,sink 节点对事件类型正确检测的次数将会增加。同时,普通传感器上报事件的次数将会减小,sink 节点检测出事件的总数也会相应减少。这样,算法的正确性将会随着 δ_o 的增加而上升。图 1(b)中,算法的错误肯定率从 40% 左右下降至 20% 左右。这是因为当 δ_o 逐渐增加时,sink 检测出事件但类型错误的次数有所下降,且随着距离对传感器读数影响的减弱,实际没有事件发生但检测出事件类型的次数也会有所下降。这样,算法的错误肯定率逐渐降低。仿真中,算法的错误否定率随着 δ_o 的增加,上升幅度超过 10%。

6.2.2 通信量

WSN 中节点相互协作需要进行通信,而频繁通信会导致节点能量消耗较快。仿真试验将评估不同参数 g_s, W_s 取值对本文所提事件协作检测算法中节点间通信量的影响。仿真

利用 WSN 节点单位仿真时间内所发送信息的平均次数来衡量节点的通信量。同时,本仿真还将评估所构造的集中式的事件检测算法的通信量。仿真结果如图 2 所示。

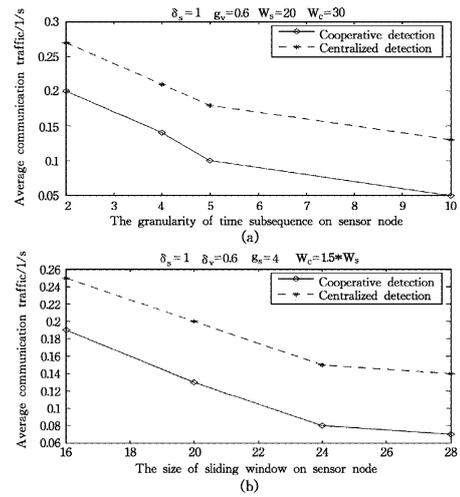


图 2 事件检测算法的通信量

由图 2(a)可以看出,事件协作检测算法中节点的通信量随 g_s 的增加而从 0.2 次/秒逐渐下降到 0.05 次/秒。这是因为当 g_s 逐渐增加时,序列所表达的关于事件的信息量也减少,节点成功发现事件并检测出事件类型的概率将会降低,因而上报事件数据通信次数也将减少。但算法的有效性将受到影响。图 2(a)中,集中式的事件检测算法的通信量要高于本文所提算法的通信量,这是因为 sensor 节点感知到的事件数据需要通过 CLH 节点最后都汇聚于 sink 节点。由图 2(b)可以看出,事件协作检测算法的通信量随 W_s 的上升而从 0.19 次/秒逐渐下降到 0.08 次/秒左右。这是因为当 W_s 逐渐增加时,传感器节点上报数据中关于事件的信息量将会相应增加,而上报的次数则会有所下降。图 2(b)中,集中式的事件检测算法的通信量同样要高于本文所提算法的通信量。

结束语 本文针对 WSN 应用环境中事件具有多维属性的特点,提出了一种新的三层多维事件协作检测方法,其不仅能够发现多维属性事件的发生,同时能够检测出多维事件的类型。针对传统事件检测精度易受通信噪声和读数误差影响的问题,算法在 PAA 方法的基础上,将均值序列转化为边界序列进行聚类分析和类型匹配,在降低干扰的同时保证检测的精度。由于采用了三层协作的架构,因此降低了算法的通信量和计算复杂度。理论分析和仿真试验对所提算法的性能进行了验证。

本文所提的事件检测算法主要是面对传感器节点所感知周围环境中的标量数据。今后的 WSN 应用中传感器节点所感知的数据将包括音频和视频等多媒体数据,sink 节点将根据对多媒体数据的深度挖掘进行事件分析,从而制定并执行复杂的控制策略。因此,针对多媒体数据集的挖掘和分析将是事件检测方法进一步研究的方向。此外,算法性能的验证下一步可以结合真实无线传感器网络,加载真实环境数据进行进一步验证和完善,从而提高检测算法的实用性。

参考文献

[1] Abdegawad A, Bayoumi M. Data Fusion in WSN[J]. Lecture Notes in Electrical Engineering, 2012, 118: 17-35

- [2] Viani F, Oliveri G, Donelli M, et al. WSN-based Solutions for Security and Surveillance[C]//Proceedings of the 3rd European Wireless Technology Conference, Piscataway, USA; IEEE, 2010; 285-288
- [3] Bhuvaneshwari P T V, Vaidehi V, Karthik M. Dasarathy Model based Fusion Framework for Fire Detection Application in WSN [C]//Proceedings of 2nd International Conference on Networks and Communications. Heidelberg, Germany; Springer Verlag, 2010; 472-480
- [4] Feng Y B, Zhang R B. Fault Detection of WSN Based on Spatial Correlation[C]//Proceedings of International Conference on Information Technology for Manufacturing Systems. Clausthal-Zellerfeld, Germany; Trans Tech Publications, 2011; 1504-1510
- [5] Shen D D. Energy-saving Routing Algorithms with Optimization for Wireless Sensor Networks[D]. Hanzhou: Zhejiang University of Technology, 2009
- [6] Huang X, Tan L, Hu K. A Cooperative Game-based Clustering Algorithm in WSN[J]. Advances in Information Sciences and Service Sciences, 2012, 4(8): 147-154
- [7] Chatzigiannakis V, Papavassilion S. Diagnosing Anomalies and Identifying Faulty Nodes in Sensor Networks[J]. Sensor Journal, 2007, 7(5): 637-645
- [8] Gao J L, Xu Y J, Li X W. Online Distributed Fault Detection of Sensor Measurements[J]. Tsinghua Science and Technology, 2007, 12(1): 192-196
- [9] Chen J R, Kher S, Somani A. Distributed Fault Detection of Wireless Sensor Networks[C]//Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks. New York, USA; ACM, 2006; 65-71
- [10] Ding M, Chen D C, Xing K, et al. Localized Fault-tolerant Event Boundary Detection in Sensor Networks[C]//Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. New York, USA; IEEE, 2005; 902-913
- [11] Zoumboulakis M, Roussos G. Complex Event Detection in Extremely Resource-constrained Wireless Sensor Networks[J]. Mobile Networks and Applications, 2011, 16(2): 194-213
- [12] Ngaopitakkul A, Pothisarn C. The Protection of Transmission Network Systems Using Discrete Wavelet Transforms[J]. International Journal of Innovative Computing Information and Control, 2012, 8(9): 6491-6502
- [13] Ohsaki M, Abe H, Yamaguchi T. Numerical Time-series Pattern Extraction based on Irregular Piecewise Aggregate Approximation and Gradient Specification[J]. New Generation Computing, 2007, 25(3): 213-222
- [14] Jiang Y L, Chen H B. Application of General Orthogonal Polynomials to Fast Simulation of Nonlinear Descriptor Systems Through Piecewise-linear Approximation [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(5): 804-808
- [15] Li H L, Guo C H. Symbolic Aggregate Approximation based on Shape Features[J]. Pattern Recognition and Artificial Intelligence, 2011, 24(5): 665-672
- [16] Zhang X P, Zhu X C, Ma C. Clustering Algorithm based on Boundary Identification[J]. Journal of Xi'an Jiaotong University, 2007, 41(12): 1387-1390
- [17] Dai Dong-Bo, Xiong Yun, Zhu Yang-Yong. Efficient Algorithm for Sequence Similarity Search based on Reference Indexing[J]. Journal of Software, 2010, 21(4): 718-731
- [18] Xiang H M, He W S. Mining Wireless Sensor Network Data based on Vector Space Model[C]//Proceedings of International Conference on Computing, Information and Control. Heidelberg, Germany; Springer, 2011; 100-104

(上接第 81 页)

- [2] Wang Xing-wei, Cheng Hui, Huang Min. Multi-robot Navigation Based QoS Routing in Self-organizing Networks [J]. Engineering Applications of Artificial Intelligence, 2013, 26(1): 262-272
- [3] Wang Xing-wei, Cheng Hui, Li Ke-qin, et al. A Cross-layer Optimization Based Integrated Routing and Grooming Algorithm for Green Multi-granularity Transport Networks [J]. Journal of Parallel and Distributed Computing, 2013, 7(6): 807-822
- [4] Lee S S W, Tseng P-k, Li Kuang-yi, et al. Interface Specific Fast Failure Rerouting for Load Balanced IP Networks [C]// 2011 IEEE Symposium on Computers and Communications (ISCC). Kerkyra, 2011; 316-319
- [5] Metnani A, Jaumard B. Connection Rerouting in GRWA Networks [C]//2011 18th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN). Chapel Hill, NC, 2011; 1-6
- [6] Xu Ming-wei, Yang Yuan, Li Qi. Selecting Shorter Alternate Paths for Tunnel-based IP Fast ReRoute in Linear Time [J]. Computer Networks, 2011, 56(2): 845-957
- [7] Amdouni N, Koubaa M, Aguilu T. Lightpath Rerouting Scheme for Dynamic Traffic in WDM All-Optical Networks [C]//2012 International Conference on Computer Systems and Industrial Informatics (ICCSII). Sharjah, 2012; 1-6
- [8] Ma Wen-chao, Fang Yu-guang. Dynamic hierarchical mobility management strategy for mobile IP networks [J]. IEEE Journal on Select Areas Commun, 2004, 22(4): 664-676
- [9] Li Xin, Qin Zhen, Yu Tao. Optimizing the QoS Performance of Fast Rerouting [C]// 2009 Ninth International Conference on Hybrid Intelligent Systems. Shenyang, 2009; 313-318
- [10] Tian Li-qin, Lin Chuang, Ni Yang. Behavior Value Analysis and Application in Evaluating Network Entity Behavior Trust [C]// 2010 2nd International Conference on Computer Engineering and Technology (ICCET). Chengdu, 2010; 313-318
- [11] 王学军, 郭亚军. 基于 G1 法的判断矩阵的一致性分析[J]. 中国管理科学, 2006, 14(3): 65-70
- [12] 王兴伟, 王军伟, 黄敏. 基于狩猎搜索的可信 QoS 路由算法[J]. 东北大学学报: 自然科学版, 2012, 33(10): 1385-1389
- [13] 杜文峰, 贾维嘉, 王国军, 等. 基于 k 跳回溯机制的服务切换路由重建算法[J]. 小型微型计算机系统, 2006, 27(10): 1850-1855