文章编号：2095-2163(2019)03-0011-06 中图分类号：TP391 文献标志码：A

基于车辆异常行为的套牌车并行检测方法

康晨傲[1,2]，曾献辉[1,2]

（1 东华大学 信息科学与技术学院，上海 201620；2 数字化纺织服装技术教育部工程研究中心，上海 201620）

摘 要：面对套牌车问题，当前检测技术误判率居高不下，稽查部门需要消耗大量人力资源审核检测结果。为了降低套牌车检测的误判率和虚警率，提出了一种基于车辆异常行为的套牌车并行检测方法。确立数种与套牌车相关联的车辆异常行为因素，针对海量通行数据，利用分布式架构建立相应异常行为的挖掘算法模型，并对某市高速公路的真实海量交通流数据进行挖掘；利用 BP 神经网络算法建立模型并对数种异常行为挖掘结果进行训练，从而综合考虑多种异常行为因素得出套牌车检测结果。研究结果表明，该并行检测算法有效地降低套牌车误判率至 18%，可大幅度提高稽查人员的工作效率。

关键词：套牌车检测；车辆异常行为；大数据；并行；Spark；高速公路

Parallel detection approach of fake plates based on abnormal vehicle behavior

KANG Chen´ao[1,2], ZENG Xianhui[1,2]

(1 School of Information Sciences and Technology, Donghua University, Shanghai 201620, China;
2 Engineering Research Center of Digitized Textile & Fashion Technology Ministry of Education, Shanghai 201620, China)

【Abstract】 Faced with massive traffic data, the existing detection technology for fake plate vehicles have high false identification rate, which causing inspection department to consume lots of human resources on auditing the detection results. A new parallel detection approach of fake plates based on abnormal vehicle behavior is proposed, in order to decrease false identification rate and false alarm probability. This paper selects several abnormal behavior factors associated with fake plates, then establishes corresponding algorithmic models on distributed architecture to mine them. A BP neural network model is established to train the results of several abnormal behaviors obtained by mining the real massive traffic flow data of a city´s expressway through corresponding algorithmic models. Therefore, the detection results of fake plates can be obtained by taking various abnormal behaviors into account. The research results show that the parallel detection approach decreases false identification rate to 18%, which can improve the inspector´s work efficiency greatly.

【Key words】 fake plate vehicle detection; abnormal traffic behavior; big data; parallel; Spark; expressway

0 引 言

套牌车是指不法分子伪造和非法套取真牌车的号牌，使非法车辆在表面披上了“合法”的外衣。近年来，随着人们生活水平日益增高，机动车的数量日渐普及，为了逃避税费、逃避交通违法或处理查究、走私、拼装、报废和盗抢来的车辆等，套牌车这类违法现象也在不时发生。严重影响了正常的交通秩序，同时更侵犯了他人的合法权益，成为了交通治安管理中的突出问题，难点问题[1-2]。

目前，有关套牌车检测技术方面的主要研究有以下几类：

文献[3]对机动车车牌和牌照框进行了改造，以方便交警辨别套牌车。但这也使得车牌的制造和丢失后的重制成本大大提高。文献[4-5]分别运用 RFID 和 ZigBee 技术将车辆信息进行加密存入电子标签中，并将电子标签植入机动车。但这类方法面临着车主的支持、政府的大量资金投资和推广、电子标签的篡改和伪造等诸多问题。文献[6-7]介绍了基于视频图像识别技术的车脸识别套牌车检测技术。但这类方法在识别同色同型号的套牌车、光线不佳、天气恶劣等情况时，识别率的表现不佳。迄至目前，以卡口时间对比法为代表的软件套牌车检测技术因其成本低、且实效性强的优势被广泛使用。文献[8]提出了一种基于网格化监控的套牌车检测系统，按照一辆车不可能“同时”出现在 2 个地点的原理，自动完成套牌检测。文献[9-10]提出了基于 MapReduce 的并行式套牌车检测模型，解决大规模数据下的套牌车检测问题。文献[11]提出一种基于历史车牌识别数据集的套牌车并行检测方法 TP-Finder。文献[12]

作者简介：康晨傲(1993-)，男，硕士研究生，主要研究方向：大数据、数据挖掘；曾献辉(1974-)，男，副教授，主要研究方向：智能信息处理、智能优化问题、决策与分析。

通讯作者：曾献辉 Email：xhzeng@ mail.dhu.edu.cn

收稿日期：2019-03-06

提出了动态速度阈值的套牌车检测方法。

而随着图像识别技术、传感器技术、视频监控设备等信息采集、识别技术的发展和普及,使得人们可以通过各类采集设备获得多维的海量交通数据存入到数据库中[13]。据统计,就国内某市高速公路的交通数据采集来说,每月就要存储超过 100 G 的数据。如何基于这样的大规模车辆通行数据,尽可能准确地检测出套牌车成为了一个关键的挑战,而利用分布式架构的并行计算就成为了首选手段[13-15]。

虽然当前在针对海量交通数据的并行式处理技术方面已有一些研究成果,但在套牌车检测方面仍存在较高的误判率(把正常车辆或者超速车辆等误归为套牌车),稽查部门需要将大量的警力消耗在对于检测结果的人工审核和确认上。针对当前套牌车检测技术误判率、虚警率较高这个问题,本文提出了一种基于车辆异常行为的套牌车并行检测方法。研究时,首先确立数种与套牌车相关联的车辆异常行为因素,利用分布式架构建立相应异常行为的挖掘算法模型,并对海量交通流数据进行挖掘。利用 BP 神经网络算法建立模型,再对数种异常行为挖掘结果进行训练,从而综合考虑多种异常行为因素得出套牌车检测结果,以达到降低误判率的效果。

1 套牌车相关的车辆异常行为确立

1.1 车辆速度异常

正常车辆在较短的时间内行驶的距离应该在一个合理的阈值范围内,如果在较短时间内车辆行驶了远远超出阈值的距离,即发生了该车辆“同时”出现在了 2 个地方的时空矛盾现象,则该车辆可能出现套牌现象。

1.2 车辆出入异常

如果车辆在某一时刻出(进入)高速公路,而在该车牌号的车辆未进入(出)高速公路的情况下,又发现相同车牌号的车辆不久后在任意出入站点再次出(进入)高速公路。该车牌号的车辆连续多次出(入)高速公路,则该车辆可能出现套牌现象。

1.3 车辆频繁出现异常

如果车辆被套牌,在一定时间内因其会频繁经过拍摄卡口,其车牌号被记录在车流量大数据中的概率必然增加。因此可以针对某段时间内在车辆流大数据中出现频率较高的车牌号进行挖掘。

2 车辆异常行为挖掘模型建立

本文在 Spark 分布式计算框架上,进行车辆异常行为挖掘模型的搭建。并选取某市高速公路的车流量数据作为数据集,车流量数据记录了车辆每次经过各拍摄卡口时所被记录下的车辆信息,包括:车牌号(license)、拍摄时间(time)、拍摄卡口 ID(id)、车辆行驶方向(direction)。对此拟展开研究论述如下。

2.1 Spark 分布式计算框架

Spark 是一种通用而且高效的分布式内存计算框架,这是基于内存的计算方式,能有效提高在大数据环境下的计算效率,同时保证高容错性和高可伸缩性。Spark 采用 master-slave 结构,主节点上运行名称为 Master 的守护进程,负责对整个集群的监控和任务分配,从节点上运行名称为 Worker 的守护进程,接收来自主节点的任务并交由该结点上的 Executor 执行该任务。

弹性分布式数据集(RDD)是 Spark 的核心数据结构,RDD 内部的数据集合在逻辑和物理上都被划分成多个子集合,这样的每一个子集合将其称为分区(partitions),每一个分区数值的计算都是在一个单独的任务中进行,因此并行任务的个数,也是由 RDD 分区的个数所决定。

Spark 程序是由对 RDD 的操作构成的,如读取文件是一个 RDD,对文件计算是一个 RDD,结果集也是一个 RDD。程序先通过输入数据创建出一系列 RDD;再使用转化(Transformation)操作对一个 RDD 进行计算后,将其转化成另外一个 RDD,然后这个 RDD 又可以进行下一次转化;最后使用行动操作(Action)对 RDD 计算出一个结果,并把结果存储到外部存储系统(比如 HDFS、ORACLE 等)中,运行流程如图 1 所示。

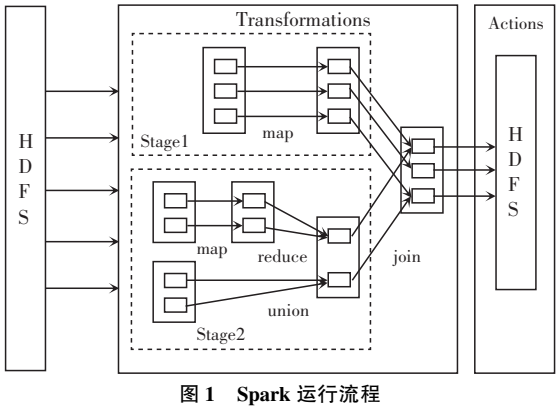

图 1 Spark 运行流程

Fig. 1 Running process of Spark

2.2 车辆速度异常挖掘模型建立

车辆速度异常判断依据:车牌号为 N 的车辆,在 T_1 时刻经过编号为 ID_A 的拍摄卡口 A。车牌号同为 N 的车在 T_2 时刻经过编号为 ID_B 的拍摄卡口 B。卡口 A 和卡口 B 之间的平均行驶距离为 Y 千米,计算该车辆的行驶速度为:

$$V = Y/(T_2 - T_1). \quad (1)$$

设置车辆速度最大阈值为 V_{max}。若 $V \leqslant V_{max}$，则说明该车辆从 A 卡口行驶到 B 卡口是符合正常情况的；但若 $V \geqslant V_{max}$，车速已超过最大阈值，则说明该车辆不可能在 $T_2 - T_1$ 的时间内从卡口 A 行驶到卡口 B。

例如，车牌号为浙 AB1234 车辆的部分车辆行驶记录如图 2 所示，该车辆于 08 点 00 分经过拍摄卡口 1011 处，又于 08 点 15 分经过拍摄卡口 1018 处，经查询得两拍摄卡口之间的平均行驶距离为 50 km，设置的阈值速度 V_{max} 为 140 km/h。根据公式(1)计算得该车辆的行驶速度 V = 200 km/h，超出阈值速度，则该车辆出现车辆速度异常现象。

卡口编号，车牌号，拍摄时间，行驶方向
…
1011，浙 AB1234，2018-12-01 08:00,1
…

卡口编号，车牌号，拍摄时间，行驶方向
…
1018，浙 AB1234，2018-12-01 08:15,1
…

卡口编号，卡口编号，卡口间行驶距离(米)
…
1011, 1018, 50000
…

图 2　车辆行驶记录

Fig. 2　Driving record of vehicle

并行式车辆速度异常挖掘模型的构建主要分为以下几个步骤：

(1)程序读取 Oracle 数据库中的预处理过后的车流量数据表生成 RDD，其中 RDD 的每个元素为每条车辆记录。

(2)将每条车辆记录转化为形如 < *license*, *info* > 的元组形式，作为 RDD 的每个元素。其中，*license* 为车牌号，*info* 为该条记录的车辆信息(*license*, *time*, *id*, *direction*)。

(3)按车牌号 *license* 对所有元素进行聚合操作，将车牌号相同的所有车辆信息 *info* 聚合在一起。即：将 RDD 的每个元素转化为 < *license*, *list* < *info* > > 的元组形式，其中 *list* < *info* > 为存储该车牌号所有车辆信息的集合。

(4)对集合 *list* 中的所有车辆信息 *info* 按照拍摄时间进行升序排序，将 RDD 的每个元素转化为形如 < *license*, *list_order* < *info* > > 的元组形式。

(5)对于集合 *list_order* 中车辆信息 *info* 的数量只有一条的元素，在 RDD 中删除该元素。

(6)针对每一个车牌号，依次从集合 *list_order* 中遍历取出相邻的车辆信息 *info*，并根据公式(1)进行异常行为判断：如果车辆速度超出阈值，则此车牌号对应的异常次数 *count_abSpeed* 加 1(初始为 0)。

(7)遍历完该车牌号对应集合 *list_order* 中所有车辆信息 *info* 后，如果异常次数 *count_abSpeed* 大于 3，则将该车牌号码、异常次数等相关异常信息存入车辆速度异常集合。

(8)重复步骤(6)、(7)，直到遍历完所有的车牌号。

(9)将车辆速度异常集合中的数据存储至 Oracle 数据库中。其大致过程如图 3 所示。

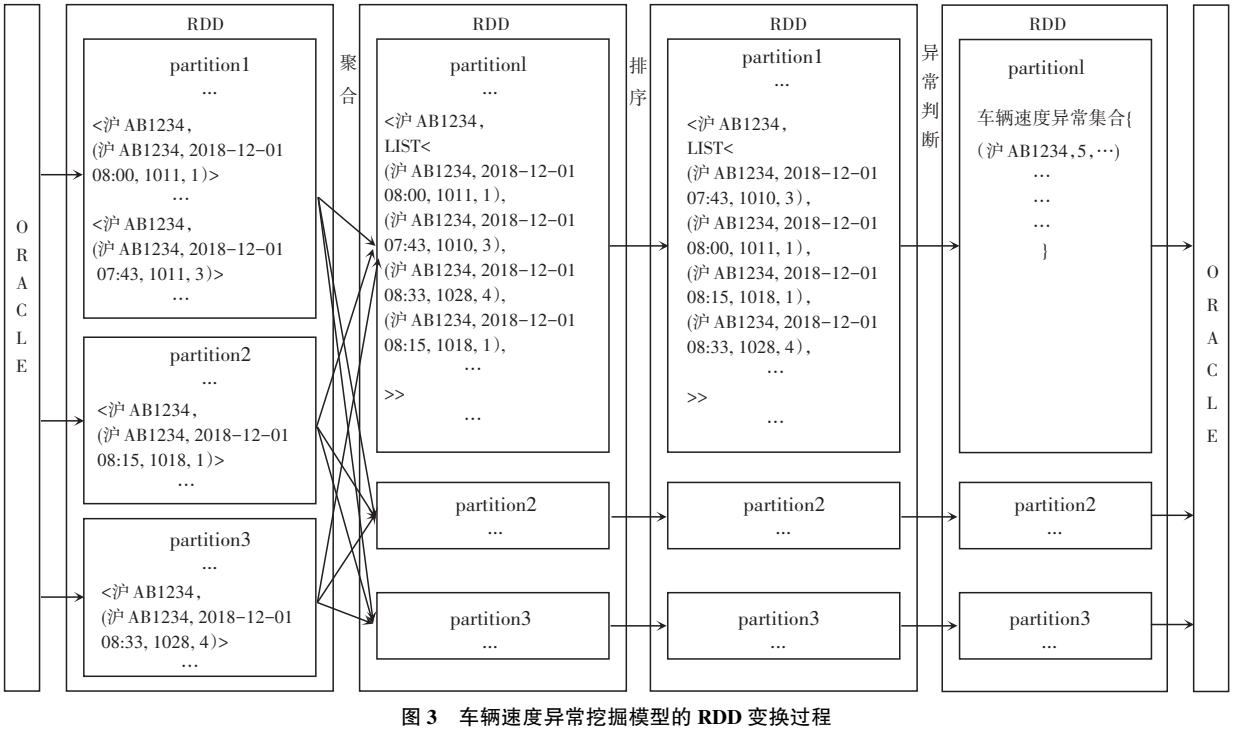

图 3　车辆速度异常挖掘模型的 RDD 变换过程

Fig. 3　RDD transform process of abnormal speed mining model

2.3 车辆出入异常挖掘模型建立

并行式车辆出入异常挖掘模型的构建主要分为以下几个步骤：

步骤(1)~步骤(5)与 2.2 节相同。

(6)删除车辆信息 *info* 中车辆行驶方向为 1 或者 2 的 *info*,仅保留为 3 和 4 的 *info*。车辆行驶方向的含义见表 1。

(7)针对每一个车牌号 *license*,依次从集合 *list_order* 中遍历取出相邻车辆信息 *info* 中车辆行驶方向 *direction* 和拍摄时间 *time*。当相邻车辆信息中车辆行驶方向同为3或者4,且时间差在1 h内时,将此车牌号对应的异常次数 *count_abInout* 加 1。

(8)遍历完该车牌号对应集合 *list_order* 中所有车辆信息 *info* 后,如果异常次数 *count_abInout* 大于5,将该车牌号码、异常次数等相关异常信息存入车辆出入异常集合。

(9)重复步骤(7)、(8),直到遍历完所有的车牌号。

(10)将车辆出入异常集合中的数据存储至 Oracle 数据库中。

表 1 车辆行驶方向含义

Tab. 1 Meaning of vehicle driving direction

车辆行驶方向	含义
行驶方向 =1	车辆上行
行驶方向=2	车辆下行
行驶方向=3	车辆从站点进入高速
行驶方向=4	车辆从站点出高速

2.4 车辆频繁出现异常挖掘模型建立

并行式车辆频繁出现异常挖掘模型的构建主要分为以下几个步骤：

(1)程序读取 Oracle 数据库中的预处理过后的车流量数据表,生成 RDD。

(2)将每条车辆记录转换为 < *license* - *date*, 1 > 的元组形式,来作为 RDD 中的每个元素。其中,*date* 为从拍摄时间中提取出的日期(例如 2018/12/01)。

(3)按车牌号 *license* 和拍摄日期 *date* 对所有元素进行聚合等操作,将 RDD 的每个元素转化为 < *license* - *date*, *frequency* > 的元组形式。其中,*frequency* 为该车牌号在一天内被记录在数据集中的次数。

(4)将 RDD 的元素按 *frequency* 进行降序排序,并选取 *frequency* 排名前 10% 的车牌号,即一天内出现频率最高的 10% 的车辆。

(5)将步骤(4)中选出的车牌号的各天出现次数进行累加,得到 < *license*, *count* > 的形式的元组,并将车牌号和总被记录次数存储至 Oracle 数据库中。

3 结果与分析

3.1 实验环境与数据集

本文的实验环境部署在由 4 台 4 核 8 GB 内存的服务器组成的 Spark 集群环境上。选取其中一台作为 Master 节点,其他三台作为 Work 节点。各节点的软件部署见表 2。

表 2 各节点软件部署

Tab. 2 Software deployment of each node

软件	版本	说明
Centos	7.5	操作系统
JDK	1.8	JAVA 开发环境
Hadoop	2.73	Hadoop 组件
Spark	2.31	分布式计算框架
Oracle	11g	数据库

3.2 数据集预处理

本文研究采用实验数据集为,某市高速公路 2018 年 12 月约 800 个拍摄卡口所录入的真实车辆流数据以及各拍摄卡口信息。数据已经过脱敏处理,车辆流数据规模约在 9 000W 条。

各拍摄卡口传输的数据通常含有属性冗余特征,各种天气因素环境因素、车速过快等,也会引起拍摄时的数据缺失、数据异常、属性不完整等问题。因此在使用数据前,需要进行冗余属性剪枝、异常数据剔除、缺失属性补全等操作。预处理过后的数据集存入 Oracle 数据库中。

车流量数据记录了车辆每次经过各拍摄卡口时被记录下的车辆信息,选取 12 月 1 日~15 日的数据集,经过预处理过后得到的部分车流量数据表 P1 见表 3。通过拍摄卡口信息可以查询到各个拍摄卡口之间的行驶距离等信息,经过预处理过后得到的部分拍摄卡口距离索引表 P2 见表 4。

表 3 高速公路车流量数据表 P1

Tab. 3 Traffic flow data table P1

拍摄时间	车牌号	拍摄卡口 ID	行驶方向
2018/12/1 12:00:01	浙 K1＊＊＊＊	1525	3
2018/12/1 12:00:02	苏 HC＊＊＊＊	3245	1
2018/12/1 12:00:02	京 A5＊＊＊＊	2325	4
2018/12/1 12:00:03	浙 CD＊＊＊＊	1539	2
2018/12/1 12:00:04	闽 H1＊＊＊＊	3241	3
…	…	…	…

表 4　高速公路拍摄卡口距离索引表 P2

Tab. 4　Index table P2 of distance between bayonets

卡口编号	卡口编号	卡口间行驶距离/m
1041	1015	101 539
1047	1015	100 879
1623	1015	147 000
1621	1015	132 560
…	…	…
1033	1017	72 000
1036	1017	87 200
…	…	…

3.3　预测结果分析

本文采用第 2 节建立的 3 种车辆异常行为挖掘模型，在 Spark 分布式计算框架上对数据集 P1 进行数据挖掘。将高速公路 12 月警方真实稽查到的套牌车名单作为标签属性，与异常挖掘结果构成数据集 P3，部分数据集见表 5。

表 5　异常行为挖掘结果数据集 P3

Tab. 5　Result data set P3 of abnormal behavior

车牌号	速度异常	出入异常	记录次数	稽查结果
浙 DH＊＊＊＊	13	45	154	1
沪 A2＊＊＊＊	12	47	227	1
…	…	…	…	…
沪 CB＊＊＊＊	7	11	42	1
豫 P8＊＊＊＊	6	5	53	0
…	…	…	…	…

注：稽查结果为 1 表示确认为套牌车，为 0 表示未确认为套牌车

使用 Spark 的 MLlib 组件构建 BP 神经网络模型，设置模型的输入层节点数为 3，中间层节点数为 4，输出层节点数为 2，传递函数为 ReLU 函数。选取数据集 P3 作为模型的数据集，并将数据集 P3 进行归一化操作。为了增强模型的泛化能力，选取数据集中 70%作为训练集，15%作为验证集，15%作为测试集。

待模型经训练完成后，选取 P3 测试集和 P3 全量数据集进行测试。为了提高稽查部门的效率，每半个月向稽查部门提交 200 名最可疑的套牌车辆名单，即分别选取测试结果中归类为套牌车的前 30 位、140 位进行验证（按车辆速度异常次数的降序排序），对于全量集准确率约为 93%，测试集准确率约为 81%。

为了进一步验证算法性能，本文选取传统的卡口时间对比法进行对比实验，该方法认为同一个车牌不能在短时间内出现在 2 个距离较远的位置，运用最大速度不可达算法对卡口历史监测数据进行处理，从而检测套牌车。选取 12 月 15 日～30 日的车流量数据集，采用 2 种方法分别对 5 天、10 天、15 天的数据量进行测试，选取测试结果中归为套牌车的前 70 位、130 位、200 位（按车辆速度异常次数的降序排序）作为最终结果，对比试验结果见表 6。

表 6　对比试验结果

Tab. 6　Comparison of test results

测试时间/d	卡口时间法	本文算法
5	55%	71%
10	61%	78%
15	65%	82%

由此可见，与仅用卡口时间对比这单一手段检测套牌车相比，本文方法考虑了多种异常因素，将误判率降至约 18%左右，能更有效地降低套牌车检测的虚警率。另外，高速公路上大部分套牌车辆一般为有目的性的长期作案，随着数据量的增多也能更好地提高检测效果。而有效的学习能力使本文方法具有较好的适应性，对提升高速公路稽查水平具有很大帮助。

4　结束语

针对当前对于海量交通数据的套牌车检测技术误判率高的问题，本文提出了一种基于车辆异常行为的套牌车并行检测方法。与传统方法相比，本文提出的方法借助分布式架构，实现了对海量数据的挖掘。并建立多种相应异常行为挖掘模型，考虑多种异常行为因素，以求达到降低误判率的效果。最后通过实际数据验证方法的有效性，从预测结果中得出，本文所提出的算法对于套牌车检测的误判率降低有着不错的效果，对于降低了稽查部门工作强度，减少了大量人力审核资源方面，取得了令人满意的效果。

不过，本文提出的方法还有一定欠缺。在套牌车判定过程中，对于阈值（速度阈值、各异常次数阈值等）的选择可能会对结果造成不小的影响，在后续研究中，将考虑采用机器学习方面的算法来解决动态阈值方面的问题，提高算法的适应性能力。

参考文献

[1] 徐鑫华. 浅谈“套牌车”产生的原因、危害及对策[J]. 安全与健康，2007(10)：22-23.

[2] 陈媛媛. 浅析套牌车辆的危害及整治对策[J]. 安全生产与监督，2010(3)：44-45.

（下转第 22 页）