ZTE COMMUNICATIONS
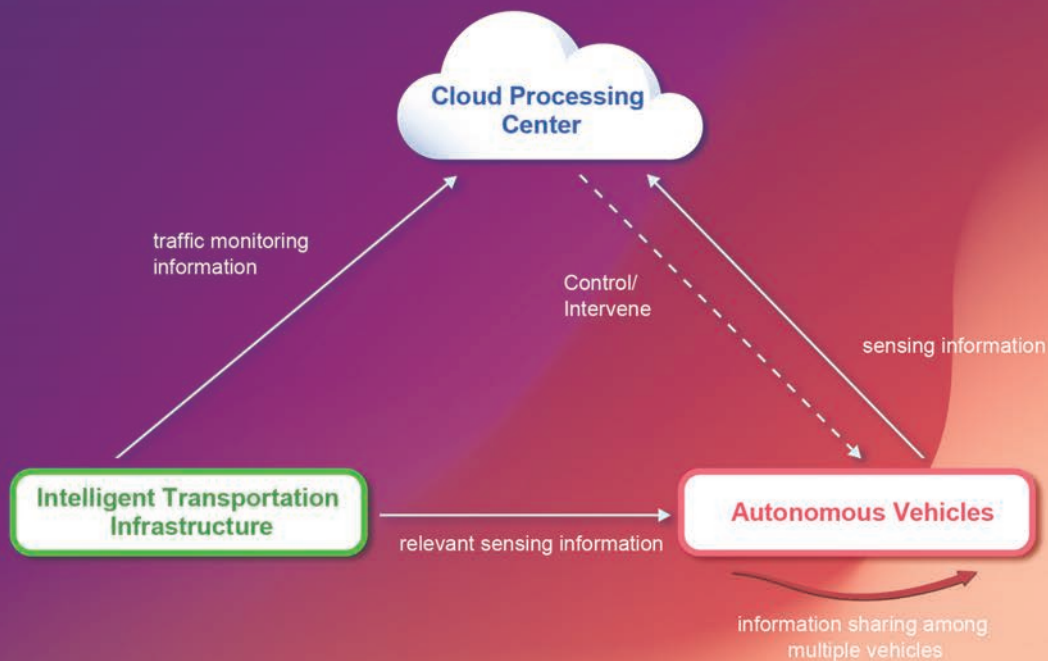
中兴通讯技术(英文版)

SPECIAL TOPIC:
Machine Learning for Wireless Networks

COVER PAPER:
Cooperative Intelligence for Autonomous Driving (P. 44)

CONTENTS

Special Topic

Machine Learning for Wireless Networks

CONTENTS

ZTE COMMUNICATIONS   June 2019   Vol. 17 No. 2 (Issue 66)

Review

Research Paper

Editorial: Special Topic on
Machine Learning for Wireless Networks

WANG Zhengdao received his bachelor's degree from the University of Science and Technology of China (USTC) in 1996, the M.S. degree from the University of Virginia, USA in 1999, and Ph.D. degree from the University of Minnesota, USA in 2002. From 2002, he has been on the faculty of the Department of Electrical and Computer Engineering in the Iowa State University, USA, where he is currently a professor. His interests are in the areas of signal processing, communications, information theory, machine learning and their applications.

He received the IEEE Signal Processing Society Best Magazine Paper Award in 2003, the IEEE Marconi Paper Prize Award in Wireless Communications in 2004, and the EURASIP Journal on Advances in Signal Processing Best Paper Award in 2008. He is an IEEE Fellow of class 2016.

He served as an Associate Editor for several journals and an organizer for several conferences, including IEEE GLOBECOM, IEEE GLOBALSIP, Asilomar Conference, and WUWNET. He served as the Regional Director at Large of the IEEE Signal Processing Society for Regions 1−6 (US) in 2017 and 2018. He is serving as a TPC co-chair for IEEE SPAWC 2020.

Guest Editor

Success in applying deep learning in speech recognition had triggered a resurgence of interests in deep learning, starting around 2010. Since then, deep learning has been applied with remarkable results to many problems, such as image recognition, speech and image synthesis, natural language processing, finance and artificial intelligence such as game playing. The success owes much to the availability of large amount of data, improved computational capabilities, and advances in algorithms.

As a specific form of machine learning, the success of deep learning demonstrates the tremendous power and versatility of machine learning in providing alternative solutions to a large array of problems. From a communication and networking engineer's perspective, it is only natural to consider applying machine learning, and deep learning more specifically, to engineering problems in communication and networking. There is a somewhat natural divide among the researchers and practitioners in the area. While some researchers have embraced machine learning enthusiastically, viewing it as yet another powerful tool to have in the tool chest, others are doubtful as to how much benefit these tools can offer, insisting on a model-based approach that has been very successful so far.

It is our hope that this special issue can add to the debate and discussion between these schools of thoughts, by providing a few more data points which demonstrates some more successes of using machine learning in solving engineering problems, and in doing so, may also reveals some shortcomings of such approaches (hopefully to a smaller extent), for example in the requirement of large amount of data, the complexity of the algorithms, and the lack of interpretability of the obtained designs.

The special issues includes a paper by Zöchmann, titled "A Framework for Active Learning of Beam Alignment in Vehicular Millimetre Wave Communications by Onboard Sensors", which proposes a Doppler bearing tracking regression algorithm that tracks the strongest multipath components for antenna beam alignment in vehicular millimetre wave communications. The obtained geometric information enables active learning algorithm to select interesting training data. The papers "A Novel Real-time System for Traffic Flow Classification and Prediction" and "A Network Traffic Prediction Method Based on LSTM" apply machine learning to the problem of network traffic prediction, where the former one uses dynamic time wrapping, XGBoost and wavelet transform, while the latter uses more recent long short-term memory (LSTM) networks. The paper "Potential Off - Grid User Prediction System Based on Spark" advertises a comprehensive machine learning based system for predicting user churning behaviors. Last but not least, the paper "Detecting Abnormal Start - Ups, Unusual Resource Consumptions of the Smart Phone: A Deep Learning Approach" proposes to use LSTM for detecting cell phone application behaviors, which although is not a communication/network problem, fits well the general theme of applying machine learning to engineering design.

We hope the readers will enjoy the special issue and can decide for themselves which side of the fence they choose to reside. It is entirely acceptable to tear down the fence and adopt a model-based machine learning approach as well.

A Framework for Active Learning of Beam Alignment in Vehicular Millimeter Wave Communications by Onboard Sensors

Erich Zöchmann

(Christian Doppler Laboratory for Dependable Wireless Connectivity for the Society in Motion, Institute of Telecommunications, TU Wien, 1040 Vienna, Austria)

Abstract: Estimating time-selective millimeter wave wireless channels and then deriving the optimum beam alignment for directional antennas is a challenging task. To solve this problem, one can focus on tracking the strongest multipath components (MPCs). Aligning antenna beams with the tracked MPCs increases the channel coherence time by several orders of magnitude. This contribution suggests tracking the MPCs geometrically. The derived geometric tracker is based on algorithms known as Doppler bearing tracking. A recent work on geometric-polar tracking is reformulated into an efficient recursive version. If the relative position of the MPCs is known, all other sensors on board a vehicle, e.g., lidar, radar, and camera, will perform active learning based on their own observed data. By learning the relationship between sensor data and MPCs, onboard sensors can participate in channel tracking. Joint tracking of many integrated sensors will increase the reliability of MPC tracking.

Keywords: adaptive filters; autonomous vehicles; directive antennas; doppler measurement; intelligent vehicles; machine learning; millimeter wave communication

1 Introduction

Millimeter wave (mmWave) frequency bands have been a candidate for vehicular communication for several decades [1]–[3]. MmWave train-to-infrastructure path loss was measured in [2], while the transmission behaviour of mmWave for communication between vehicles was examined in [1]. Recent advances in mmWave circuit technology have aroused interest in mmWave vehicular communication [3] and in joint vehicular communication and radar [4]. MmWaves offer large bandwidths and enable raw data exchange between vehicles [5]. The main problems with vehicular mmWave communication are the direct proportionality of the maximum Doppler shift and the carrier frequency as well as the beam alignment challenge in the dynamic environment. In [6] and [7], however, it

has been shown theoretically that directional antennas intended for mmWaves function as spatial filters. The Doppler effect and thus the time selectivity is drastically reduced by beamforming. This is shown experimentally in [8] and [9]. There seems to be a consensus that channel tracking tackles the second challenge of the dynamic environment [10]–[21]. Channel tracking is the process of causally estimating the current or future direction of the line-of-sight (LOS) component or other strong multipath components (MPCs) based on previous measurements. The main advantage of channel tracking is the extended coherence time after successful beamforming. The channel coherence time of the beam aligned channel is several orders of magnitude longer than that for omnidirectional reception [7]. A subsequent channel estimation therefore runs on a coarser time grid.

The work in [10] adopts the idea and formalism of [21] and applies them directly to THz lens antennas. Extended Kalman filters are used in [11], [18], and [19] to track the beam directions based on channel gain measurements. In [15], domain

A Framework for Active Learning of Beam Alignment in Vehicular Millimeter Wave Communications by Onboard Sensors | *Special Topic*

Erich Zöchmann

knowledge is used and it is argued that the road implicitly determines the direction in which a vehicle is expected. Beam training is avoided by using this geometric prior knowledge. Assuming a constant angular acceleration that is motion along circles, [20] proposes an algorithm based on the unscented Kalman filter. Probabilistic beam tracking is suggested in [16]. Moreover, in [13] and [14] the stochastic Newton method is used, and these algorithms surpass IEEE 802.11ad based approaches and compressive sensing based approaches [17]; the work in [13] and [14], shows good performance for angular velocities of up to 5°/s.

In [22], it was first proposed to utilize the Doppler information for mmWave beam tracking. Measurements in [23] clearly demonstrate that interacting objects, such as overtaking cars, produce distinguishable MPCs in the Doppler profile. The proposed algorithm herein, exploiting Doppler information, is assessed in scenarios where the angular velocity exceeds 100°/s for a short duration.

This contribution proposes to track the MPCs geometrically given quantized angular (azimuth) measurements and noisy Doppler observations. The quantized angular information is obtained by an analog or hybrid beamforming array or a dielectric lense [24]. "Geometric" refers to the (x,y) coordinates originating in the antenna array and the relative velocity $(\dot{x},\dot{y})$ to the receiver motion. We assume that the transmitter, the receiver, and the interacting objects move without acceleration. Under these assumptions, algorithms performing target-motion analysis by means of Doppler-bearing measurements [25]–[27] are directly applicable. The work in [25]–[28] proposes a formulation called "pseudolinear." Pseudolinear refers to a formulation where the nonlinearities are either hidden in a measurement (regression) matrix or are lumped within the noise term. This leads to the undesirable consequence of noise correlation of the measurements and the measurement matrix, eventually leading to biased solutions [26]. An early work [25] removes this bias by the method of instrumental variables. Due to a potential divergence of the instrumental variables approach [27], later work [26], [27] employs the method of total least squares. To apply total least squares, error covariance matrices must be known . The proposed approach is inspired by [27], but does not need knowledge about the error covariances.

In addition to the excellent angular tracking performance of the proposed algorithm, the obtained geometric information of the MPCs can be utilized to learn the MPCs from other sensors on board of automated vehicles [5]. The concept of using external information for improved channel estimation was recently reintroduced, see [29] and [30] and the reference therein. Machine learning for configuring wireless links has also been proposed in the context of WLAN and mobile communications [31]–[33]. The actual machine learning implementation is not within the scope of this contribution. This contribution focuses on a framework for active learning of beam alignment. This paper is an extended version of [34].

(1) Brief Review of Geometric Tracking or Wireless Positioning:

Ground-based radio-frequency localization has become an established technique. Based on known anchor positions, techniques such as fingerprinting, hop counts, receive signal strength, time-(difference)-of-arrival, frequency-difference-of-arrival, and angle-of-arrival are at hand [35]. Knowing the position of the communication partner is extremely valuable for the task of beam alignment [36]. In [37], a mmWave base station was equipped with a 360° camera; both positional information sources—vision and the mmWave link—were fused to enhance the precision. The situation changes however once vehicle-to-vehicle communication is considered. There, mainly GNSS positions of communication partners are exchanged by low-rate messages [38]. Future automated self-driving cars will be equipped with a plurality of sensors and will thereby perform massive sensing [5]. The smart use of all of these sensors will renders it possible to determine the position of communication partners solely by onboard sensors.

(2) Notation:

Matrices Z and vectors z are denoted by bold letters. The all zeros vector (matrix) is expressed by 0 and the identity matrix is expressed by I. The Euclidean norm is symbolized by $\|\cdot\|$. A quantity defined with a start index i and stop index k is indicated via the subscript $(\cdot)_{i:k}$. Estimated quantities are marked with $(\hat{\cdot})$. The four-quadrant inverse tangent is denoted by $\arctan(\cdot,\cdot)$. The dagger $(\cdot)^{\dagger}$ is used for pseudo inverses and $(\cdot)^{T}$ is used for transposition.

2 Active Learning by Onboard Sensors

The idea behind active learning is to actively select the "optimal" training data. For some applications statistically optimal choices are computable [39]. Selective sampling [40] is a rudimentary form of active learning and especially suited for problems where the cost of labelling is high. The survey paper [41] provides a good introduction to active learning. According [41]: "The key idea behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer labelled training instances if it is allowed to choose the data from which is learns."

On board of automated (self-driving) cars, there will be sensors such as global navigation satellite systems (GNSSs), automotive radars (for automatic cruise control and collision detections), lidar (for measuring distances to other objects), and 360° camera vision systems. All these sensors have in common that they track objects via target states [42]. At the simplest, this target state consists of the relative (x,y) position and the relative velocities $(\dot{x},\dot{y})$.

Due to the high-resolution of lidar, radar, and vision, self-driving cars produce several gigabytes of data per second and hence hundreds of terabytes per day [43], [44]. Processing all these data for the indented use case of driving poses already a

Special Topic | A Framework for Active Learning of Beam Alignment in Vehicular Millimeter Wave Communications by Onboard Sensors

Erich Zöchmann

challenge and more and more tasks are already shifted towards fog and cloud computing units [45], [46]. To use these gigabytes of data for the tracking of MPCs, every tracked object of the onboard sensors must be labelled as "MPC" or "no MPC" (pedestrian, non-communicating car, static objects, etc.). This leads to high labelling efforts and to a huge amount of training data where most of labels will be "no MPC".

The key idea is now to exploit the geometric position of the MPC and thus to only label those targets that are in the vicinity of the MPC. The process of associating MPCs to "targets" is illustrated with black circles in **Fig. 1**. Instead of human (or any other oracle) labelling there is an active choice of the system which targets to consider for learning the beam alignment. After a successful learning phase, all sensors on board should later do the channel tracking. By using machine learning, one can eliminate or significantly reduce the beam measurements needed for the currently proposed tracking algorithm. If for all of these target states it is known whether they belong to the LOS component or to a specular reflection, the onboard sensors will track the MPCs.

In this sense this paper provides an algorithm which determines the geometric positions of the communication partners in order to label them as interesting training samples.

3 Measurement, Regression, and Projection Model

The regression model is based on the model proposed in [27]. The main idea of [27] is to track non-accelerating objects on linear trajectories in polar coordinates; target motion analysis in polar coordinates yields a smaller bias than in rectangular coordinates. The regression model is hence formulated in polar coordinates. This idea is illustrated in **Fig. 2**. The original tracking problem of [27] uses a running reference (blue). Thereby at each time the current state is estimated. This ap-
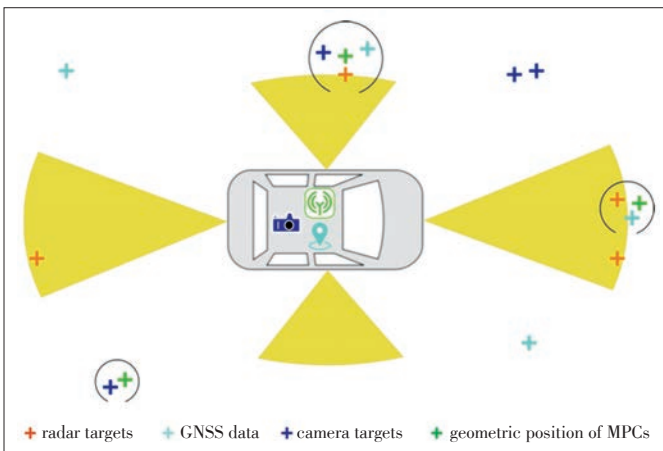
▲ Figure 2. Geometric relationship of all variables used for the algorithm. The MPC to track is marked as squqre. The employed array geometry (uniform circular array) is sketched as well.

proach produces an increasing system of equations, anew, at any time. In contrast to [27], the proposed algorithm will use a fixed reference (red) and gather only one new equation per time step. Thereby the estimate of the initial state is refined and its accuracy is improved over time, as in [25]. Through this reformulation, the initial state-vector is estimated recursively. The state vector at current and future times is predicted by a projection.

3.1 Quantized Angular Measurements by the UCA Codebook

For target motion analysis a noisy bearing (angular) observation is assumed where the noise is usually modelled Gaussian [27]. In this study, however, quantized angular observations will occur. A 60 GHz uniform circular array (UCA) with $N = 64$ elements equidistantly spaced on a radius of $r_{UCA} = N/2 \cdot \lambda/2 \approx 8$ cm is used. The half power beam width is $\theta_{3dB} \approx 2\pi/N \approx 6°$. The UCA is inherently symmetric in its azimuthal resolution. In contrast to uniform planar arrays, the UCA beam pattern does not change with the pointing direction. To save cost, analog precoding (beamforming) with 4 bit RF phase-shifters is employed. The phase shifts are pre-computed in a codebook spaced by $\theta_{3dB}/2$ which gives $2\pi/[(2\pi/N)/2]$ $= 2 \cdot N = 128$ codebook entries. Beampattern of the UCA are shown in **Fig. 3**.

3.2 Regression Model

The angle spanning from initial azimuth ϕ_i to the current

▲ Figure 1. Active learning example. The estimated state vectors of the communication link (green crosses) label the data from other sensors as a valid MPC. Therby active learning of possible beam direction from other sensors is rendered possible.

A Framework for Active Learning of Beam Alignment in Vehicular Millimeter Wave Communications by Onboard Sensors *Special Topic*

Erich Zöchmann

▲ Figure 3. Beampattern of the proposed uniform circular array with $N = 64$ elements and 4 bit quantization of all steering vectors. The desired array pattern is marked in green; the array factor for this direction (0) is marked with a green dot; the neighbouring codebook pattern are drawn in black. The uniform circular array (UCA) pattern is inherently symmetric for all directions; illustrated by the red pattern pointing in opposite direction.

azimuth ϕ_k is denoted by $b_{i:k}$. In Fig. 2, w.l.o.g. i is set to zero. The range (at time k) is denoted by r_k. The time intervals are denoted by $t_{i:k}$. The Doppler relevant angle at time k, that is α_k, is measured from the velocity vector v to the radial speed component. The basis equation is the sine law evaluated for each observation time $k > i$:

$$\frac{r_i}{\sin(\alpha_k)} = \frac{vt_{i:k}}{\sin(b_{i:k})}. \tag{1}$$

Now Equ. (1) is reformulated into a so called pseudo-linear formulation [28]:

$$\underbrace{\left[\sin(b_{i:k}), -t_{i:k}\cos(b_{i:k}), t_{i:k}\sin(b_{i:k})\right]}_{a_{B,i:k}^T}\underbrace{\begin{bmatrix} r_i \\ v\sin(\alpha_i) \\ v\cos(\alpha_i) \end{bmatrix}}_{x^i} = 0, \tag{2}$$

where all nonlinearities are regressors now. For each observation-time k, one equation in the form of Equ. (2) is obtained. This is written compactly in matrix-vector notation:

$$A_{B,i:k}x^i = 0. \tag{3}$$

The solution to Equ. (3) is not unique. Next Doppler-shift observations ν_k are exploited. These are calculated by

$$\nu_k = \frac{v}{\lambda}\cos(\alpha_k) = \frac{v}{\lambda}\cos(\alpha_i - b_{i:k}). \tag{4}$$

Equ. (4) is re-written into the same form as Equ. (2):

$$\underbrace{\left[0, \frac{1}{\lambda}\sin(b_{i:k}), \frac{1}{\lambda}\cos(b_{i:k})\right]}_{a_{D,i:k}^T}\underbrace{\begin{bmatrix} r_i \\ v\sin(\alpha_i) \\ v\cos(\alpha_i) \end{bmatrix}}_{x^i} = v_k. \tag{5}$$

This leads again to a system of equations in form of

$$A_{D,i:k}x^i = \nu_{i:k}. \tag{6}$$

One finally arrives at the augmented system of equations:

$$\begin{bmatrix} A_{B,i:k} \\ A_{D,i:k} \end{bmatrix} x^i = \begin{bmatrix} 0 \\ \nu_{i:k} \end{bmatrix}. \tag{7}$$

This system of equations has a unique solution and is called "Doppler-bearing tracking" in the literature [25]–[27]. In this contribution, Equ. (7) is solved via the method of least squares (LS). Note that Equ. (7) is equivalent to

$$\begin{bmatrix} A_{B,i:(k-1)} \\ A_{D,i:(k-1)} \\ ---- \\ a_{B,i:k}^T \\ a_{D,i:k}^T \end{bmatrix} x^i = \begin{bmatrix} 0 \\ \nu_{i:(k-1)} \\ --- \\ 0 \\ \nu_k \end{bmatrix}. \tag{8}$$

In Equ. (8), the previous observations are separated from the current one. This structure allows for a recursive least squares (RLS) implementation. The recursive estimate of x^i at time k will be denoted by $\hat{x}_{i:k}^i$, in the sequel.

The angular separations $b_{i:k}$ in Equs. (1)–(8) are not known and must be estimated. The variable $b_{i:k}$ has to be replaced by $\hat{b}_{i:k}$, the estimated quantity, in the equations above. Such measurement equations are called "errors-in-variables model" in the statistics literature [47]. The estimation of $b_{i:k}$ is initially done by training sequences. For each time k, the current azimuth angle $\hat{\phi}_k$ is estimated by aid of beam sweeping, that is, a codebook scan. All possible beams are iterated and the codebook entry (azimuth direction) with largest receive power is selected. Next we obtain $\hat{b}_{i:k} = \hat{\phi}_k - \hat{\phi}_i$. Later on, onboard sensors might provide the estimate of $\hat{\phi}_k$ and codebook scans can be avoided or at least performed less frequently.

With the estimate of the initial state-vector $\hat{x}_{i:k}^i$, we calculate the initial (x, y) position and the velocity vector $(\dot{x}, \dot{y})$ based on the polar representation. The range $\hat{r}_i$ is the first element of the initial state-vector estimate, that is $\hat{x}_{i:k}^i(1)$. The azimuth angle $\hat{\phi}_k$ is estimated through designated pilots. The velocity $\hat{v}$ is calculated through the initial state-vector estimate and the angle of the velocity vector to the x-axis φ is calculated through the initial state-vector estimate as well:

$$\hat{v} = \sqrt{(\hat{x}_{i:k}^i(2))^2 + (\hat{x}_{i:k}^i(3))^2},$$
$$\hat{\varphi} = \hat{\alpha}_i + \hat{\phi}_i = \arctan\{\hat{x}_{i:k}^i(2), \hat{x}_{i:k}^i(3)\} + \hat{\phi}_i. \tag{9}$$

3.3 Projection Model

A suitable projection from the initial state vector to arbitrary time points was recently derived from [48]. The assumption of a linear trajectory with non-accelerating MPCs leads to a static velocity vector. Only the range $\hat{r}_i$ and the azimuth angle $\hat{\phi}_i$ need to be projected to current (or future) time points k. The range projection $\hat{r}_k$ is calculated through [48]:

Special Topic | A Framework for Active Learning of Beam Alignment in Vehicular Millimeter Wave Communications by Onboard Sensors

Erich Zöchmann

$$\hat{r}_k = \sqrt{\hat{r}_i^2 + (\hat{v}t_{i:k})^2 + 2\hat{r}_i\hat{v}t_{i:k}\cos\hat{\alpha}_i} =$$
$$\sqrt{(\hat{\boldsymbol{x}}_{i:k}^i(1))^2 + t_{i:k}^2((\hat{\boldsymbol{x}}_{i:k}^i(2))^2 + (\hat{\boldsymbol{x}}_{i:k}^i(3))^2) + 2t_{i:k}\hat{\boldsymbol{x}}_{i:k}^i(1)\hat{\boldsymbol{x}}_{i:k}^i(3)}, \tag{10}$$

and the azimuth projection $\hat{\phi}_k$ is calculated through [48]

$$\hat{\phi}_k = \hat{\phi}_i + \arctan(\hat{v}t_{i:k}\sin\hat{\alpha}_i, \hat{r}_i + \hat{v}t_{i:k}\cos\hat{\alpha}_i) =$$
$$\hat{\phi}_i + \arctan(t_{i:k}\hat{\boldsymbol{x}}_{i:k}^i(2), \hat{\boldsymbol{x}}_{i:k}^i(1) + t_{i:k}\hat{\boldsymbol{x}}_{i:k}^i(3)). \tag{11}$$

4 Proposed Robust Recursive Tracker

The utilized regression model falls in the following time-variant state-space model:

$$\boldsymbol{x}_{i:k}^i = \boldsymbol{x}_{i:(k-1)}^i + \boldsymbol{u}_k, \tag{12}$$

$$\underbrace{\begin{bmatrix} 0 \\ \boldsymbol{\nu}_k \end{bmatrix}}_{y_k} = \underbrace{\begin{bmatrix} \boldsymbol{a}_{B,i:k}^T \\ \boldsymbol{a}_{D,i:k}^T \end{bmatrix}}_{A_k}\boldsymbol{x}_{i:k}^i + \boldsymbol{n}_k, \tag{13}$$

where $\boldsymbol{u}_k$ is a process noise or driving disturbance with unknown distribution. The first component of the noise vector $\boldsymbol{n}_k$ stems from the quantization noise of the codebook based angle estimation. The second component is the measurement noise of the Doppler observation and is assumed to be i.i.d. zero mean Gaussian, that is, $\boldsymbol{n}_k(2) \sim N(0, 100^2)$. The high standard deviation in the Doppler noise term (100 Hz) already takes into account that current mmWave equipment suffers greatly from phase noise, see for example the measurement results in [8] and [23] or **Fig. 4** of this contribution. The assumption of zero mean Doppler noise is well justified as automated cars can use GNSS-disciplined oscillators, so that the carrier frequency off-
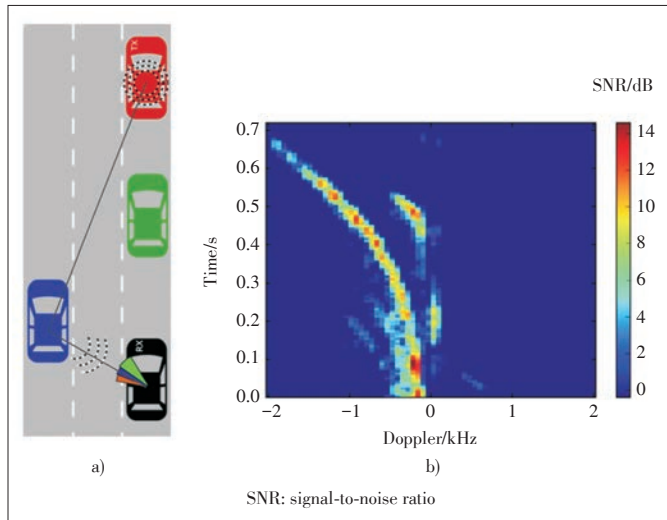
▲Figure 4. Blocked manoeuvre: a) sketch of the scenario. The green car block a direct communications between the red TX and black RX car; b) the Doppler shift estimate for this scenario obtained from a real world experiment in [9] and [23].

set between cars becomes very small. To increase robustness against quantization effects of the codebook and against the geometry dependent structure of $\boldsymbol{A}_k$, an $\mathcal{H}_\infty$ filter with a finite time horizon [49], [50] is applied. The objective of an $\mathcal{H}_\infty$ filter is to keep the error relation below bounded:

$$\sup_{\hat{\boldsymbol{x}}_0^i, \boldsymbol{u}, \boldsymbol{n}} \frac{\sum_{k=i}^N \|\hat{\boldsymbol{x}}_{i:k}^i - \boldsymbol{x}_{i:k}^i\|^2}{\|\hat{\boldsymbol{x}}_0^i - \boldsymbol{x}_0^i\|^2 + \sum_{k=i}^N \|\boldsymbol{u}_k\|^2 + \sum_{k=i}^N \|\boldsymbol{n}_k\|^2} < \gamma^2. \tag{14}$$

The vector $\hat{\boldsymbol{x}}_0^i$ denotes the initial guess of the state vector (in the simulations $\hat{\boldsymbol{x}}_0^i \equiv 0$, $\boldsymbol{u}_k \equiv 0$, and $\gamma = 2$). The $\mathcal{H}_\infty$ filter has a higher error floor and a higher complexity than the plain RLS solution. As a quantized codebook is used, subsequent measurements potentially provide equal azimuth angle measurements. Due to the structure of $\boldsymbol{a}_{B,i:k}$ and $\boldsymbol{a}_{D,i:k}$, the regression matrix is likely to be rank-deficient, initially. Therefore, the algorithm starts with the $\mathcal{H}_\infty$ filter and switches to the RLS filter after three different azimuth angles are measured. The recursive solution to Equ. (8) is given as

$$\hat{\boldsymbol{x}}_{i:k}^i = \hat{\boldsymbol{x}}_{i:(k-1)}^i + \boldsymbol{P}_{i:k}\boldsymbol{A}_k^T\boldsymbol{H}_{i:k}\left(\boldsymbol{y}_k - \boldsymbol{A}_k\hat{\boldsymbol{x}}_{i:(k-1)}^i\right), \tag{15}$$

where

$$\boldsymbol{H}_{i:k} = \begin{cases} \boldsymbol{I}, & \text{for RLS} \\ (\boldsymbol{I} + \boldsymbol{A}_k\boldsymbol{P}_{i:k}\boldsymbol{A}_k^T)^{-1}, & \text{for } \mathcal{H}_\infty \end{cases}. \tag{16}$$

The covariance matrix $\boldsymbol{P}_{i:k}$ fulfils the recursion (17). By aid of the Woodbury matrix identity, the inverse of the covariance matrix reveals a remarkably simple structure, see Equ. (19). Iff $\boldsymbol{P}_{i:k}^{-1}(\gamma)$ is a positive-definite matrix, the possible worst case energy (14) is bounded by γ^2 [50]. Updating $\boldsymbol{P}_{i:k}^{-1}$ and performing an inverse of a symmetric, positive-definite matrix of size 3×3 is more efficient than updating $\boldsymbol{P}_{i:k}$ directly.

$$\boldsymbol{P}_{i:k} = \begin{cases} \boldsymbol{P}_{i:(k-1)} - \boldsymbol{P}_{i:(k-1)}\boldsymbol{A}_k^T(\boldsymbol{I}_{2\times2} + \boldsymbol{A}_k\boldsymbol{P}_{i:(k-1)}\boldsymbol{A}_k^T)^{-1}\boldsymbol{A}_k\boldsymbol{P}_{i:(k-1)}, & \text{for RLS} \\ \boldsymbol{P}_{i:(k-1)} - \boldsymbol{P}_{i:(k-1)}\begin{bmatrix}\boldsymbol{A}_k^T & \boldsymbol{A}_k^T\end{bmatrix}\boldsymbol{R}_{e,i:k}^{-1}\begin{bmatrix}\boldsymbol{A}_k \\ \boldsymbol{A}_k\end{bmatrix}\boldsymbol{P}_{i:(k-1)}, & \text{for } \mathcal{H}_\infty \end{cases}, \tag{17}$$

$$\boldsymbol{R}_{e,i:k} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\gamma\boldsymbol{I} \end{bmatrix} + \begin{bmatrix}\boldsymbol{A}_k \\ \boldsymbol{A}_k\end{bmatrix}\boldsymbol{P}_{i:(k-1)}\begin{bmatrix}\boldsymbol{A}_k^T & \boldsymbol{A}_k^T\end{bmatrix}, \tag{18}$$

$$\boldsymbol{P}_{i:k}^{-1} = \begin{cases} \boldsymbol{P}_{i:(k-1)}^{-1} + \boldsymbol{A}_k^T\boldsymbol{A}_k, & \text{for RLS} \\ \boldsymbol{P}_{i:(k-1)}^{-1} + (1 - \gamma^{-2})\boldsymbol{A}_k^T\boldsymbol{A}_k, & \text{for } \mathcal{H}_\infty \end{cases}. \tag{19}$$

5 Performance Bounds-Genie Estimators

Due to quantized angular observations, the already derived Cramér-Rao bound [25] is not applicable. We will compare the obtained estimation results to two other bounds; firstly, to the "error-free regressors model." Here, the azimuth angles ϕ_k are assumed to be perfectly known, hence unquantized, and

A Framework for Active Learning of Beam Alignment in Vehicular Millimeter Wave Communications by Onboard Sensors | *Special Topic*

Erich Zöchmann

Equ. (3) is utilized without estimates. Secondly, if the angles ϕ_k are known, Equ. (3) is not only satisfied in the least-squares sense but rather determines a solution subspace. In other words, any solution vector $\hat{\boldsymbol{x}}_{i:k}^i$ needs to fulfil

$$\hat{\boldsymbol{x}}_{i:k}^i = \left(I - A_{B,i:k}^{\dagger} A_{B,i:k} \right) \boldsymbol{m}, \ \boldsymbol{m} \in \mathbb{R}^3. \tag{20}$$

Now, Equ. (7) is solved by a nullspace projection:

$$\hat{\boldsymbol{x}}_{i:k}^{i,\text{NS}} = \text{argmin}_{\boldsymbol{m}} \| \boldsymbol{\nu}_{i:k} - A_{D,i:k} \left(I - A_{B,i:k}^{\dagger} A_{B,i:k} \right) \boldsymbol{m} \|. \tag{21}$$

As zero mean Gaussian noise is assumed for the Doppler measurements $\boldsymbol{\nu}$, the LS solution (21) is the maximum likelihood estimator of the state vector.

6 Simulations

The simulations focus on line-of-sight scenarios. Note that the proposed approach works for specular reflections as well (see next section). Clustered reflections will lead to a higher uncertainty in determining the azimuth angle. Similar to the IEEE 802.11ad standard, it is assumed that the TX is transmitting its reference signal omni-directionally. Initially, the RX is scanning all entries from the codebook and determines the direction towards the TX. We compare the performance if this procedure is repeated every 20 ms or 50 ms. After the 10th iteration, the projection (11) is used to predict the future azimuth angle. Having the projected azimuth angles at hand, the algorithm only probes the closest three codebook entries for 20 ms update rate or five codebook entries for 50 ms update rate. This gives a speed up of a factor $128/3 \approx 43$ or $128/5 \approx 26$ as compared to a full codebook scan.

The first scenario, entitled "half-overtaking", starts when the overtaking, red, TX car is at the same height as the slower, black, RX car. The overtaking car has 20 m/s excess speed and is observed for 3 s. In the second scenario, entitled "full-overtaking", the TX starts behind the RX and overtakes with an excess speed of 10 m/s. The manoeuvre is now observed for 6 s. The lateral distance was chosen such that the resulting maximum angular velocity $\omega_{\text{max}} = v/r_{\text{min}} = (20 \text{ m/s})/(8 \text{ m}) = (10 \text{ m/s})/(4 \text{ m}) = 2.5 \text{ rad/s} \approx 140^\circ/\text{s}$ is equal in both scenarios. The presented Monte Carlo mean is calculated from 10 000 runs. To obtain different channel realizations, the lateral distance is varied uniformly in $\Delta r_{\text{min}} \sim U(-1 \text{ m}, 1 \text{ m})$ around the mean lateral distances, and the angle between both cars is varied uniformly in $\varphi \sim U(-2^\circ, 2^\circ)$. These variations are drawn within the sketch of the manoeuvre as black arrows in **Fig. 5a**. The normalized mean squared error of the prior work [27], the proposed sequential implementation from Section 4, and both error bounds from Section 5 are plotted in **Figs. 5c** (20 ms update rate) and **5e** (50 ms update rate). **Figs. 5b** and **5d** show the respective scatter plots of the estimated (x,y) position of the proposed sequential estimator for the first Monte Carlo run.

Half overtaking (with update rate of 20 ms) turns out to be not so burdensome than full-overtaking. That is because right from the beginning, the TX car is seen at different azimuth angles and close to the initial solution of (0,0) and the algorithm convergences fast. The regression model of [27] suffers from a strong bias due to the error correlation of the current observation and the regression matrix. The sequential algorithm outperforms the prior non-sequential modelling approach. The "errors-in-variables" approach comes very close to the error-free regression matrix. Furthermore, there is only a small loss to the nullspace projection. Keep in mind that the "error-free regressors" and the nullspace projection approach make use of exact (yet unknown), unquantized azimuth angles! The full overtaking manoeuvre is characterized by a difficult geometry. At the beginning the TX car is always seen at the same codebook index and the algorithm struggles to converge. In this region the $\mathcal{H}_\infty$ algorithm is used to prevent divergence. After approximately 0.5 s, three different azimuth angles have been measured and the algorithm hands over to RLS. Even with an estimate of the initial state and the covariance matrix, the RLS algorithm needs a considerable time to converge afterwards. The situation is aggravated by the fact that the toughest part (TX car closest to RX car $\rightarrow \omega_{\text{max}}$) comes before convergence sets in. Nevertheless, an acceptable tracking result can be achieved here as well.

For an update rate of 50 ms the peformance loss at "full-overtaking" is minor. In contrast, the previously simpler case of "half-overtaking" has now a larger performance loss. Due to the slower update rate, after only a few measurements, the overtaking car is already at steeper angles where the Doppler shift does not change so much and convergence is harder to achieve.

7 The LOS Blocked Scenario

Until now, all cases considered LOS. The proposed tracker, however, is also applicable to scenarios with specular reflections. Fig. 4 shows a blockage scenario. The direct LOS between TX (the red car) and RX (the black car) is blocked by the green car. If another car overtakes this platoon, it can act as reflector and can be tracked by its Doppler shift. The feasibility of this approach has been verified experimentally in [9] and [23]. The Doppler shifts of overtaking vehicles produce very distinct Doppler traces. Such an exemplary trace is illustrated on the right-hand side of Fig. 4. The only adaptation for the algorithm is a factor 2 occuring in the Doppler shift equations.

8 Conclusions

Geometric tracking of specular multipath components in vehicular millimeter wave channels is possible with low complexity algorithms. The proposed algorithm achieves good tracking even under very dynamic scenarios. This considerably relaxes

Erich Zöchmann
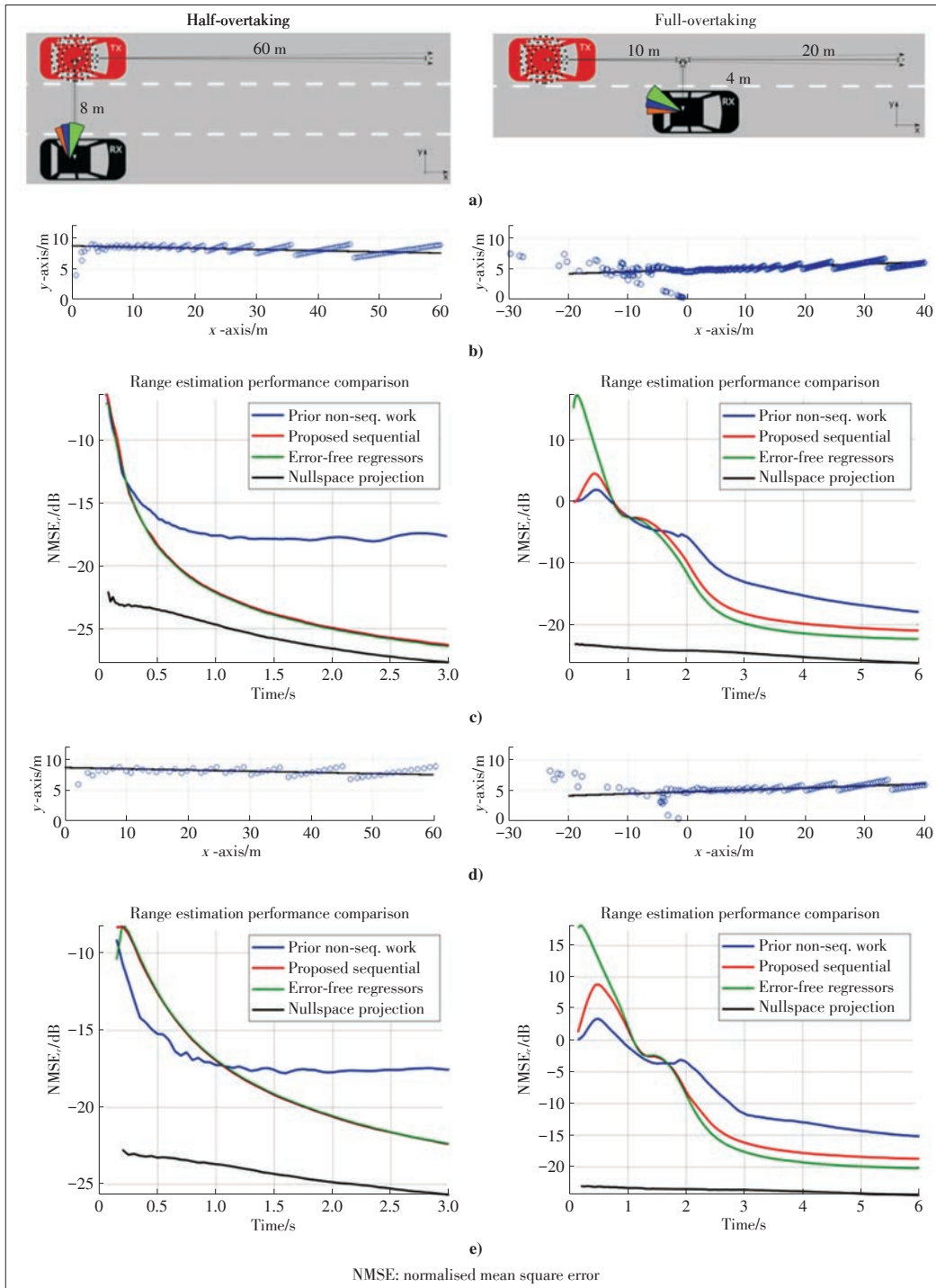
◀Figure 5. a) Manoeuvre sketch of the scenarios; b) exemplary relative position estimates (x, y) of the proposed estimator with 20 ms update rate; c) range normalized mean squared error comparisons with 20 ms update rate; d) (x, y) estimates with 50 ms update rate; e) NMSE with 50 ms update rate

the time required for beam training. In addition, the proposed algorithm outputs a state vector that reflects the relative position and velocity of the multipath components. With this knowledge, it is possible to label the targets for onboard sensors as multipath components. This enables active learning for onboard sensors.

References
[1] AKIHITO K, KATSUYOSHI S, FUJISE M, et al. Propagation Characteristics of 60‐GHz Millimeter Waves for ITS Inter‐Vehicle Communications [J]. IEICE Transactions on Communications, 2001, 84(9): 2530–2539
[2] MEINEL H, PLATTNER A. Millimeter‐Wave Propagation along Railway Lines [J]. IEE Proceedings F (Communications, Radar and Signal Processing), 1983, 130(7): 688. DOI: 10.1049/ip-f-1.1983.0102
[3] VA V, SHIMIZU T, BANSAL G, et al. Millimeter Wave Vehicular Communications: A Survey [J]. Foundations and Trends in Networking, 2016, 10(1): 1–113. DOI:10.1561/1300000054
[4] KUMARI P, CHOI J, GONZALEZ-PRELCIC N, et al. IEEE 802.11ad-Based Radar: An Approach to Joint Vehicular Communication‐Radar System [J]. IEEE Transactions on Vehicular Technology, 2018, 67(4): 3012–3027. DOI: 10.1109/

A Framework for Active Learning of Beam Alignment in Vehicular Millimeter Wave Communications by Onboard Sensors | *Special Topic*

Erich Zöchmann

tvt.2017.2774762

[5] CHOI J, VA V, GONZALEZ‑PRELCIC N, et al. Millimeter‑Wave Vehicular Communication to Support Massive Automotive Sensing [J]. IEEE Communications Magazine, 2016, 54(12): 160−167. DOI: 10.1109/mcom.2016.1600071cm

[6] LORCA J, HUNUKUMBURE M, WANG Y. On Overcoming the Impact of Doppler Spectrum in Millimeter‑Wave V2I Communications [C]//2017 IEEE Globecom Workshops. Singapore, Singapore, 2017: 1− 6. DOI: 10.1109/GLO-COMW.2017.8269039

[7] VA V, CHOI J, HEATH R W. The Impact of Beamwidth on Temporal Channel Variation in Vehicular Channels and Its Implications [J]. IEEE Transactions on Vehicular Technology, 2017, 66(6): 5014−5029. DOI: 10.1109/tvt.2016.2622164

[8] ZÖCHMANN E, HOFER M, LERCH M, et al. Statistical Evaluation of Delay and Doppler Spread in 60 GHz Vehicle-To-Vehicle Channels during Overtaking [C]//2018 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC), Cartagena des Indias, Colombia, 2018: 1− 4. DOI: 10.1109/APWC.2018.8503750

[9] ZOCHMANN E, HOFER M, LERCH M, et al. Position-Specific Statistics of 60 GHz Vehicular Channels during Overtaking [J]. IEEE Access, 2019, 7: 14216−14232. DOI:10.1109/access.2019.2893136

[10] GAO X Y, DAI L L, ZHANG Y, et al. Fast Channel Tracking for Terahertz Beamspace Massive MIMO Systems [J]. IEEE Transactions on Vehicular Technology, 2017, 66(7): 5689−5696. DOI: 10.1109/tvt.2016.2614994

[11] JAYAPRAKASAM S, MA X X, CHOI J W, et al. Robust Beam-Tracking for mmWave Mobile Communications [J]. IEEE Communications Letters, 2017, 21 (12): 2654−2657. DOI: 10.1109/lcomm.2017.2748938

[12] LI J H, SUN Y, XIAO L M, et al. Analog Beam Tracking in Linear Antenna Arrays: Convergence, Optimality, and Performance [C]//51st Asilomar Conference on Signals, Systems, and Computers. Pacific Grove, USA, 2017: 1193−1198. DOI: 10.1109/ACSSC.2017.8335540

[13] LI J H, SUN Y, XIAO L M, et al. Super Fast Beam Tracking in Phased Antenna Arrays: Theory and Performance [EB/OL]. (2017-10-22). https://arxiv.org/abs/1710.07873

[14] LI J H, SUN Y, XIAO L M, et al. How to Mobilize Mmwave: A Joint Beam and Channel Tracking Approach [C]//IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, Canada, 2018: 3624−3628. DOI:10.1109/ICASSP.2018.8461760

[15] LOCH A, ASADI A, SIM G H, et al. Mm-Wave on Wheels: Practical 60 GHz Vehicular Communication without Beam Training [C]//9th International Conference on Communication Systems and Networks (COMSNETS). Bangalore, India, 2017: 1−8. DOI:10.1109/COMSNETS.2017.7945351

[16] PALACIOS J, DE DONNO D, WIDMER J. Tracking mm-Wave Channel Dynamics: Fast Beam Training Strategies under Mobility [C]//IEEE Conference on Computer Communications (INFOCOM). Atlanta, USA, 2017: 1− 9. DOI: 10.1109/INFOCOM.2017.8056991

[17] RODRÍGUEZ-FERNÁNDEZ J, GONZÁIEZ-PRELCIC N, HEATH R W. Frequency-Domain Wideband Channel Estimation and Tracking for Hybrid MIMO Systems [C]//51st Asilomar Conference on Signals, Systems, and Computers. Pacific Grove, USA, 2017: 1829−1833. DOI: 10.1109/ACSSC.2017.8335678

[18] VA V, VIKALO H, HEATH R W. Beam Tracking for Mobile Millimeter Wave Communication Systems [C]//IEEE Global Conference on Signal and Information Processing (GlobalSIP). Washington, DC, USA, 2016: 743− 747. DOI: 10.1109/GlobalSIP.2016.7905941

[19] ZHANG C, GUO D N, FAN P Y. Tracking Angles of Departure and Arrival in a Mobile Millimeter Wave Channel [C]//IEEE International Conference on Communications (ICC). Kuala Lumpur, Malaysia, 2016: 1− 6. DOI:10.1109/ICC.2016.7510902

[20] ZHAO J W, GAO F F, JIA W M, et al. Angle Domain Hybrid Precoding and Channel Tracking for Millimeter Wave Massive MIMO Systems [J]. IEEE Transactions on Wireless Communications, 2017, 16(10): 6868− 6880. DOI: 10.1109/twc.2017.2732405

[21] ZHOU Y F, YIP P C, LEUNG H. Tracking the Direction-Of-Arrival of Multiple Moving Targets by Passive Arrays: Algorithm [J]. IEEE Transactions on Signal Processing, 1999, 47(10): 2655−2666. DOI: 10.1109/78.790648

[22] ZÖCHMANN E, CABAN S, LERCH M, et al. Resolving the Angular Profile of 60 GHz Wireless Channels by Delay-Doppler Measurements [C]//IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM). Rio de Janerio, Brazil, 2016: 1−5. DOI:10.1109/SAM.2016.7569652

[23] ZÖCHMANN E, MECKLENBRÄUKER C F, LERCH M, et al. Measured Delay and Doppler Profiles of Overtaking Vehicles at 60 GHz [C]//12th European Conference on Antennas and Propagation (EuCAP). London, UK: 1−5, 2018. DOI: 10.1049/cp.2018.0470

[24] ZENG Y, ZHANG R. Millimeter Wave MIMO with Lens Antenna Array: A New Path Division Multiplexing Paradigm [J]. IEEE Transactions on Communications, 2016, 64(4): 1557−1571. DOI: 10.1109/tcomm.2016.2533490

[25] CHAN Y T, RUDNICKI S W. Bearings-Only and Doppler-Bearing Tracking Using Instrumental Variables [J]. IEEE Transactions on Aerospace and Electronic Systems, 1992, 28(4): 1076−1083. DOI: 10.1109/7.165369

[26] HO K C, CHAN Y T. An Asymptotically Unbiased Estimator for Bearings-Only and Doppler-Bearing Target Motion Analysis [J]. IEEE Transactions on Signal Processing, 2006, 54(3): 809−822. DOI: 10.1109/tsp.2005.861776

[27] HO K C, CHAN Y T. Geometric-Polar Tracking from Bearings-Only and Doppler-Bearing Measurements [J]. IEEE Transactions on Signal Processing, 2008, 56(11): 5540−5554. DOI: 10.1109/tsp.2008.928701

[28] RAO S K. Pseudo-Linear Estimator for Bearings-Only Passive Target Tracking [J]. IEE Proceedings—Radar, Sonar and Navigation, 2001, 148(1): 16. DOI: 10.1049/ip-rsn:20010144

[29] ALIEIEV R, HEHN T, KWOCZEK A, et al. Predictive Communication and Its Application to Vehicular Environments: Doppler-Shift Compensation [J]. IEEE Transactions on Vehicular Technology, 2018, 67(8): 7380−7393. DOI: 10.1109/tvt.2018.2835662

[30] GONZALEZ-PRELCIC N, ALI A, VA V, et al. Millimeter-Wave Communication with Out-Of-Band Information [J]. IEEE Communications Magazine, 2017, 55(12): 140−146. DOI: 10.1109/mcom.2017.1700207

[31] DANIELS R C, CARAMANIS C M, HEATH R W. Adaptation in Convolutionally Coded MIMO - OFDM Wireless Systems through Supervised Learning and SNR Ordering [J]. IEEE Transactions on Vehicular Technology, 2010, 59(1): 114−126. DOI: 10.1109/tvt.2009.2029693

[32] DJOUAMA A, ZÖCHMANN E, PRATSCHNER S, et al. Predicting CSI for Link Adaptation Employing Support Vector Regression for Channel Extrapolation [C]//20th International ITG Workshop on Smart Antennas (WSA 2016). Munich, Germany, 2016: 1−7

[33] VA V, CHOI J, SHIMIZU T, et al. Inverse Multipath Fingerprinting for Millimeter Wave V2I Beam Alignment [J]. IEEE Transactions on Vehicular Technology, 2018, 67(5): 4042−4058. DOI: 10.1109/tvt.2017.2787627

[34] ZÖCHMANN E, VA V, RUPP M, et al. Geometric Tracking of Vehicular mmWave Channels to Enable Machine Learning of Onboard Sensors [C]//2018 IEEE Globecom Workshops. Abu Dhabi, United Arab Emirates, 2018: 1− 6. DOI: 10.1109/GLOCOMW.2018.8644440

[35] TAHAT A, KADDOUM G, YOUSEFI S, et al. A Look at the Recent Wireless Positioning Techniques with a Focus on Algorithms for Moving Receivers [J]. IEEE Access, 2016, 4: 6652−6680. DOI: 10.1109/access.2016.2606486

[36] SHAHMANSOORI A, GARCIA G E, DESTINO G, et al. Position and Orientation Estimation through Millimeter - Wave MIMO in 5G Systems [J]. IEEE Transactions on Wireless Communications, 2018, 17(3): 1822− 1835. DOI: 10.1109/twc.2017.2785788

[37] TRULLENQUE ORTIZ M, et al. Vehicle Tracking Through Vision-Millimeter Wave Doppler Shift Fusion [Z]. to be published at IEEE-APS APWC, 2019.

[38] HOANG M G, DENIS B, HÄRRI J, et al. (2015). Distributed Link Selection and Data Fusion for Cooperative Positioning in GPS - Aided IEEE 802.11p VANETs [C]//12th Workshop on Positioning, Navigation and Communication. Dresden, Germany, 2015

[39] COHN D, GHAHRAMANI Z, JORDAN M J. Active Learning with Statistical Models [J]. Journal of Artificial Intelligence Research, 1996, 4: 129−145. DOI: 10.1613/jair.295

[40] COHN D, ATLAS L, LADNER R. Improving Generalization with Active Learning [J]. Machine Learning, 1994, 15(2): 201−221. DOI: 10.1007/bf00993277

[41] SETTLES B. Active Learning Literature Survey [R/OL]. (2010-01-26). http://burrsettles.com/pub/settles.activelearning.pdf

[42] ROCKL M, STRANG T, KRANZ M. V2V Communications in Automotive Multi-Sensor Multi - Target Tracking [C]//IEEE 68th Vehicular Technology Conference. Calgary, Canada, 2008: 1−5. DOI: 10.1109/VETECF.2008.440

[43] GRZYWACZEWSKI A. NVIDIA Corporation: Training AI for Self-Driving Vehicles: the Challenge of Scale [R/OL]. (2017). https://devblogs.nvidia.com/parallelforall/training-self-driving-vehicles-challenge-scale

[44] LIU S S, TANG J, ZHANG Z, et al. Computer Architectures for Autonomous Driving [J]. Computer, 2017, 50(8): 18−25. DOI: 10.1109/mc.2017.3001256

[45] HUANG C, LU R X, CHOO K−K R. Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges [J]. IEEE Communications Magazine, 2017, 55(11): 105−111. DOI: 10.1109/mcom.2017.1700322

[46] KONG L H, KHAN M K, WU F, et al. Millimeter-Wave Wireless Communications for IoT-Cloud Supported Autonomous Vehicles: Overview, Design, and Challenges [J]. IEEE Communications Magazine, 2017, 55(1): 62−68. DOI:

Novel Real-Time System for Traffic Flow Classification and Prediction

YE Dezhong[1], LV Haibing[1], GAO Yun[2], BAO Qiuxia[2], and CHEN Mingzi[2]

(1. ZTE Corporation, Shenzhen, Guangdong 518057, China;
2. Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China)

Abstract: Traffic flow prediction has been applied into many wireless communication applications (e.g., smart city, Internet of Things). With the development of wireless communication technologies and artificial intelligence, how to design a system for real-time traffic flow prediction and receive high accuracy of prediction are urgent problems for both researchers and equipment suppliers. This paper presents a novel real-time system for traffic flow prediction. Different from the single algorithm for traffic flow prediction, our novel system firstly utilizes dynamic time wrapping to judge whether traffic flow data has regularity, realizing traffic flow data classification. After traffic flow data classification, we respectively make use of XGBoost and wavelet transform-echo state network to predict traffic flow data according to their regularity. Moreover, in order to realize real-time classification and prediction, we apply Spark/Hadoop computing platform to process large amounts of traffic data. Numerical results show that the proposed novel system has better performance and higher accuracy than other schemes.

Keywords: traffic flow prediction; dynamic time warping; XGBoost; echo state network; Spark/Hadoop computing platform

1 Introduction

In recent years, as the advances of multimedia communications and technologies [1]–[3], traffic flow data has been widely studied and used in various situations. For one thing, mobile users pay more attention on their experience of surfing online; for another, some service providers like Zhongxing Telecommunication Equipment Corporation (ZTE) want to provide users with better service by traffic flow prediction and allocation. Therefore, traffic flow research has become one of the hottest research topics in our daily lives. As we all know, quality of experience (QoE) has been considered as one of the most evaluation criteria for user experience description, which is mixed with subjective and objective indicators [4]. To improve users' QoE, service providers will predict traffic flow data so that they could substitute their devices to provide users with higher satisfaction of service. Obviously, how to accurately predict traffic flow data and quickly estimate traffic flow trends are very important for both users and service providers.

Generally speaking, traffic flow prediction greatly depends on historical records from different base stations. With the increasing number of users, the amount of traffic flow data becomes larger than before, which may bring different types of traffic trends. For example, **Fig. 1** shows three different traffic trends among the dataset. Given these flow diagrams, we have the following observations. Figs. 1a and 1b show the strong regularity and characteristics in the usage of traffic flow data for different regions like commercial district and residential area. For example, Fig. 1a shows that the traffic flow data starts to
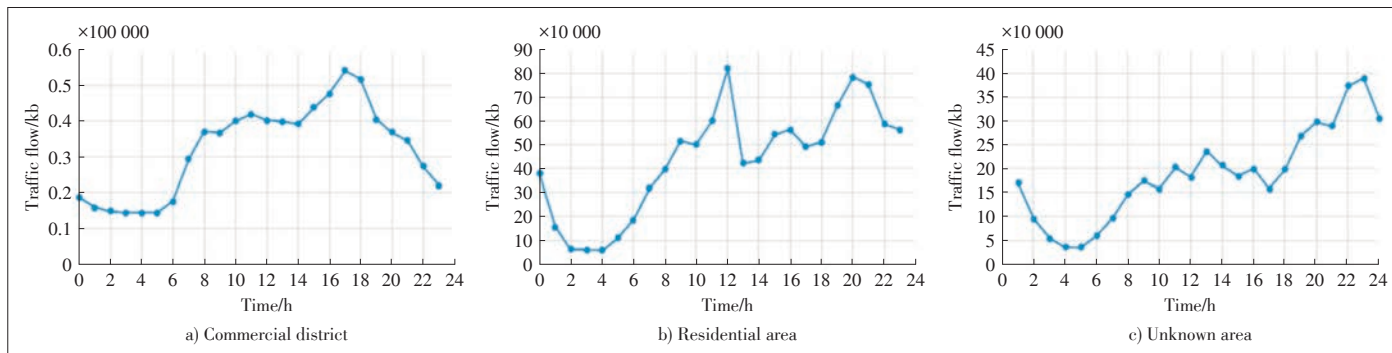
▲Figure 1. Different traffic trends among the dataset.

rise at 8 a.m. and decrease at about 6 p.m. Fig. 1b is the opposite of Fig. 1a. Compared with the two flow diagrams mentioned before, Fig. 1c seems to have weak regularity. It may contain the combinations of many non-linear factors such as base station's positions and different time periods [5], [6]. Besides, there are some other types of traffic flow regularity in the dataset. Therefore, how to predict these different types of traffic flow data has become a urgent problem for researchers.

Moreover, based on the observations mentioned before, how to define the regularity for traffic flow data is also very important. As we all know, most of us only use subjective feelings to classify the regularity for traffic flow data. However, faced with increasing number of traffic flow data, it would be impossible for us to judge regularity manually. Essentially, we must find the objective method to classify these traffic flow data.

Faced with issues mentioned before, we design a novel real-time system for traffic flow classification and prediction. It is an integrated system concerning both traffic flow classification and prediction. This system is divided into two main models: a traffic flow classification model and a traffic flow prediction model. In particular, for the classification model, we propose the dynamic time warping (DTW) algorithm for traffic classification, and it could classify traffic data into two categories including regularity and non-regularity. Subsequently, for the prediction model, we adopt XGBoost algorithm to predict traffic flow data with regularity. For traffic flow data with non-regularity, we first use wavelet transform (WT) to decompose them into different coefficients. Then we make full use of echo state network (ESN) to predict each coefficient. Specially, to meet the real-time requirement for the whole system, we also do data preprocessing and feature selection by Spark/Hadoop computing platform. In summary, the main contributions of this work could be summarized as follows:

(1) We propose the DTW algorithm to classify traffic flow data, which is beneficial to improve the accuracy of traffic flow prediction.

(2) We use the combination of WT and ESN to predict the traffic flow data without regularity. For the regular data, we adopt XGBoost algorithm to do prediction. In other words, for different categories of data, we select different adapt-

able algorithms to do the traffic flow prediction.

(3) We design the novel real-time system for traffic flow prediction, and it will favor us in handling traffic flow data automatically and quickly.

The rest of this paper is organized as follows. In Section 2, we discuss the related work in this area. In Section 3, system overview is described in detail. In Sections 4 and 5, we respectively introduce the traffic flow classification and prediction models. In Section 6 we give our experimental results. Finally, we give the conclusion in Section 7.

2 Related Work

There are some existing works that focus on the study of traffic flow data prediction and classification. Generally speaking, existing works usually do not consider different types of traffic trends or diagrams, and often use one single algorithm or its extensions to do the prediction. Actually, how to design a suitable system for different types of traffic data prediction is still an open problem for network service providers.

For one thing, the main solutions of traffic flow classification are divided into two categories, which are machine learning classification algorithms and time-series statistical similarity measurement. Williams et al. [7] introduce supervised algorithms which are Bayesian Network, C4.5 Decision Tree, and Naïve Bayes for traffic flow classification. Experimental results show that these supervised learning algorithms have better performances in the area of classification compared with other competing models. Moreover, Hu et al. [8] and Hochst et al. [9] both make use of the clustering algorithm which is the typical type of unsupervised learning algorithms for classification. In addition, some scholars concentrate on statistical learning to measure the similarity between different traffic diagrams. For example, Jeong et al. [10] propose weighted dynamic time warping (WDTW) for time series classification, which is a novel distance measurement for traffic data. Compared with machine learning algorithms for classification, statistical learning methods like DTW pay more attention on the similarity measurement in the shape of traffic diagrams.

For another, owing to the strong generality of boosting and

decision tree, Chen et al. [11] propose the XGBoost algorithm to improve the accuracy of classification and regression. Moreover, with the huge development of neural networks, Zhu et al. [12] propose a novel BP neural network model for traffic prediction, which brings better learning ability than some common machine learning algorithms. However, faced with complex nonlinear factors to traffic data, these algorithms would not perform better than other competing models for all of traffic diagrams. Therefore, Yang et al. [13] make full use of WT method to decompose the traffic flow data into different coefficients, and use radial basis function network for traffic prediction.

In summary, most of existing works do not concern about the classification of traffic data, but only use some techniques to do the optimization. Moreover, there is less study about the real‑time system for traffic flow prediction and management. Based on these analysis, we design a novel real‑time for both traffic classification and prediction.

3 System Overview

3.1 System Brief Description

For the purpose of the whole system for traffic flow data treatment and management, we fully consider the complete process for traffic flow data, including data preprocessing, traffic flow classification and traffic flow prediction. The flow diagram of framework for system is shown in **Fig. 2**.

Briefly speaking, we firstly do data preprocessing containing data cleaning and feature engineering by Spark/Hadoop computing platform, which will greatly improve the speed of data processing. Then, the processed data would be sent to traffic

▲Figure 2. Algorithm flow chart of the proposed novel real‑time system.

flow classification model. The realization of this model is based on DTW algorithm. After performing by DTW, if the ratings of traffic flow data are above the threshold value, they are taken as data with regularity. And they will be directly transferred to do the prediction by Xgboost algorithm. Otherwise, they are classified as data with non‑regularity. They are decomposed into different coefficients by wavelet transform and then do the traffic prediction by ESN algorithm. Finally, we could get the promising results through this novel system even if the traffic flow data has different categories.

3.2 Data Collection and Preprocessing

For our study, the dataset we used comes from ZTE, which is one of the largest service providers in China. In particular, the traffic flow data in the dataset is from 5 762 different base stations. The specific meanings of the indicators are introduced in **Table 1**. Moreover, the collecting time of the data ranges from May 1, 2015 to June 20, 2017, and the order of magnitude for traffic flow data reaches GB.

Essentially, efficient and real‑time system depends on not only the optimization of learning algorithms, but also the high speed of data processing. Owing to redundant and duplicate values in the dataset, we must do data preprocessing before traffic flow data classification and prediction. To satisfy the efficient requirement, we make full use of Spark/Hadoop computing platform to do the data preprocessing, including data cleaning and feature engineering. Specifically, Spark/Hadoop computing platform [14], [15], on behalf of the distributed computing, is a unified analytics engine for large‑scale data processing. For so much missing data and outliers in the dataset, we clean these data by Spark/Hadoop. In addition, we process and perform the analysis of features which may influence the regularity of the traffic flow data.

As we all know, the important element of the traffic flow analysis is the inspection of the traffic flow fluctuations, related to the following factors:

- Time: time of day, day of week, month of year, etc.
- Base station position: school, high‑speed railway station, business zone, etc.
- Traffic type: long term evolution (LTE), optical line terminal (OLT), etc.

Therefore, we observe that these following specific features are vital to the traffic flow prediction.

(1) The concrete date of the traffic data. For example, 2016‑05

▼Table 1. Data attributes

Indicators	Meanings
Noid	ID of the base station
Name	Location information of the base station
Time	Collecting time of the data
kb	Traffic value of the base station by one day
Area	Administrative region of the base station
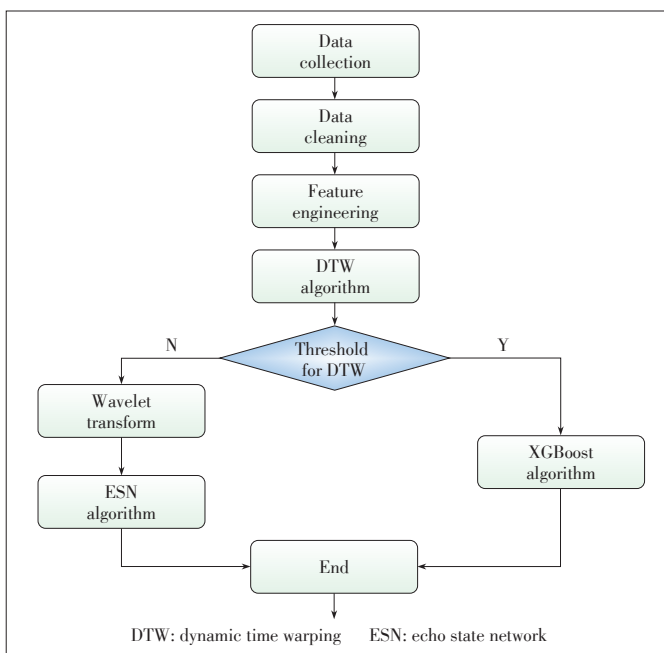
-10, etc.

(2) Whether the day is the working day plays an important role in prediction. For example, 2016-05-10 is the working day and we will mark this day 0 to represent working day. Otherwise, we could label 1.

(3) The day before a week is also one factor to do the prediction.

(4) The day before one year. For example, if we want to predict traffic flow data at 2016-05-10, the traffic data at 2015-05-10 will be this new feature.

(5) The sliding window of the current day derives some new features. From our analysis, we select current day before 1 day, 3 days and 5 days as new features. For instance, we will select 2016-05-09, 2016-05-07 and 2016-05-05 as new features to predict traffic flow data at 2016-05-10.

Theoretically, we should have selected base station's positions as a feature. To our disappointment, the base stations' positions are all different codes, and it is hard for us to translate the positions to real place in the map. Therefore, we do not deal with these values in our feature engineering and experiments.

4 Traffic Flow Classification Model

4.1 Introduction of DTW Algorithm

Dynamic time warping (DTW) is the algorithm to compare the similarity between two time series, which finds the minimum path by providing non-linear alignments. It has been applied into many fields, such as speech processing, time series classification, and clustering.

Briefly speaking, the methodology for DTW is as follows. For example, **Fig. 3** shows the optimal warping path between two different time series. We assume that one time series A has the length of p. We also assume another time series B with the length q, which is a different length from the time series A. Subsequently, we create the p-by-q path matrix where the element in the matrix represents the distance between two points. More importantly, there are also several constraints in building warping path [16]. The best match between two sequences is the one path with the lowest distance. Specifically, the optimal
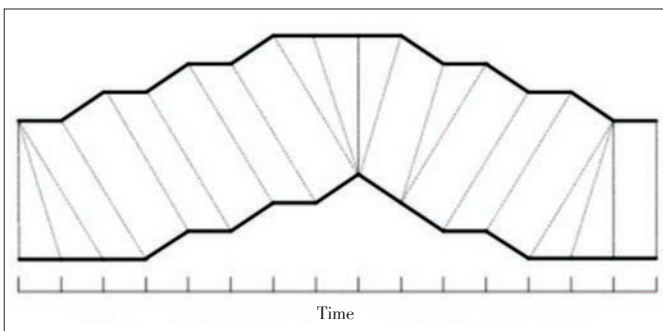
▲Figure 3. Optimal warping path between two different time series.

warping path could be found as the following equations:

$$DTW_p(A,B) = \sqrt[p]{\gamma(i,j)}, \tag{1}$$

where $\gamma(i,j)$ is the cumulative distance and the index p means the l_p norm in the equation. Generally speaking, most of us use Euclidean distance to calculate distance, where p equals two. Specifically, $\gamma(i,j)$ is described by

$$\gamma(i,j) = \left|a_i - b_j\right|^p + \min\{\gamma(i-1,j-1),\gamma(i-1,j),\gamma(i,j-1)\}, \tag{2}$$

where a_i and b_j are from two sequences with different lengths.

In summary, the DTW algorithm could accurately measure the similarity between time series even if two time series have different lengths.

4.2 Ratings of Different Traffic Data

In our system, we will select some typical traffic flow diagrams with obvious regularity as the templates, and these traffic flow data could be estimated and predicted easily. The DTW algorithm could measure the similarity between the other traffic flow data and the templets. Finally, we could get the ratings from the DTW algorithm. The distributions of the ratings are shown in **Table 2**.

We are surprised to find from the traffic flow diagrams that the lower rating the traffic flow diagram has, the more obvious regularity it has. On the basis of our analysis for the distributions of the ratings, we set the ratings to 5, which is the threshold for the regularity of traffic flow data. In other words, if the ratings of the traffic flow data beyond 5, we consider it regular in the dataset. **Fig. 4** shows the specific traffic flow diagram.

5 Integrated Traffic Flow Prediction Model

In this section, we mainly introduce the integrated traffic flow prediction model. For the classified traffic flow data, we adopt different algorithms for prediction, which are the XGBoost and WT-ESN. Therefore, this section is divided into two parts. One is for the traffic data with obvious regularity, and the other is for the other traffic data.

5.1 Prediction for Traffic Data with Strong Regularity

Figs. 1a and 1b display strong regularity of traffic flow data. For the business zone, the traffic flow reaches at the peak val-

▼Table 2. Distributions of ratings for dynamic time warping (DTW)

Ratings	Percentage
0–5	57%
5–10	15%
10–15	12%
more than 15	11%
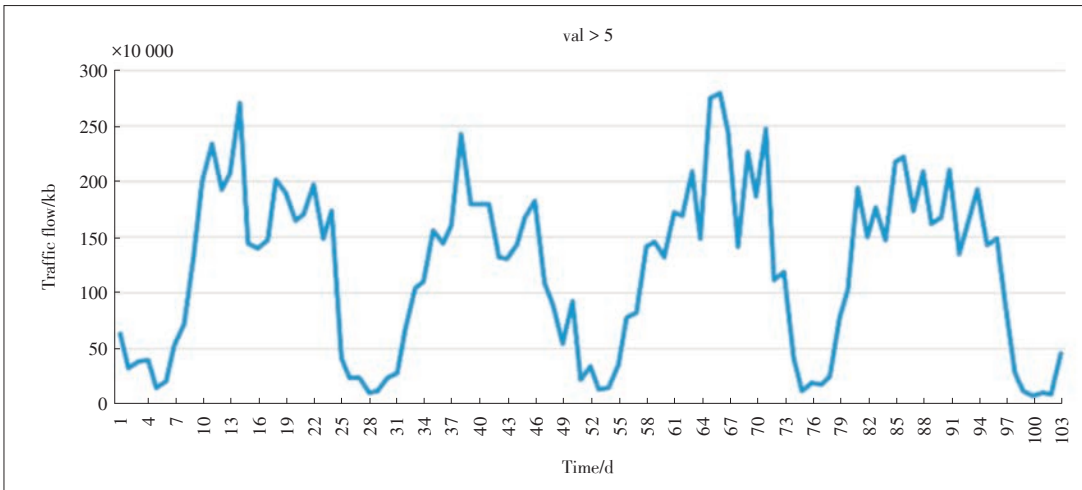
◀**Figure 4.**
Specific traffic flow for dynamic time warping (DTW).

ue at around 9 a.m−11 a.m. Accordingly, it decreases to the valley value at 1 a.m−4 a.m. Theoretically, machine learning algorithms could learn the knowledge and regularity from different datasets like the human, which is the main reason for its popularity. XGBoost algorithm [11], as the emerging algorithm in the supervised learning, has better learning ability compared with others supervised learning algorithms. In fact, XGBoost combines the ensemble method with the optimization of gradient descent. The following equations mainly introduce the core of this algorithm:

● Objective

$$Obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right). \tag{3}$$

● Objective, with constants removed

$$\sum_{i=1}^{n}\left[g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right)\right] + \Omega\left(f_t\right), \tag{4}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right), h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$. Moreover, in Eqs. (3) and (4), y_i means the true value and $\hat{y}_i$ is the predicted value. l represents the loss function and Ω is the penalty term. f is the regression tree. Compared with traditional optimization methods like Decision Tree, Gradient Boosting Decision Tree (GBDT) [17], XGBoost uses the second order Taylor expansion to approximate the objective method, which could optimize it quickly in the general setting.

5.2 Prediction for Traffic Data with Non-Regularity

As mentioned before, we consider that the traffic flow data has regularity when the rating of traffic flow data beyond 5. Faced with these data, we believe that the combinations of nonlinear factors result in this phenomenon. Therefore, we adopt WT to decompose the traffic flow data into different coefficients, which could predict it separately. Moreover, with the better generality and fast computing in neural networks, we choose a kind of recurrent neural network called ESN to do

each prediction.

ESN is a kind of dynamic neural network, which mainly includes input layer, middle layer and output layer. The middle layer is also called dynamic reservoir and it contains a large amount of sparse connected neurons, which plays the important role in short‑term memory achievement and prediction. The structure of network is shown in the **Fig. 5**. Specifically, the memory and prediction function depend on two equations called the state update equation and the output state equation.

$$x(t+1) = f(W_{in} \cdot u(t+1) + W \cdot x(t)), \tag{5}$$

$$\hat{y}(t+1) = f^{out} \cdot (W_{out} \cdot (u(t+1), x(t+1), y(t))), \tag{6}$$

where $x(t)$ is the state of reservoir at time t and $u(t)$ is the state of input. f is the neuron activation function. W_{in} and W are the input and middle layer weight matrix. $y(t)$ is the output at time t and $\hat{y}(t+1)$ is the predicted output at time $t+1$. Essentially, ESN could avoid the shortcomings of typical gradient‑descent‑based RNN algorithm and have the simple learning procedure, which is the main reason for its applications in the
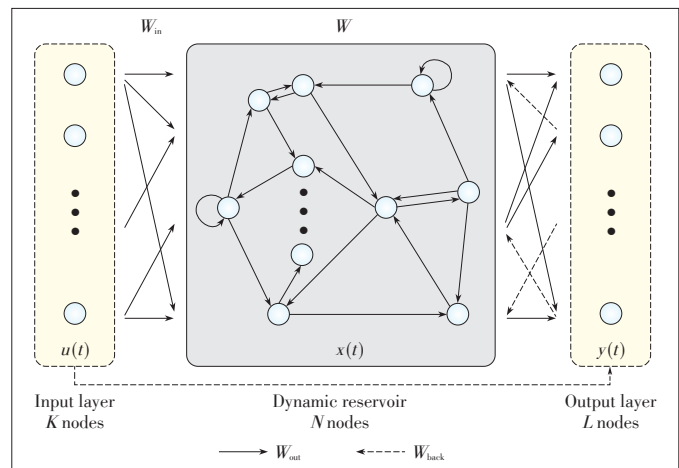
▲**Figure 5. Network structure of echo state network (ESN).**

field of time series prediction. However, faced with traffic data without regularity, ESN could not accurately predict each type of traffic data well for its different regularities.

To solve these issues, we make efforts to adopt wavelet transform (WT) to decompose the initial traffic flow data into different coefficients, which could reduce the complexity of prediction. Briefly speaking, WT is a method of transform analysis and its main characteristic is to analyze the localization of frequency. For example, **Fig. 6** shows the details of WT, where the blue one represents original signal and yellow ones represent different degrees of details. Specifically, C3 is the lowest frequency factor among all coefficients, and it has the familiar trend with original signal. D1, D2 and D3 are all the high-frequency parts of the original signal and we use ESN algorithm to predict them. For non-regularity traffic flow data, we utilize

Haar function as the core of the WT. The concrete steps of non-regularity traffic flow data prediction are described as follows:

Step 1: WT decompose the initial non-regularity data y_i into four coefficients called C3, D1, D2 and D3, which are the inputs of the ESN.

Step 2: Through single reconstruction, four coefficients including C3, D1, D2 and D3 are respectively sent to ESN for network parameters W adjustment. And it receive the predicted value for each single line.

Step 3: After training ESN, four predicted values are aggregated as the final output, which is also the prediction of the non-regularity traffic flow data $\hat{y}_i$.

6 Results and Analysis

In this section, we will conduct our experiments on the dataset collected from ZTE. We will first introduce two evaluation indexes in detail and then show and analyze the experimental results.

6.1 Evaluation Indexes Introduction

Here we briefly introduce two evaluation indexes called Normalized Root Mean Square Error (NRMSE) and R-Squared (R^2), which are the most common evaluation indexes in traffic flow prediction. Specifically, the definitions of NRMSE and R^2 are shown as follows:

$$NRMSE = \sqrt{\frac{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}{n}} / (y_{max} - y_{min}), \qquad (7)$$

$$R^2 = 1 - \sum (\hat{y}_i - \bar{y})^2 / \sum (y_i - \bar{y})^2, \qquad (8)$$

where $\hat{y}$ means the predicted value and y_t is the true value. On one hand, lower values of NRMSE represent better accuracy of the model. On the other hand, R^2 indicates the degree of fit for traffic flow data, and thus higher values of R^2 represent better degree of fit. Compared with NRMSE, R^2 focuses on the similarity of traffic flow data. In our experiments, we put more emphasis on NRMSE, because accuracy is our first choice.

6.2 Analysis of Results

As mentioned before, we select the dataset that ranges from May 1, 2015 to June 20, 2017 as a training set. In order to show the accuracy and degree of fit in detail, we choose one week's data from June 21, 2017 to June 27, 2017 as testing set. To prove the better performances of XGBoost for regular traffic flow data, we compare the NRMSE and the degree of fit with some supervised learning algorithms and other commonly used algorithms. From the results shown in **Table 3** and **Fig. 7**, we could conclude that XGboost has the best performances among all algorithms. Specifically, the prediction of XGBoost fits the real data very well. From the perspective of evaluation
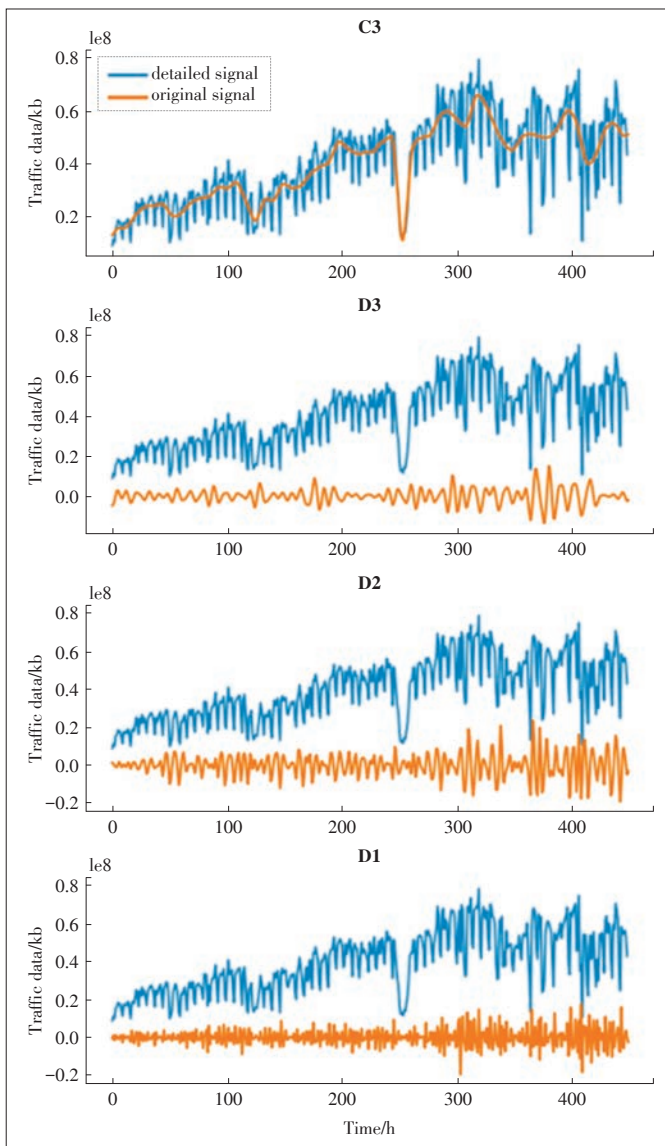
▲Figure 6. Three-layer wavelet decomposition and single reconstruction.

▼Table 3. Results of NRMSE for six algorithms

Algorithms	NRMSE
XGBoost	0.012
Linear regression	0.089
ARIMA	0.017
MLP	0.050
Wavelet transform	0.157
LSTM	0.062

ARIMA: autoregressive integrated moving average   NRMSE: normalized root mean square error
LSTM: long short-term memory   MLP: multi-layer perception

ARIMA: autoregressive integrated moving average   MLP: multi-layer perception
LSTM: long short-term memory   WT: wavelet transform
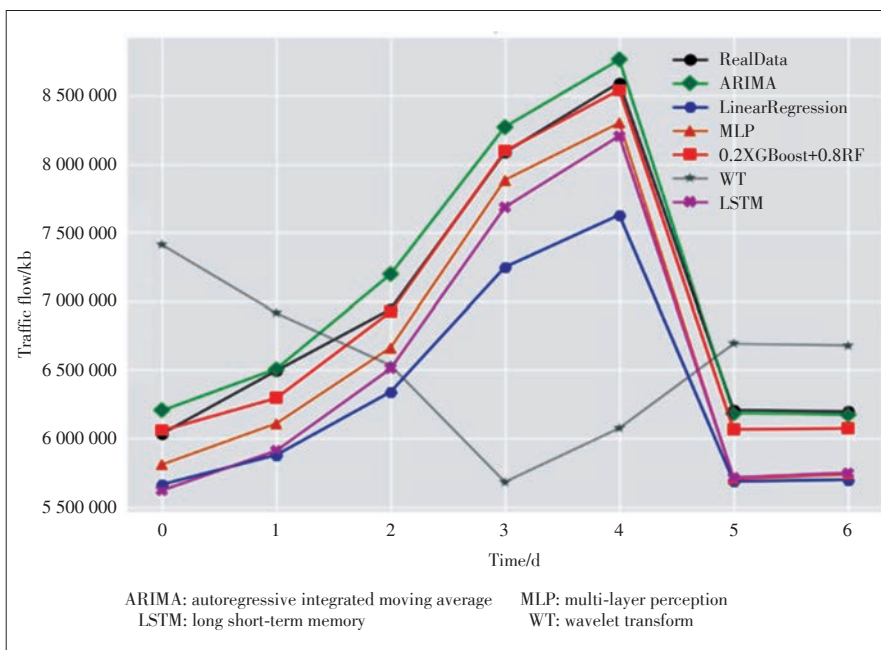
▲Figure 7. Degrees of fit for different algorithms.

index, XGboost has the lowest value of normalized root mean square error (NRMSE) among all algorithms, which means high accuracy of prediction.

Moreover, we compare performances of our integrated model with traditional algorithms for traffic flow data prediction, including autoregressive moving average model (ARMA) time series analysis, LSTM and WT algorithm. **Tables 4** and **5** show

▼Table 4. Results of Normalized Root Mean Square Error (NRMSE) in the Experiments

MAE	Integrated model	ARMA	WT	LSTM
≤15%	60.12%	22.9%	51.3%	56.6%
≤25%	81.14%	35.4%	70.5%	75.1%
≤30%	86.06%	52.6%	77.4%	82.5%
≤50%	93.23%	74.6%	86.5%	90.0%
≤1	96.98%	95.0%	95.3%	96.0%

ARMA: autoregressive moving average model   MAE: mean absolute error
LSTM: long short-term memory   WT: wavelet transform

▼Table 5. Results of R^2 in the Experiments

R^2	Integrated model	ARMA	WT	LSTM
≥0.8	8.5%	6.11%	7.50%	7.46%
≥0.5	19.46%	15.2%	18.4%	17.39%
≥0	59.32%	41.0%	59.0%	59.1%

ARMA: autoregressive moving average model   WT: wavelet transform
LSTM: long short-term memory

the results of two evaluation indexes mentioned before for our integrated model and four extra algorithms. From these tables, we could conclude that our integrated model has the lowest values of NRMSE that correspondingly represents highest accuracy among all algorithms. In addition, another evaluation index R^2 for traffic flow prediction pays more attention on the degree of fit for them. As shown in the tables, our integrated model has better degree of fit than others, which means the shape of traffic flow prediction is more similar with the true data. Generally speaking, XGBoost and WT-ESN have their own advantages but it is impossible for them to accurately predict all types of traffic flow data. In this case, our integrated model makes full use of advantages of these two algorithms, obtaining the best prediction performance.

On the theoretical aspect, we analyze the feature importance and contributions to the predicted value among all features, which are also the interesting aspects for service providers. Among all features referred in Section 3.2, we extract top-5 features influencing the predicted value, as shown in **Fig. 8**. In particular, F-score in this figure measures the importance of the features, and higher value represents more effective feature. The concrete meanings of these features are presented in **Table 6**. This could help us know what features have deep impact on the predicted value. Finally, **Fig. 9** shows the overview of our system. There are mainly four parts in our system, including raw data overview, data statistical analysis, data classification, and data prediction. When the raw data is processed in the background, relevant information will be displayed in this system.

7 Conclusions

Traffic flow prediction has been a hot research topic in recent years. However, how to systematically realize real-time traffic prediction in the context of big data era is urgent to be solved. In this paper, we propose a system for traffic flow prediction and classification. Compared with traditional algo-
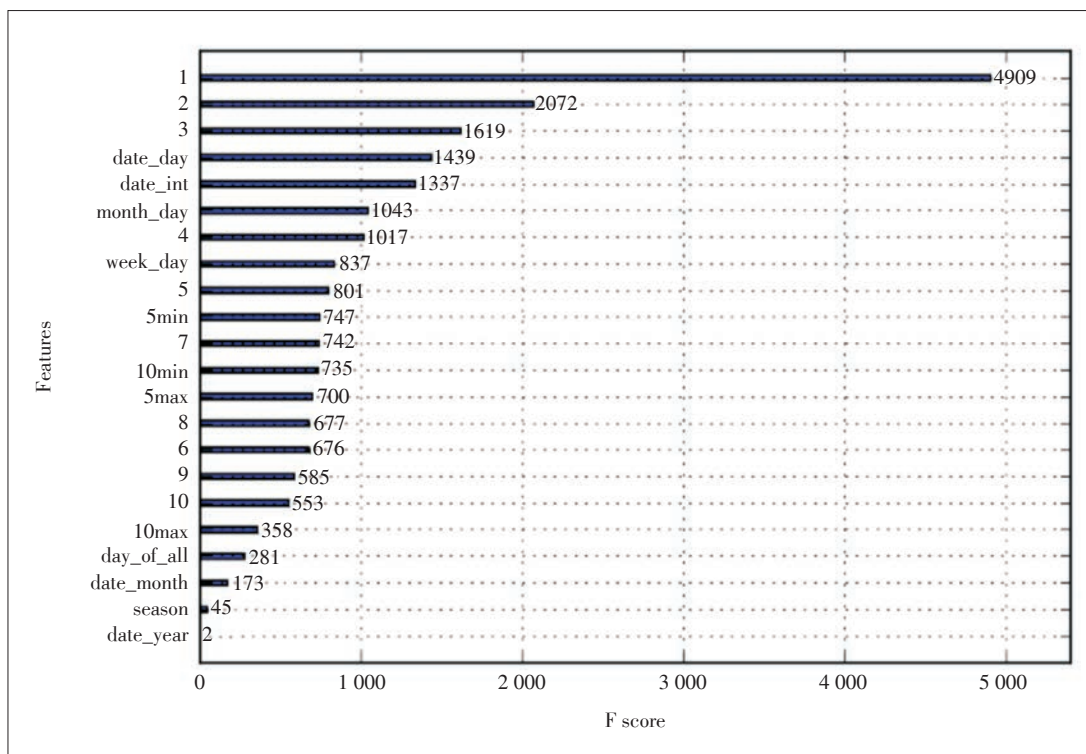
◀Figure 8.
Feature importance rankings in the experiment.

▼**Table 6. Feature selection in the experiment**

Features	Meanings
1	Average values one week before
2	Average values two weeks before
3	Average values three weeks before
Week_day	Today is a weekend or working day
Date_int	The concrete date of today

rithms or models for traffic flow prediction, our system firstly utilizes DTW algorithm for classification, which is a method to solve different trends of traffic flow prediction. Then, we mainly make full use of XGBoost and ESN to predict traffic flow data. Experiment results show that our system has better performances than other traditional algorithms.

References
[1] ZHOU L, WU D, DONG Z J, et al. When Collaboration Hugs Intelligence: Con-

◀Figure 9.
Overview of the system.

tent Delivery over Ultra-Dense Networks [J]. IEEE Communications Magazine, 2017, 55(12): 91–95. DOI: 10.1109/mcom.2017.1700481

[2] ZHOU L, WU D, CHEN J X, et al. Greening the Smart Cities: Energy-Efficient Massive Content Delivery via D2D Communications [J]. IEEE Transactions on Industrial Informatics, 2018, 14(4): 1626–1634. DOI: 10.1109/tii.2017.2784100

[3] WU D, WANG J L, HU R Q, et al. Energy-Efficient Resource Sharing for Mobile Device-To-Device Multimedia Communications [J]. IEEE Transactions on Vehicular Technology, 2014, 63(5): 2093–2103. DOI: 10.1109/tvt.2014.2311580

[4] ZHAO T S, LIU Q, CHEN C W. QoE in Video Transmission: A User Experience-Driven Strategy [J]. IEEE Communications Surveys & Tutorials, 2017, 19(1): 285–302. DOI: 10.1109/comst.2016.2619982

[5] PAMULA T. Classification and Prediction of Traffic Flow Based on Real Data Using Neural Networks [J]. Archives of Transport, 2012, 24(4): 519–529. DOI: 10.2478/v10174-012-0032-2

[6] GUO F C, POLAK J W, KRISHNAN R. Comparison of Modelling Approaches for Short Term Traffic Prediction under Normal and Abnormal Conditions [C]// 13th International IEEE Conference on Intelligent Transportation Systems. Funchal, Portugal, 2010: 1209–1214. DOI: 10.1109/ITSC.2010.5625291

[7] WILLIAMS N, ZANDER S, ARMITAGE G. A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification [J]. ACM SIGCOMM Computer Communication Review, 2006, 36(5): 5. DOI: 10.1145/1163593.1163596

[8] HU C C, YAN X H. Mining Traffic Flow Data Based on Fuzzy Clustering Method [C]//Fourth International Workshop on Advanced Computational Intelligence. Wuhan, China, 2011: 245–248. DOI: 10.1109/IWACI.2011.6160011

[9] HÖCHST J, BAUMGÄRTNER L, HOLLICK M, et al. Unsupervised Traffic Flow Classification Using a Neural Autoencoder [C]//IEEE 42nd Conference on Local Computer Networks (LCN). Singapore, Singapore, 2017: 523–526. DOI: 10.1109/LCN.2017.57

[10] JEONG Y S, JEONG M K, OMITAOMU O A. Weighted Dynamic Time Warping for Time Series Classification [J]. Pattern Recognition, 2011, 44(9): 2231–2240. DOI: 10.1016/j.patcog.2010.09.022

[11] CHEN T, GUESTRIN C, XGBoost: A Scalable Tree Boosting System [C]//ACM International Conference on Knowledge Dicovery and Data Mining. San Francisco, USA, 2016: 785–794

[12] LI Z, LEI Q, XUE K Y, et al. A Novel BP Neural Network Model for Traffic Prediction of Next Generation Network [C]//Fifth International Conference on Natural Computation, Tianjin, China, 2009: 32–38. DOI: 10.1109/ICNC.2009.673

[13] YANG W, YANG D Y, ZHAO Y L, et al. Traffic Flow Prediction Based on Wavelet Transform and Radial Basis Function Network [C]//International Conference on Logistics Systems and Intelligent Management (ICLSIM). Harbin, China, 2010: 969–972. DOI: 10.1109/ICLSIM.2010.5461098

[14] ZAHARIA M, CHOWDHURY M, FRANKLIN M, et al. Spark: Cluster Computing with Working Sets [C]//2nd Usenix Conference on Hot Topicsin Cloud Computing. Boston, USA, 2010: 10–10

[15] DEAN J, GHEMAWAT S. MapReduce: Simplified Data Processing on Large Clusters [C]//6th Symposium on Operating Systems Design & Implementaion. San Francisco, USA, 2004: 137–149

[16] SAKOE H, CHIBA S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1978, 26(1): 43–49. DOI: 10.1109/tassp.1978.1163055

[17] MA X L, DING C, LUAN S, et al. Prioritizing Influential Factors for Freeway Incident Clearance Time Prediction Using the Gradient Boosting Decision Trees Method [J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18 (9): 2303–2310. DOI:10.1109/tits.2016.2635719

Biographies

YE Dezhong received his B.S. degree from Jilin University, China in 1996. He is currently a chief big data engineer at ZTE Corporation. His research interests include big data in wireline communication and fixed network product.

LV Haibing received his B.S. degree from China University of Mining and Technology, China in 2002. He is currently a wireline product architecture director at ZTE Corporation. His research interests include fixed network products and IPTV systems.

GAO Yun received his B.E. degree from Nanjing University of Posts and Telecommunications (NUPT), China in 2016. He is currently pursing Ph.D. degree in NUPT. His research interests include machine learning, deep learning, and QoE in multimedia communication.

BAO Qiuxia received her B.E. degree from Nanjing University of Posts and Telecommunications (NUPT), China in 2017. She is currently pursing M.S. degree in NUPT. Her research interests include machine learning in big data, etc.

Chen Mingzi (1018010415@njupt.edu.cn) received her B.S. degrees in computer science from both New York Institute of Technology, USA and Nanjing University of Posts and Telecommunications (NUPT), China in 2018. She is currently pursing Ph.D. degree in NUPT. Her research interests include machine learning on big data, QoE in multimedia communication, and 5G tactile internet toward artificial intelligence.

A Network Traffic Prediction Method Based on LSTM

Based on LSTM

WANG Shihao, ZHUO Qinzheng, YAN Han, LI Qianmu, and QI Yong

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu 210094, China)

Abstract: As the network sizes continue to increase, network traffic grows exponentially. In this situation, how to accurately predict network traffic to serve customers better has become one of the issues that Internet service providers care most about. Current traditional network models cannot predict network traffic that behaves as a nonlinear system. In this paper, a long short-term memory (LSTM) neural network model is proposed to predict network traffic that behaves as a nonlinear system. According to characteristics of autocorrelation, an autocorrelation coefficient is added to the model to improve the accuracy of the prediction model. Several experiments were conducted using real-world data, showing the effectiveness of LSTM model and the improved accuracy with autocorrelation considered. The experimental results show that the proposed model is efficient and suitable for real-world network traffic prediction.

Keywords: recurrent neural networks; time series; network traffic prediction

1 Introduction

As Transmission Control Protocol/Internet Protocol (TCP/IP) networks become more and more important in modern society, how to better understand and correctly predict the behavior of the network is a vital point in the development of information technology. For medium/large network providers, network prediction has become an important task and get more and more attention [1]. By improving the accuracy of network prediction, network providers can better optimize resources and provide better service quality [2]. Not only that, network traffic prediction can help detect malicious attacks in the network too. For example, denial of service or spam attacks can be detected by comparing real traffic and predicting traffic [3]. The earlier these problems are detected, the more reliable network services can be obtained [4], [5].

With the development of computational science, some relatively reliable methods of network traffic prediction have been proposed to replace the intuitive prediction, especially in the field of sequential sequence prediction.

Nancy and George [6] used the time series analysis method to make an accurate prediction of the National Science Foundation Network (NSFNET) traffic data, and then carried out a series of research on the method of time series data, especially network traffic data prediction [7], [8]. Modeling methods include the autoregressive moving average model (ARMA), Autoregressive Integrated Moving Average model (ARIMA), Fractional Autoregressive Integrated Moving Average model (FARIMA), etc. Large-scale network system is a complex nonlinear system, and it is influenced by many external factors. Because of that its macroscopic flow behavior is often complex and changeable, and the data contains many kinds of periodic fluctuations. In addition, it shows non-linear trend and contains uncertain random factors, which may prevent the flow model with linear characteristics from being predicted accurately. There-

fore, how to select and optimize the nonlinear model is the key to predict network traffic in recent years. Support Vector Machine (SVM), Least Squares Support Vector Machine (LSS-VM) , artificial neural network, echo state network and so on all have certain improvement to the prediction accuracy. However, these models ignore the network traffic data from the autocorrelation, although they consider the nonlinear characteristics of data. Therefore, how to use nonlinear time sequence data to predict network traffic linked with autocorrelation characteristic is a problem to be solved eagerly.

2 Flow Prediction Model

This section describes recurrent neural networks (RNN), which are used for network traffic prediction, and also introduce a special type of RNN, long short-term memory networks (LSTM). On the basis of this, we put forward the model of LSTM combination with the characteristics of network traffic autocorrelation.

2.1 Recurrent Neural Networks

RNN is a popular learning method in the fields of machine learning and deep learning in recent years, which is different from the traditional feedforward neural network (FNN). FNN neurons transmit information through the input layer, hidden layer, and the connection of the output layer. Each input item is independent of others, and there is no connection between neurons in the same layer. However, RNN introduces the recurrent structure in the network, and establishes the connection of the neuron to itself. Through this circular structure, neurons can "remember" information from the previous moment in the neural network and influence the output of the current moment. Therefore, RNN can better deal with the data with time series, and in the prediction of timing data, it often performs better than FNN. The structure diagram of RNN is shown in **Fig. 1**.

The prediction process of RNN is similar to that of FNN, which is calculated by the forward propagation algorithm (**Algorithm 1**).

Algorithm 1. Forward propagation algorithm

For t from 1 to T do
$$u_t \leftarrow w_{hv}v_t + w_{hh}h_{t-1} + b_n$$
$$h_t \leftarrow e(u_t)$$
$$o_t \leftarrow W_{ot}h_t + b_o$$
$$z_t \leftarrow g(o_t)$$
End for

In Algorithm 1, T is the length of the input data, v_t is the input for time t, z_t is the output for time t. W_{hv}, W_{hh}, and W_{oh} are the link matrix of input layer to hidden layer, hidden layer to
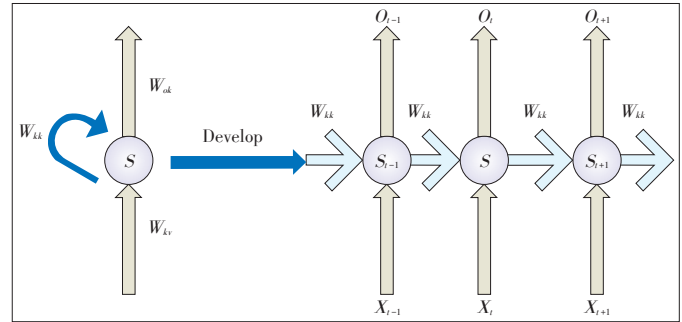
▲Figure 1. The recurrent neural networks structure diagram.

hidden layer and hidden layer to output layer. Functions e and g are the activation functions of the hidden layer and the output layer respectively.

There are some differences between RNN training and FNN. FNN is implemented by back propagation (BP) algorithm. However, RNN affects the error of the output layer because of the hidden layers of the previous moments, so the results of the backward propagation need to be propagated on the time dimension, named back propagation through time (BPTT) algorithm. The BPTT of RNN defines the partial derivative of the loss function to the t input value of the neuron j at time t firstly, and then solves the function with the derivative of the chain (**Algorithm 2**).

Algorithm 2. Back propagation through time (BPTT) algorithm

For t from 1 to T do
$$do_t \leftarrow g'(o_t) * \frac{dL(z_t, y_t)}{dz_t}$$
$$db_o \leftarrow db_o + do_t$$
$$dW_{oh} \leftarrow dw_{oh} + do_t h_t^T$$
$$dh_t \leftarrow dh_t + W_{oh}^T do_t$$
$$du_t \leftarrow e'(u_t)dh_t$$
$$dW_{hv} \leftarrow dW_{hv} + du_t v_t^T$$
$$db_h \leftarrow db_h + du_t$$
$$dW_{hh} \leftarrow dW_{hh} + du_t h_{t-1}^T$$
$$dh_{t-1} \leftarrow W_h^h du_t$$
End for
Return $d\theta = [dW_{hv}, dW_{hh}, dW_{oh}, db_h, db_o, dh_o]$

The partial derivative between the loss function and the neuron is affected by the output layer of the current time t and the hidden layer of $t+1$ at the next moment. For each time step, we use the chain rule to sum up all the results in the time dimension and get the partial derivative of the loss function to the weight w of the neural network. The weight of the recursive neural network is updated by gradient descent method until the condition is met.

The final step in the RNN training process is that the gradi-

ent is multiplied repeatedly in the propagation process. If the eigenvalues $W_{hh} > 1$, this will result in a gradient explode; if the eigenvalues $W_{hh} < 1$, this will result in a gradient vanish [9]−[12],. For this problem, Hochreiter et al. [13] put forward a LSTM neural network.

2.2 Long Short Term Memory

LSTM neural network is a variant of RNN. The key is to replace the neurons with cell states. The cell state is delivered over the time chain, with only a few linear interactions, and information is easily maintained on cell units. Each cell contains one or more memory cells and three nonlinear summation units. The nonlinear summation unit is also called the "gate", which is divided into three kinds: "Input gate", "Output gate" and "Forget gate". They control the input and output of memory cells by matrix multiplication. The structure diagram of LSTM is shown in **Fig. 2** [14].

The forward propagation algorithm of LSTM is similar to RNN, and the input data is a time series of length *T*:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2}$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * C'_t, \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t * \tanh(C_t). \tag{6}$$

Eq. (1) is the parameter of forget gate, where 1 means "complete reservation", while 0 means "completely abandoned". Eqs. (2) and (3) calculate the value of the input gate. Then Eq. (4) is used to discard the information that we need to discard, plus useful information entered at time *t*. Eqs. (5) and (6) determine which part of the cell state we are going to output to the next neural network and the next time.

The BPTT algorithm of LSTM is similar to that in RNN. It starts from the end of the time series (time T), gradually reverses the gradient of each parameter, and then updates the network parameters with the gradient of each time step. The output value corresponding to the partial derivative of memory
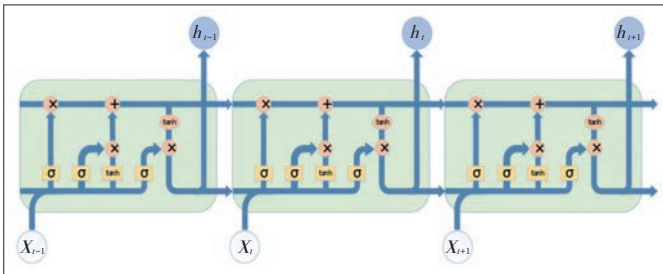
▲Figure 2. LSTM structural diagram of neural network model.

cells is calculated first, then the partial derivative of the output gate is calculated, and the corresponding partial derivatives of the memory cell state, forget gate and input gate are calculated respectively. Eventually, the model is updated using the gradient descent method connection weights.

2.3 Long Short Term Modeling Applicable to Autocorrelation Sequence

Time series is an ordered data set in time. Each data corresponds to the corresponding time. The time series model assumes that the data of the past can be applied to the future, and we can predict future data by studying the information of historical data.

In the conventional sequential sequence prediction problem, let's say the time series is $(y_1, y_2, y_3 \ldots y_t)$, y_t represents the data value at time *t*. We predict the value of y_{m+1} by using $(y_{m-k+1}, y_{m-k+2} \ldots y_m)$, where *k* represents the number of steps to be used for each prediction.

Four types of prediction can be defined according to the granularity of time [15]: current-time prediction, short-term prediction, medium-term prediction, and long-term prediction. The current-time prediction requires the shortest time interval between data, which can be used to establish an online real-time forecasting system. The short term interval is usually between an hour and a few hours, often used for optimal control or abnormal detection. The medium-term forecast time interval is one day, which can be used to guide resource planning. The long term interval is between months and years and can be used as a reference for strategy and economic investment.

The autocorrelation coefficients r_k is used to indicate the autocorrelation between the time sequence itself and the data of the lag *k* period [15]:

$$r_k = \frac{\sum_{t=1}^{T-k}(y_t - y')(y_{t+k} - y')}{\sum_{t=1}^{T}(y_t - y')}, \tag{7}$$

where $y_1, y_2 \ldots, y_T$ represent time series data, T represents the size of the dataset, y' represents the average of the sequence. The higher the value of r_k, the higher autocorrelation of that data in the period *k*. This means that the data set is in a time dimension and cycles through *k*. Data at the same time in different cycles tend to have the same data characteristics.

The traditional RNN sequence prediction model is based on the changing trend of the learning sequence to predict the value of next moment. According to our experience, the network traffic data has a certain autocorrelation in the 24 hours and 7 days cycles. This means that the data of 24 hours or 7 days at a certain time can be very good at representing the data characteristics of the present moment. However, because the data of 24 hours or 7 days ago does not constitute a continuous sequential relationship with T time steps before, the traditional RNN neural network does not apply to the modeling of autocorrela-

tion characteristics.

Using the improved LSTM + deep neural network (DNN) neural network and keeping the original LSTM network unchanged, the predicted value and several values before the autocorrelation cycle make the input of DNN neural network. By training the DNN network, the autocorrelation characteristics and timing characteristics of network data traffic are combined to achieve the accurate prediction of network data traffic. **Fig. 3** shows the Neural network structure of LSTM and DNN .

3 Network Traffic Prediction

In order to verify the accuracy of network traffic prediction model, three real network traffic data sets were selected for experiment. Data set A comes from the network traffic history data of network service provider in 11 European cities. The data was collected from 06:57 am on June 7, 2004 to 11:17 am on June 29, 2004 and collected every 30 seconds. Data set B comes from the web traffic history data from the United Kingdom Education and Research Networking Associations (UKERNA) website for academic research. Data was collected from 9:30 am November 2004 to 11:11 am on January 27, 2005, every five minutes. Data set C comes from the network traffic data collected by the backbone node of China's education network, Beijing University of Posts and Telecommunications, in 2012. Data is collected every minute

Based on the period and time of the three experimental data sets, this paper only considers the first two prediction types. According to Cortezs selection of time granularity [16], 5 minutes and 1 hour were used as current-time prediction and short-term prediction of time granularity. When we divide the training set and test set, we use the first two thirds of the data set for training, and then one third to test the accuracy of the model.

In order to evaluate the accuracy of the prediction model, the mean absolute percentage error (MAPE) is used as the evaluation indicator of the model [17]:

$$MAPE = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{y_n - Y_n}{y_n}\right| \times 100\%. \qquad (8)$$

As shown in Eq. (8), the average absolute percentage error is a commonly used index for estimating the accuracy of the model. In Eq. (8), y represents the real value of data, Y_n represents predicted value, and N is the test machine size. The smaller the error, the better the prediction.

Fig. 4 shows the autocorrelation analysis of
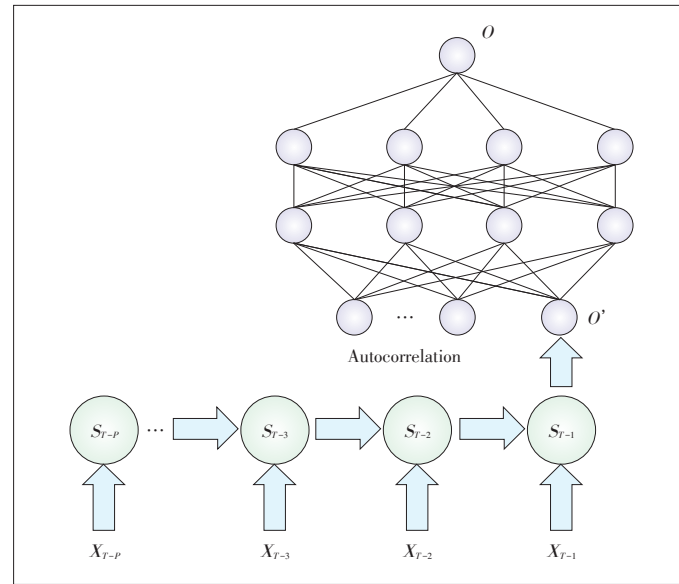
▲Figure 3. Neural network structure of LSTM and DNN.
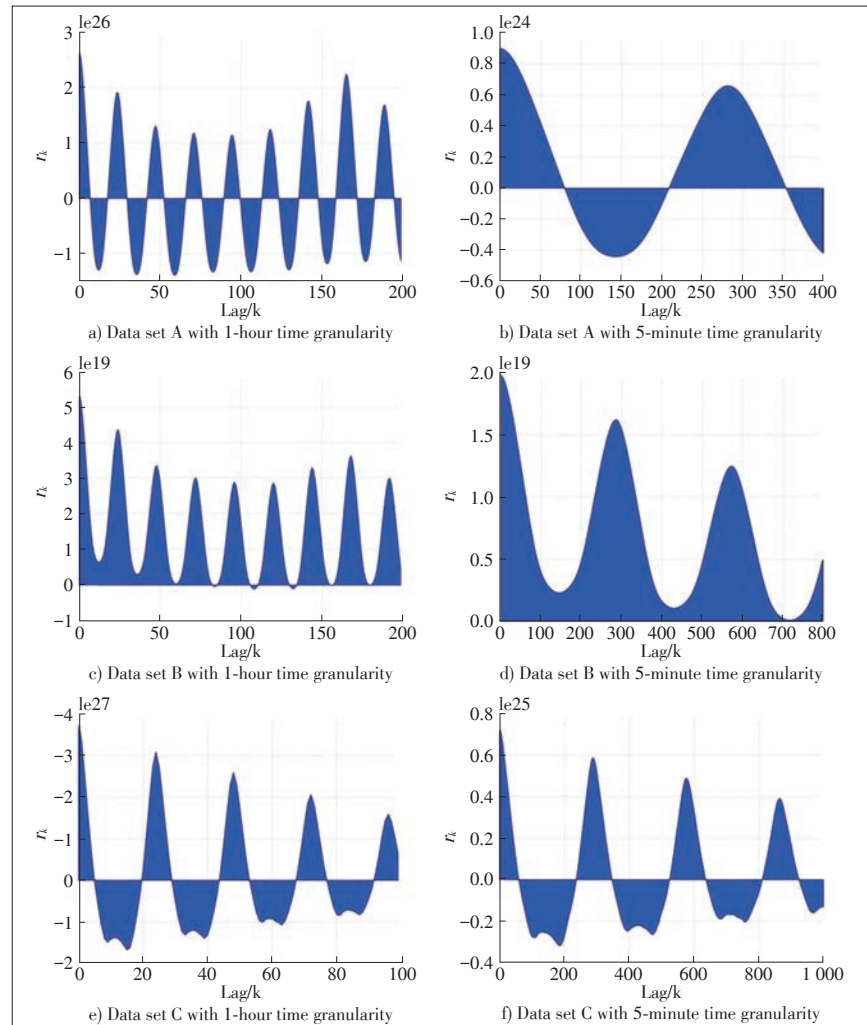
▲Figure 4. Autocorrelation analysis.

these data sets. For the data of 5minute granularity of time, the degree of autocorrelation is high when $k = 288$. And for the data of 1 hour for particle size, $k = 24$ and $k = 168$ were the two peaks. This indicates that the autocorrelation cycle of network traffic is 24 hours and 7 days, which is correspond with our characteristics of network traffic data.

In this paper, echo state network (ESN) is used as a comparison of LSTM model. The ESN is also a form of RNN, consisting of input layers, hidden layers, and output layers, and has a connection between the hidden layer and the hidden layer to retain information from the previous moments. Unlike RNN, the connection statuses of neurons in hidden layers are random, however, the connection weights are changeless. In the course of training, we only need to train the connection weights of the hidden layer to the output layer. After experiment, the parameters of the ESN model in **Table 1** are finally selected:

The neurons in the LSTM model were treated by Dropout in order to prevent the overfitting. Dropout was first proposed by Hintion [18], and the main idea was to let the neural network randomly make some neurons not work during the training. It is proved by experiment that this can effectively prevent the overfitting and improve the generalization ability of the model. Zaremba [19] improved the Dropout rate so that it could be applied to RNN. In the model of this paper, the probability that each neuron does not work is 10% during the training. **Table 2** shows LSTM neural network parameters.

In the LSTM-DNN model, the parameters of the LSTM neural network are consistent with Table 2. Depending on the time granularity of the data set, the values of different autocorrelation and the calculated value of LSTM feedforward algorithm were selected and entered into DNN. For example, a data set that time granularity is 5 minutes, $(xj - 287, xj - 288, xj - 289)$ is a good indicator of the data volume characteristics. There-

fore, put these data input current-time network traffic prediction model, the result we get is predicted traffic data. In a data set of time granularity 1 hour, the data $(xj - 23, xj - 24, xj - 25)$ is fed into the current-time network traffic prediction model to predict xj.

The average absolute percentage errors of the three algorithms are shown in **Table 3**.

The relationship between loss function and training frequency in LSTM and LSTM-DNN is shown in **Fig. 5**.

Due to the limitation of data set length, the test set divided

▼Table 3. Average absolute percentage error analysis

Date	Time granularity	ESN	LSTM	LSTM and DNN
Date set A	5 min	2.81%	1.41%	1.39%
	1 h	7.63%	4.69%	4.51%
Data set B	5 min	3.50%	3.29%	1.32%
	1 h	5.04%	4.47%	2.80%
Data set C	5 min	4.41%	12.04%	
	1 h	0.59%	14.00%	

DNN: deep neural network   ESN: echo state network   LSTM: long short-term memory network

▼Table 1. Echo status network parameters

Parameter	Value
In size	1
Reservoir size	1 000
Leaking rate	0.3
Spectral radius	0.136
Linear regression algorithm	Ridge regression

▼Table 2. LSTM neural network parameters

Parameter	Value
Dropout fraction	10%
Time steps	10
RNN units	200
RNN layers	1
Dense units	[10,10]
Batch size	10

RNN: recurrent neural network

a) Data set A with 1-hour time granularity

b) Data set B with 1-hour time granularity

DNN: deep neural network   LSTM: long short-term memory network

▲Figure 5. Comparison of performance of loss function under different data sets and training iterations.

by the data set C is not satisfied with the requirements of LSTM-DNN training and prediction. It can be seen from Table 3 that LSTM can effectively predict network traffic and have good performance at different time granularity of two data sets. According to the optimized LSTM-DNN model, the autocorrelation of data is added; when the time granularity was 5 m, the accuracy was slightly increased, but the accuracy of the prediction was significantly improved when the time granularity was 1 h. This indicates that the autocorrelation makes up for the insufficient data training that LSTM requires. The LSTM-DNN model has some advantages over traditional LSTM in the case of coarse time granularity or less data volume. We can see from Fig. 5 that as the complexity of neural network layer increases and the number of layers increases, LSTM-DNN is relatively slow and not easy to converge compared to LSTM in the training effect. **Figs. 6−9** show the prediction results of LSTM-DNN.

4 Conclusions

Aiming at implementing the network traffic with autocorrelation, this paper proposes a model of neural network which combines LSTM with DNN. By using several real data sets our experiments show that LSTM can be used as a timing sequence forecast model well. It can provide better performance than the other traditional model. Making use of the autocorrelation features, the neural network of LSTM and DNN has certain advantages in the accuracy of the large granularity data sets, but there is a requirement for the data size. High accuracy network traffic prediction provides some support to deal with possible network congestion, abnormal attack, etc. LSTM is just one of

▲Figure 8. Prediction effect of the data set B with 1-hour time granularity.
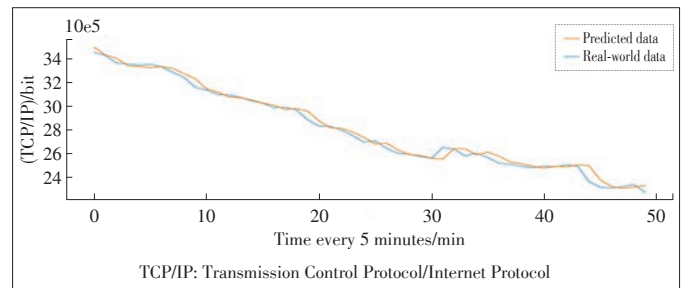
▲Figure 9. Prediction effect of the data set B with 5-minute time granularity.

the many variations of RNN. Some of other RNN models are also popular and well represented in many problems, such as Gated Recurrent Unit (GRU). The next step is to explore different RNN structures and combine the characteristics of network traffic data to explore the possibility of further improving the prediction accuracy.

▲Figure 6. Prediction effect of the data set A with 1-hour time granularity.

▲Figure 7. Prediction effect of the data set A with 5-minute time granularity.

References

[1] BABIARZ R, BEDO J. Internet Traffic Midterm Forecasting: a Pragmatic Approach Using Statistical Analysis Tools [M]//BOAVIDA F, PLAGEMANN T, STILLER B, et al. (eds). Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems. Lecture Notes in Computer Science: vol 3976. Berlin, Heidelberg, Germany: Springer, 2006: 111−121

[2] ALARCON-AQUINO V, BARRIA J A. Multiresolution FIR Neural-Network-Based Learning Algorithm Applied to Network Traffic Prediction [J]. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 2006, 36(2): 208−220. DOI: 10.1109/tsmcc.2004.843217

[3] KRISHNAMURTHY B, SEN S, ZHANG Y, et al. Sketch-Based Change Detection: Methods, Evaluation, and Applications [C]//3rd ACM SIGCOMM Conference on Internet Measurement. Miami Beach, USA, 2003: 234− 247. DOI: 10.1145/948205.948236

[4] YANG M M, ZHU T Q, ZHOU W L, et al. Attacks and Countermeasures in Social Network Data Publishing [J]. ZTE Communications, 2016, 14(S0): 2− 9. DOI: 10.3969/j.issn.1673-5188.2016.S0.001

[5] CHRISTIAN J, MOHAMED B. A Software-Defined Approach to IoT Networking [J]. ZTE Communications, 2016, 14(1): 61−68. DOI: 10.3969/j.issn.1673 - 5188.2016.01.009

[6] GROSCHWITZ N K, POLYZOS G C. A Time Series Model of Long-Term NSF-NET Backbone Traffic [C]//International Conference on Communications. New Orleans, USA, 1994: 1400−1404. DOI: 10.1109/ICC.1994.368876

[7] BASU S, MUKHERJEE A, KLIVANSKY S. Time Series Models for Internet Traffic [C]//Conference on Computer Communications. San Francisco, USA, 1996: 611−620. DOI: 10.1109/INFCOM.1996.493355

[8] SHU Y T, WANG L, ZHANG L F, etc. Internet Network Business Forecast Based on FARIMA Model [J]. Chinese Journal of Computers, 2001, 24(1): 46−54

[9] LIU X W, FANG X M, QIN Z H, et al. A Short-Term Forecasting Algorithm for Network Traffic Based on Chaos Theory and SVM [J]. Journal of Network and Systems Management, 2011, 19(4): 427−447. DOI: 10.1007/s10922-010-9188-3

[10] WANG H F, HU D J. Comparison of SVM and LS-SVM for Regression [C]//International Conference on Neural Networks and Brain. Beijing, China, 2005: 279−283. DOI: 10.1109/ICNNB.2005.1614615

[11] BENGIO Y, SIMARD P, FRASCONI P. Learning Long-Term Dependencies with Gradient Descent is Difficult [J]. IEEE Transactions on Neural Networks, 1994, 5(2): 157−166. DOI: 10.1109/72.279181

[12] HOCHREITER S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions [J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998, 6(2): 107−116. DOI: 10.1142/s0218488598000094

[13] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory [J]. Neural Computation, 1997, 9(8): 1735−1780. DOI: 10.1162/neco.1997.9.8.1735

[14] COLAH. Understanding LSTM Networks [EB/OL]. (2015-08-27). http://colah.github.io/posts/2015-08-Understanding-LSTMs

[15] DING X, CANU S, DENOEUX T. Neural Network Based Models for Forecasting [C]//Proc. ADT'95. New York, USA: Wiley and Sons, 1995: 243−252

[16] CORTEZ P, RIO M, ROCHA M, et al. Multi-Scale Internet Traffic Forecasting Using Neural Networks and Time Series Methods [J]. Expert Systems, 2010. DOI: 10.1111/j.1468-0394.2010.00568.x

[17] FARIA A E Jr. Review of Forecasting: Methods and Applications (by MAKRIDAKIS S, WHEELWRIGHT S C, HYNDMAN R J, Third Edition) [J]. International Journal of Forecasting, 2002, 18(1): 158−59. DOI: 10.1016/s0169-2070(01)00130-3

[18] HINTON G E, SRIVASTAVA N, KRIZHEVSKY A, et al. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors [EB/OL]. (2012-07-03). https://arxiv.org/abs/1207.0580

[19] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent Neural Network Regularization [EB/OL]. (2014-09-08). https://arxiv.org/abs/1409.2329

Biographies

WANG Shihao is a postgraduate student of Nanjing University of Science and Technology, China. He received the bachelor's degree from Nanjing Institute of Technology, China in 2017. His research interests include network traffic prediction and network intrusion detection.

ZHUO Qinzheng was a postgraduate student of Nanjing University of Science and Technology, China. He received the bachelor's and master's degree from Nanjing University of Information Science and Technology, China in 2015 and 2018. His research interests include network traffic prediction, data mining, and deep learning.

YAN Han received the Ph.D. degree from Nanjing University of Information Science and Technology in 2000. He is an associate professor with Nanjing University of Science and Technology, China. His research interests include software modeling, web computation, information security, and agile software development, and his current focus is on computing system management. He received the second class prizes for national defense science and technology. More than 40 academic papers have been published.

LI Qianmu (liqianmu@126.com) received the B.Sc. and Ph.D. degrees from Nanjing University of Science and Technology, China in 2001 and 2005, respectively. He is a professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He is also a Ph.D. supervisor and the director of Department of Software Engineering. His research interests include information security, computing system management, and data mining. He received the China Network and Information Security Outstanding Talent Award and multiple Education Ministry Science and Technology Awards. More than 120 high-level papers have been published and more than 150 patents have been applied and authorized.

QI Yong is a Ph.D. supervisor and professor with Nanjing University of Science and Technology, China. He is also the director of Jiangsu Intelligent Transportation Information Perception and Data Analysis Engineering Laboratory. His main research interests include data mining and intelligent transportation systems. He received more than 10 scientific awards, such as China University Research Cooperation Innovation Award, the Ministry of Educations Scientific and Technological Progress Award, and the Jiangsu Science and Technology Award. More than 100 academic papers have been published, and more than 60 patents for inventions and applications have been granted.

Potential Off-Grid User Prediction System Based on Spark

LI Xuebing[1,3], SUN Ying[1,2], ZHUANG Fuzhen[1,2], HE Jia[1,2], ZHANG Zhao[1,2], ZHU Shijun[4], and HE Qing[1,2]

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China;
3. College of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China;
4. ZTE Corporation, Shenzhen, Guangdong 518057, China)

Abstract: With the increasingly fierce competition among communication operators, it is more and more important to make an accurate prediction of potential off-grid users. To solve the above problem, it is inevitable to consider the effectiveness of learning algorithms, the efficiency of data processing, and other factors. Therefore, in this paper, we, from the practical application point of view, propose a potential customer off-grid prediction system based on Spark, including data pre-processing, feature selection, model building, and effective display. Furthermore, in the research of off-grid system, we use the Spark parallel framework to improve the gcForest algorithm which is a novel decision tree ensemble approach. The new parallel gcForest algorithm can be used to solve practical problems, such as the off-grid prediction problem. Experiments on two real-world datasets demonstrate that the proposed prediction system can handle large-scale data for the off-grid user prediction problem and the proposed parallel gcForest can achieve satisfying performance.

Keywords: data mining; off-grid prediction; Spark; parallel computing; deep forest

1 Background

In recent years, the competition among communication operators is fiercer with the advent of the 4G era. The communication operators face new challenges in many ways, including fine management, precision marketing, products innovation, promotion of user terminals, retaining users, data analysis, and so on. How to win the competition? It plays a significant role that doing your best to retain old users while attracting new users through improving service quality and others. Therefore, it is an urgent need to solve the problem that the operators can make a good prediction of potential off-grid users.

The past decade has witnessed the remarkable growth of Internet communication technology and sensor networks for perceiving and obtaining information. There is a huge amount of data which has great potential values in all walks of life, including the telecommunications industry. Big data is high-volume, high-velocity, and high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery, and process optimization [1]. The potential values of big data need advanced data mining techniques [2], [3] to discover and they have drawn much attention from the practitioners. Now there are many good algorithms to achieve latent useful information such as neural networks, cluster analysis, genetic algorithms, decision trees, decision rules, support vector machines, and more. Nevertheless, conventional approaches come across significant challenges when computing power and memory space are limited in big data era. So distributed technology is necessary to solve the problem of limited computing power and memory space. Apache Hadoop, which is an open-source software framework used for distributed stor-

age and processing of dataset of big data based on MapReduce programming model [4]－[7], becomes a good framework to solve the above problem. The main idea of the MapReduce model is to hide details of parallel execution and allow users to focus only on data processing strategies. MapReduce has brought revolution in computing model, and also provides a new processing platform for data mining environment. However, Hadoop processes data from disk which makes it inefficient for data mining applications that often require iteration. Spark is a more recent distributed framework that works with Hadoop and provides in‑memory computation that allows iterative jobs to be processed much faster, so it is a more suitable base for data mining [8].

Previous works mainly focus on analyzing off-grid user problem from the angle of algorithms. GUI et al. studied the main technology of customer loss prediction and analyzed the advantages and disadvantages of the decision tree, neural network, and so on. Louis [9], Rosset [10] and Nash [11] have studied customer churn prediction as a classification problem and used Logistic Regression, Decision Tree and Bayes to establish a customer churn prediction model. However, the ultimate aim of algorithm research is to solve the practical problem. In this paper, we propose a complete solution—a potential customer off‑grid prediction system based on Spark—to solve the problem. And the data mining techniques include data storage, data clearing, feature selection, training model, and result showing. Furthermore, we use the Spark parallel framework to improve the gcForest algorithm which is a state‑of‑the‑art decision tree ensemble approach. The new parallel gcForest algorithm can be used to solve off‑grid prediction problem. Moreover, the parallel gcForest can be used to solve more practical problems.

The rest of this paper is organized as follows. In section 2, we introduce techniques related to our proposal. In sections 3 and 4, we introduce the system architecture and parallel gcForest in detail, which are both under the Spark computational framework. In section 5, we analyze experimental results. Section 6 concludes this paper.

2 Related Technologies

In this section, we will introduce the mostly related techniques of prediction system, involving Spark computing framework, Hadoop Distributed File System (HDFS), MapReduce, and Hive.

2.1 Apache Spark

Apache Spark is an open‑source cluster computing framework, which runs programs in memory, developed at the University of California, Berkeley's AMPLab. As a MapReduce‑like cluster computing engine, Spark also possesses good characteristics such as scalability and fault tolerance as MapReduce does. However, Apache Spark has its architectural foun-

dation the resilient distributed dataset (RDD). This is a read‑only multiset of data items distributed over a cluster of machines and users can explicitly cache an RDD in memory across machines and reuse it in multiple MapReduce-like parallel operations. Therefore, Spark can be well qualified to process iterative jobs, including PageRank algorithm, K-means algorithm, etc. Besides, it can outperform Hadoop by 10x in iterative machine learning jobs. Spark has strong fault‑tolerant performance through a notion of lineage: If a partition of an RDD is lost, the RDD has enough information about how it was derived from other RDDs to be able to rebuild just that partition. With the help of the lineage, Spark recovers the lost data quickly and effectively. Spark shows great performance in processing iterative computation because it can reuse intermediate results and keep data in memory across multiple parallel operations.

Currently, Spark has devolved into a large data processing platform based on memory computing, and its architecture includes resource management, distributed storage, data processing, and application. Berkeley called the entire ecosystem of Spark as the Berkeley data analysis stack (BDAS). For resource management, Spark supports standalone (native Spark cluster), Hadoop YARN, or Apache Mesos. For distributed storage, Spark can interface with a wide variety, including Hadoop Distributed File System (HDFS), MapR File System (MapR‑FS), Cassandra, and more. Furthermore, Spark has a distributed memory file system, Tachyon, which is used to cache data in memory more than 100 times faster than HDFS. Data processing engine is the Spark computing framework based RDDs, which are the foundation of the overall project. On the top of Spark, there are many distributed subprojects, such as Spark Streaming, GraphX, Spark Structured Query Language (SQL), MLlib, and more. In addition, as shown in **Fig. 1**, there is a very important advantage that Spark ecosystem is compatible with the Hadoop ecosystem.

2.2 Hadoop Distributed File System

HDFS is the system component of Hadoop, which is designed to store very large data sets reliably and to stream those data sets at high bandwidth to user applications [12]. Comparing with other existing distributed file systems, HDFS has

HDFS: Hadoop Distributed File System　　MR: Map Reduce
MPI: Message Passing Interface　　YARN: Yet Another Resource Negotiator
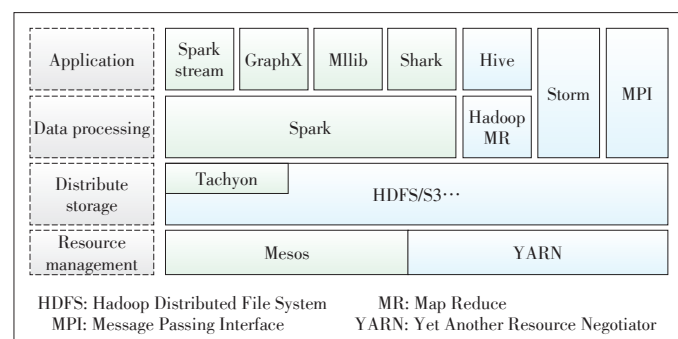
▲Figure 1. Spark Ecosystem.

LI Xuebing, SUN Ying, ZHUANG Fuzhen, HE Jia, ZHANG Zhao, ZHU Shijun, and HE Qing

some advantages as follow. On the one hand, HDFS architecture ensures that it could detect fault and automatically recover quickly from those faults. Therefore, HDFS is highly fault-tolerant. Moreover, it can be deployed on low-cost hardware. On the other hand, HDFS provides high throughput access to application data and is suitable for applications that have large datasets.

2.3 MapReduce

MapReduce [13] is such a programming model that provides an associated implementation for processing and generating large-scale data sets using Map( ) and Reduce( ) procedures. Specifically, users design a map function that takes a set of data and converts it into another set of data, where individual elements are broken down into <key, value> pairs and a reduce function takes the output from a map as input and combines those data <key, value> pairs into a smaller set of <key, value> pairs. Generally, the reduce job is performed after the map job.

The major advantage of MapReduce is that it processes and generates big data sets with a distributed, parallel algorithm on a computer cluster. We use a typical example to promote the understanding of MapRecuce idea. Consider the following pseudo-code for a task that counts the appearance of each word in a set of documents:

```
map(String key, String value):
// key: document name
// value: document contents
for each word w in value: output(w, 1);

reduce(String key, Iterator values):
// key: a word
// values: a list of counts
int sum count = 0;
for each v in values:
sum count += v;
```

Here, the map function counts each word which appears in the document. And the reduce function sums sum all of its input values get the total counts of a single word.

2.4 Hive

Hive is a data warehouse software project to facilitate reading, writing, and managing of large datasets in distributed storage using SQL and it builds on Hadoop. In general, data query, statistics, and other data operations on distributed data need to be implemented in the MapReduce Java Application Programming Interface (API) to execute SQL applications. This way is very inconvenient. To solve the problem, Hive provides the necessary SQL abstraction to integrate SQL-like queries into the underlying Java without the need to implement queries in the low-level Java API. Hive provides a SQL-like interface to

handle data stored in a distributed file system that integrates with Hadoop. Since most data warehousing applications work with SQL-based querying languages, Hive defines SQL-like query language, called HQL (Hibernate Query Language), which allows users who are familiar with SQL to query data on Hadoop.

Hive supports the analysis of large-scale data stored in the HDFS file system. It provides HQL and converts queries to MapReduce and Spark jobs which can run in Hadoop YARN. Major components of the Hive architecture are shown in **Fig. 2**. Metastore stores metadata for each of the tables such as the name of the table, the partition of the table, and its properties. Metadata is highly crucial because it can help the driver to track the data. A driver plays a role of receiving the HQL statements, which likes a controller. And it creates sessions to start the execution of statement and monitors the progress of the execution. The compiler, optimizer, and executor complete HiveQL query statements from lexical analysis, syntax analysis, compilation, optimization, and query plan generation. The generated query plan is stored in the HDFS and then executed. The command-line interface (CLI), user interface (UI), and thrift server are user interfaces which mainly use for user interaction. Hive is simple to learn and use. It can quickly implement MapReduce statistics through SQL-like language and is very suitable for statistical analysis for the big data which stores distributed system.

3 System Architecture

We present a complete solution to deal with the problem of off-grid user prediction in this paper, named the potential off-grid user prediction system. The whole system is built on Spark, a distributed computing framework and can be applied to large-scale data. The proposed system architecture is shown in **Fig. 3**.

In Fig. 3, the system includes four parts, i.e., data storage, data pre-processing, statistical analysis, training model and user interface. In the following subsections, we will detail the

CLI: Command-Line Interface   ODBC: Open Data Base Connectivity
JDBC: Java Data Base Connectivity   UI: User Interface
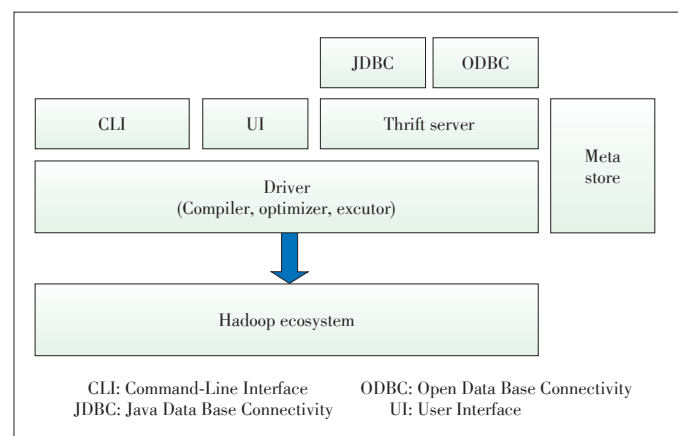
▲Figure 2. Hive architecture.
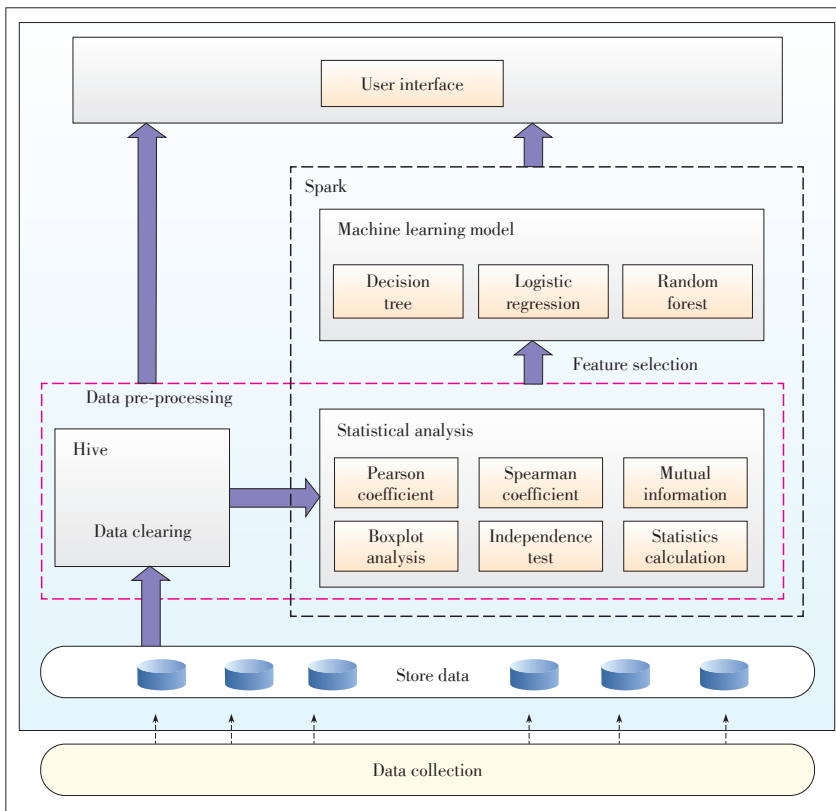
▲Figure 3. Architecture of the potential off-grid user prediction system based on Spark.

structure of the whole system, including the data processing process, the algorithm, and the specific design of the system.

3.1 Data Pre-Processing

The data pre-processing plays an important role in data mining. The phrase "garbage in, garbage out" are particularly applicable to data mining and machine learning and also prove that data quality is very important for achieving good performance. If there is much irrelevant and redundant information present or noisy and unreliable, knowledge discovery during the training phase will be more difficult. In our system, the pre-processing of operation is divided into two parts, including data cleaning and statistical analysis with the purpose of feature selection.

3.1.1 Data Cleaning

Data cleaning[1] is the process of detecting and correcting (or removing) corrupted or inaccurate records from a database, involving incomplete, incorrect, inaccurate or unrelated parts of the data, and then replacing, modifying, or deleting the dirty or course data. In the system, we have the following ways to do simple data cleaning.

● Removing obvious useless feature. There are a few use-

---

[1] https://en.wikipedia.org/wiki/Data cleansing

less features in the user data provided by the operator, which seriously affect the final result performance. So we need to find and get rid of them. For example, a feature that has a large number of null values or the same values is an obvious useless feature.

● Data transformation. This function allows the mapping of the data from its given format into the format expected by the appropriate application. This is important for the training model. For instance, for the non‑vector feature in the data, we need to convert it to a vector feature that can be computed. In addition, there are some features that need to be normalized to enhance the performance of training.

● Labelling data. When the system reads data from the HDFS and saves it to the hive, the data is not labeled. This function is to label the data according to the rules which is essential to the training of off-grid models.

Based on Hive, we can also do a lot of operations of processing data. The function‑based Hive can continue to expand in our system.

3.1.2 Statistical Analysis

In our system, the major purpose of statistical analysis is to make feature selection better. The simple data clearing in the previous step has cleared the obvious useless features, and this step is further feature selection. As we all know, feature selection, or called variable selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. There are four reasons to do feature selection:

● Simplification of models to make them easier to interpret by researchers

● Reducing training times

● Removing redundancy and irrelevant features to avoid the curse of dimensionality

● Enhancing generalization by reducing overfitting and make a better model.

The analytical methods used in the system are divided into two categories: the correlation analysis of features and data distribution analysis. The correlation analysis of features is very important in the process of data pre‑processing. The main objective of feature analysis is to eliminate extraneous and redundant features without losing important information. After acquiring the data in the machine learning task, it is usually necessary to do correlation analysis of feature and select feature correlation. There are four correlation analysis methods we used, involving the pearson correlation coefficient, spearman's rank correlation coefficient, mutual information and chi‑squared test.

(1) Pearson correlation coefficient:

In statistics, the pearson correlation coefficient is a measure of the linear correlation between two variables X and Y by a value between 1 and -1. For example, 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation. The definition of pearson correlation coefficient is the covariance of the two variables. And it is always represented by ρ when applied to a population. The formula for $ρ$ is Eq. (1).

$$\rho(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}, \tag{1}$$

where cov is the covariance; σ_X is the standard deviation of X and σ_Y is the standard deviation of Y. When applied to a sample, it is commonly represented by the letter r and may be called the sample correlation coefficient. We can obtain a formula for r by substituting estimates of the co‑variances and variances based on a sample into the formula above. So if we have one dataset $\{x_1, \cdots, x_n\}$ containing n values and another dataset $\{y_1, \cdots, y_n\}$ containing n values, that formula for r is:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{2}$$

where n is the sample size; x_i and y_i are the single samples indexed with i; $\bar{x} = (1/n)\sum_{i=1}^{n}x_i$, which is used to calculate the sample mean. The way to calculate $\bar{y}$ is similar.

(2) Spearman's rank correlation coefficient:

In statistics, Spearman's rank correlation coefficient is a nonparametric measure of rank correlation, i.e., the statistical dependence between the rankings of two variables. It assesses how the relationship between two variables is described by monotone functions. The difference of pearson correlation and spearman correlation is that the former assesses linear relationships, while the later assesses monotonic relationships [14].

Spearman's rank correlation coefficient often be denoted as r_s. If we have two variables whose sizes are n and X_i, Y_i are their raw scores, X_i, Y_i are converted to ranks rgX_i, rgY_i. The formula for r_s is Eq. (3).

$$r_s = \frac{\text{cov}(rg_X, rg_y)}{\sigma_{rg_x}, \sigma_{rg_y}}, \tag{3}$$

where $\text{cov}(rg_X, rg_y)$ is the covariance of the rank variables and $\sigma_{rg_x}, \sigma_{rg_y}$ are the standard deviations of the rank variables.

(3) Mutual information:

In probability theory and information theory, the mutual information of two random variables is a measure of the mutual dependence between the two variables. It is one of many quantities that measures how much one random variable tells us

about another. It is a dimensionless quantity with units of bits and can be thought of as the reduction in uncertainty about one random variable given knowledge of another. High mutual information indicates a large reduction in uncertainty; low mutual information indicates a small reduction; and zero mutual information between two random variables means the variables are independent. Formally, the mutual information of two discrete random variables X and Y can be defined as Eq. (4).

$$I(X;Y) = \sum_{y \in Y}\sum_{x \in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right), \tag{4}$$

where $p(x,y)$ is the joint probability function of X and Y; $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively. In the case of continuous random variables, the summation is replaced by a definite double integral:

$$I(X;Y) = \int_Y \int_X p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)dx\, dy, \tag{5}$$

where $p(x,y)$ is now the joint probability density function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y respectively. If the log base 2 is used, the units of mutual information are bits.

(4) Chi-squared test:

The chi-squared test is a common hypothesis testing method based on the χ^2 distribution. It is the deviation between the actual observed value and the theoretical inference value of the statistical sample. The deviation between the actual observation value and the theoretical inference value determines the size of the chi square value. The larger the chi square value is, the more it does not conform; the smaller the value of the value, the smaller the deviation, the more consistent. If two values are exactly equal, the chi square value is 0, and it indicates that the theoretical value is in full conformity.

As we mentioned above, the analytical methods for feature selection are divided into two categories: correlation analysis of features and data distribution analysis. In our system, we mainly use box plot to do data distribution analysis.

(5) Box plot:

The box plot is a method for graphically depicting groups of numerical data through their quartiles. Box plots are useful for identifying outliers and for comparing distributions. Box plots may also have lines extending vertically from the boxes indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot. Outliers may be plotted as individual points which are out of upper extreme or lower extreme. The values of upper extreme and upper extreme are calculated by the interquartile range (IQR)². **Fig. 4** shows two kinds of boxplots. The first one is a generic example of box plot with the maximum, third quartile, median, first quartile, and minimum values labeled. And the second one is that outliers are plotted as individual dots and are in‑line with whiskers. The relative

---

² https://en.wikipedia.org/wiki/Interquartile range
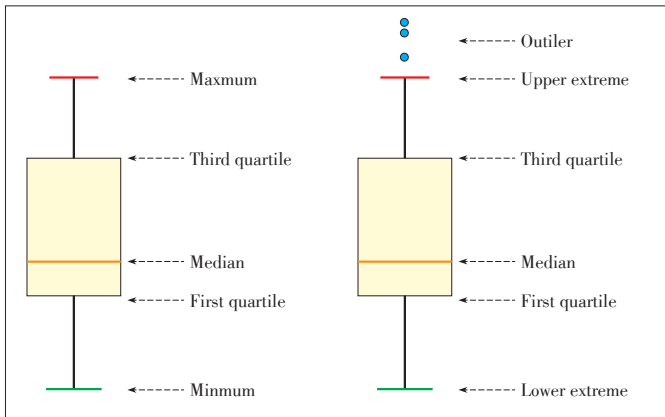
▲Figure 4. Two kinds of box plots.

vertical spacing between the labels reflects the values of the variable in proportion. Furthermore, in some situations, two or more box plots can be placed side‑by‑side on a coordinate plane to show how a phenomenon or scenario evolves with time, which is plotted along the independent‑variable or x‑axis. In our system, the box plot plays an important role in data distribution analysis and feature selection.

3.2 Machine Learning Algorithm

In practical applications, our prediction system based Spark integrates three machine learning algorithms to solve the problem of off‑grid user prediction, including logistic regression, decision tree, and random forest. The system also supports the extension to integrate new machine learning methods. Next, we will introduce the three algorithms and the principle of parallelization based Spark.

(1) Logistic Regression

Logistic regression is one of the most commonly-used prediction methods. It is used with data in which there is a binary outcome variable, or where the outcome takes the form of a binomial proportion. Like linear regression, one estimates the relationship between predictor variables and an outcome variable. In logistic regression, however, one estimates the probability that the outcome variable assumes a certain value, rather than estimating the value itself. It uses a logistic function, as shown in Eq. (6), to describe the probabilities of outcomes.

$$f(z) = \frac{1}{1 + e^{-z}}. \tag{6}$$

If y is the possibility that sample x is a positive example as well as parameter is w and b, we can gain Eqs. (7) and (8).

$$p(y = 1 | x) = \frac{e^{w^{\mathrm{T}} x + b}}{e^{w^{\mathrm{T}} x + b} + 1}, \tag{7}$$

$$p(y = 0 | x) = \frac{1}{e^{w^{\mathrm{T}} x + b} + 1}. \tag{8}$$

Then we'll estimate w and b with maximum likelihood meth-

od. For a given dataset $\{(x_i, y_i)\}_m^{i=1}$, we need to maximize log‑likelihood, as shown in Eq. (9), which is the objective function of the model. Then we could use optimization methods in convex optimization theory, such as gradient descent method and Newton method, to get his optimal solution.

$$l(w, b) = \sum_{i=1}^{m} \ln p(y_i | x_i; w, b). \tag{9}$$

(2) Decision Tree

Decision trees and their ensembles are popular methods for the machine learning tasks of classification and regression. Decision trees are widely used since they are easy to interpret, handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture nonlinearities and feature interactions. Tree ensemble algorithms such as random forests and boosting are among the top performers for classification and regression tasks. Our system supports decision trees for binary and multiclass classification and for regression, using both continuous and categorical features. The implementation partitions data by rows, allowing distributed training with millions of instances.

(3) Random Forest

Random forests [15] are ensembles of decision trees, which are one of the most successful machine learning models for classification and regression. They combine many decision trees in order to reduce the risk of overfitting. Like decision trees, random forests handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture nonlinearities and feature interactions. Random forests train a set of decision trees separately, so the training can be done in parallel. The algorithm injects randomness into the training process so that each decision tree is a bit different. Combining the predictions from each tree reduces the variance of the predictions, improving the performance on test data. In training model, the randomness injected into the training process includes two methods, including subsampling the original dataset on each iteration to get a different training set (a.k.a. bootstrapping) and considering different random subsets of features to split on at each tree node. Then a random forest will aggregate the predictions from its set of decision trees to make a prediction on a new instance.

(4) Parallelization

Parallel implementation of the machine learning algorithm mentioned above is based on Spark. For a spark data mining program, generally, the relationship between the RDD and operations is shown in **Fig. 5**. It is necessary to complete a job through creation operation, transformation operation, control operation, and action operation. Of course, there are multiple jobs in a Spark application. In Fig. 5, these concepts are important in spark programming.

- RDD: The main abstraction Spark provides is a RDD, which is a collection of elements partitioned across the nodes

LI Xuebing, SUN Ying, ZHUANG Fuzhen, HE Jia, ZHANG Zhao, ZHU Shijun, and HE Qing

▲Figure 5. Spark program flowchart.

of the cluster that can be operated on in parallel.

● Creation operation: RDD initial creation is the responsibility of the SparkContex, and the memory of the collection or external file system is the input source.

● Control operation: RDD can be saved in disk or memory for later reuse.

● Action operation: Since Spark is lazy computing, doing action operation on any RDD will trigger a spark job run and produce a result.

According to the above basic knowledge of Spark parallelization, all the algorithms that we integrate in our system can achieve the parallelization based on Spark. We will take the parallel gcForest algorithm as an example to introduce the implementation of the parallel algorithm in Section 4.

3.3 System Detailed Design

The ultimate goal of a software system is to solve practical problems. Next, we will introduce the detailed design of our system through unified modeling language (UML) class diagrams. In software engineering, a class diagram in UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, attributes, operations (or methods), and the relationships among objects. The class diagram is an important part of object-oriented programming and used for detailed modelling translating the models into programming code. The detailed design of the proposed system at the programming level is shown in **Fig. 6**.

4 Parallel Deep Forest

4.1 Deep Forest

In this paper, in addition to a potential off-grid user predic-

tion system, we also improve the deep forest algorithm based on Spark to solve the problem of off-grid prediction. The deep forest algorithm, called gcForest [16], is a novel decision tree ensemble approach which could compete to deep neural networks in many tasks. As shown in **Fig. 7**, there are two crucial components in the gcForest. The main one is cascade forest structure which capacitates gcForest to do representation learning. The other one is multi-grained scanning structure and it is useful to enhance the representational learning ability of cascade forest structure.

Cascade forest is the core of the whole algorithm structure. As we all know, deep neural networks [17], [18] always depend on the layer-by-layer processing of feature to make the networks have representation learning ability. Similar to neural networks, gcForest also has its own ability to do representation relying on cascade forest. Each layer of the cascade forest is composed of multiple random forests [15]. And every forest would produce an estimate of class distribution. Then all estimated class distribution will form a class vector, which combines with the original feature vector to form new vector as input to the next layer. As we all know, the diversity is essential to ensemble model [19]. Therefore, in order to encourage diversity in cascade forest, gcForest use two kinds of random forests, one is random forest, and the other one is completely-random tree forest [20]. As illustrated in Fig. 7, there is an example of binary classification task which is solved by gcForest. In cascade structure, each layer is made up of four forests, including two random forests and two completely-random tree forests. Each of the four forests will produce a two-dimensional vector, and finally one layer gains eight-dimensional vector which will combine with original feature vector to pass to next layer. In addition, we conduct cross validation on each forest to reduce overfitting. Furthermore, after building a new level, the performance of this level will compare to the performance of the previous level, which is estimated on validation. The training procedure will automatically stop if there is no significant performance gain. Thus, it is an important advantage that gcForest adaptively decides its model complexity by terminating training.

Multi-grained scanning plays a crucial role in the gcForest. It can enhance the performance of cascade forest. It is well known that handling feature relationships is very important in deep neural networks. For example, using spatial relationships among the raw pixels in convolution networks is effective to solve image problem. Inspired by this recognition, multi-grained scanning is to make use of the feature relationships by sliding windows which are used to scan the raw features. The window size we can set is multiple values. As illustrated in Fig. 7, there are raw feature of 400-dim and two windows sizes, 50-dim and 100-dim, respectively. Then the instances extract-

◀Figure 6.
The unified modeling language (UML) class diagram of the proposed potential off‑grid user prediction system.

ed from sliding windows are produced to form new representation vectors through random forest finally.

4.2 Parallel gcForest

There is no denying that gcForest is state‑of‑the‑art ensemble algorithm based decision tree. However, there are also certain limitations; for example, the algorithm becomes particularly slow when the volume of data is large. In practical application, it is an unavoidable problem to deal with large‑scale data. Therefore, in our paper, we implement the parallelization of gc-

LI Xuebing, SUN Ying, ZHUANG Fuzhen, HE Jia, ZHANG Zhao, ZHU Shijun, and HE Qing
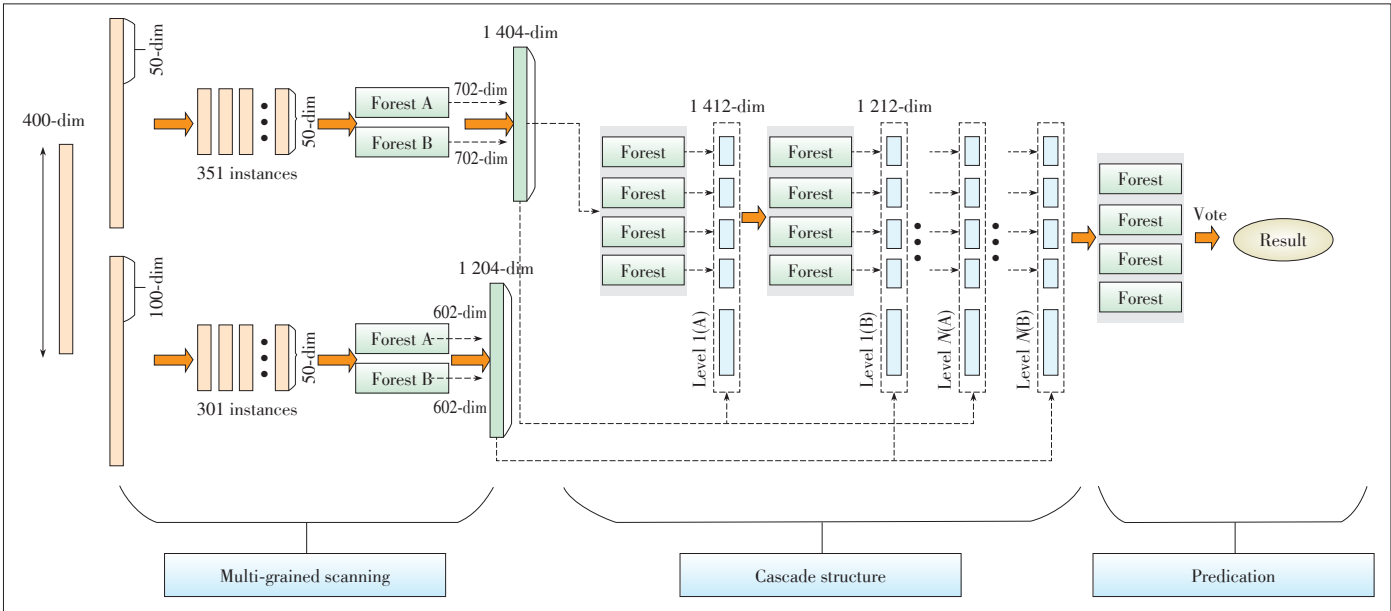
▲Figure 7. The gcForest architecture. Suppose there are two classes to predict, two sliding windows are used and original features are 400-dim.

Forest algorithm based on spark parallel framework, as shown in **Fig. 8**.

As mentioned above, the structure of gcForest algorithm is divided into two parts, including cascading structure, which capacitates gcForest to do representation learning, and multi‐grained scanning structure, which is useful to enhance performance. Similarly, the algorithm is divided into two modules to achieve the multi‐grained scanning structure and cascading structure. In the realization of parallelization, there are mainly the following innovations:

● The index‐based scanning algorithm. In multi‐grained scanning structure, the gcForest uses sliding windows to produce a large number sub‐instances by scanning the raw features. The result of this is that more storage space is needed.
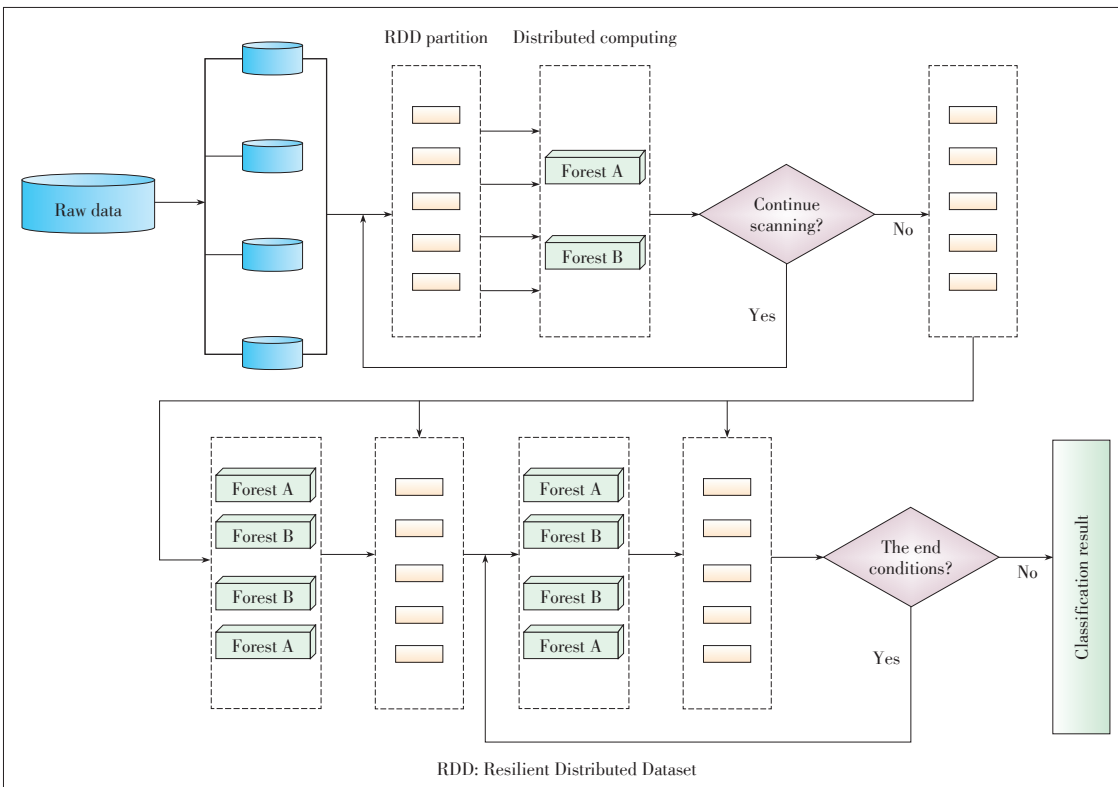
◄Figure 8.
The Parallel gcForest Algorithm.

To solve the problem, we put forward an index-based scanning algorithm. When scanning, we generate a unique ID for each generated sample, and then we record the sample information, such as the original sample number and the beginning and the end of the scan, which can find newly generated instances from the original sample. When we build a decision tree in a random forest, we produce samples based on recorded information. After building the decision tree, the sample is deleted and only the model is kept. By doing this, we save a lot of storage space.

• Random sampling to construct random forest. This method is also to solve the newly generated samples and requires a lot of storage space problems in multi-grained scanning structure. Before training the stochastic forest model, we sampled the newly generated distances randomly. In other words, we do not use all the newly generated distances.

• Parallelization of Random Forest algorithm. In cascading structure of gcForest, the main part of each layer is the random forest model, involving random forest and completely-random tree forest. These two kinds of random forest models are easy to be implemented based on spark framework. As shown in **Fig. 9**, the construction of decision tree in forest is distributed evenly to each node of the cluster, and all trees grow in parallel with different nodes in the cluster. When choosing the optimal partition of the node of the decision tree, the sub-partitioning strategy is used. Each node records the corresponding partition in the data partition of the nodes, summarizes the data, and then finds the best partition by traversing. In addition, the

growth of the decision tree in the forest is trained by the method of layer by layer.

5 Experiment

5.1 Datasets

In order to validate the effectiveness of our system and the parallel gcForest algorithm, we conducted experiments on two real-world datasets. The first dataset is from a phone company which is used for churn management. The second dataset is operation data of China Telecom users. The information of experimental data is shown in **Table 1**.

(1) Churn Management Dateset:

This dataset contains summarized data records for each customer for a phone company. Our goal is to build a model so that this company can predict potential churners. There are 21 variables to represent the customers' record. This is a common set of off-grid data and the meaning of few variables as shown in **Table 2**.

(2) Telecom Users Dataset:

The Telecom user dataset is provided by telecom operators, which records the behavior of Telecom users with 131 dimensions. In addition, this dataset is raw data, not processed or simply processed, and the data scale is very large. The purpose of our system is to process these dirty data step by model and practical application. The data size is shown in Table 1. A few features are shown in **Table 3** to facilitate the understanding of
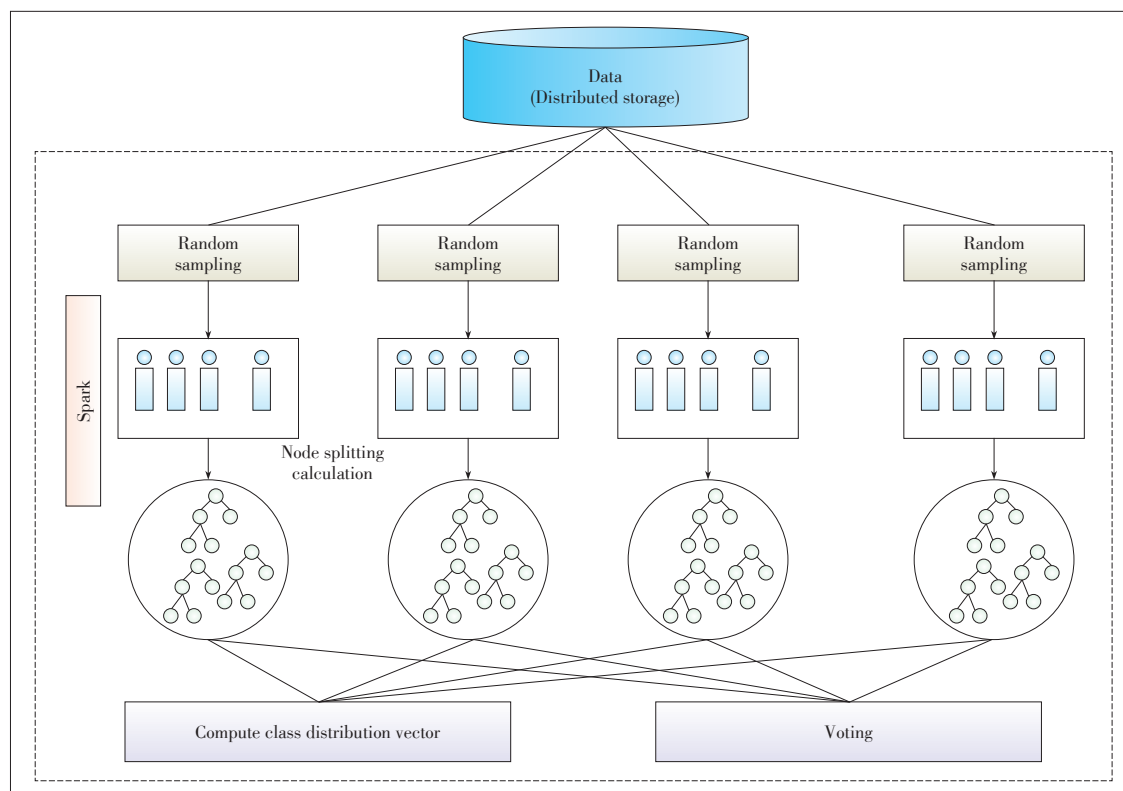
◀**Figure 9.**
Structure of the parallel random forest algorithm.

▼Table 1. Experiment data

Dataset	Dimension	Training size	Test size
Churn management dateset	21	2 000	1 033
Telecom users dataset	131	1 million+	1 million+

▼Table 2. Features in the churn management dataset

Feature	Meaning
Account_length	How long this person has been in this plan
International_plan	This person has international plan=1, otherwise plan=0
Voice_mail_plan	This person has voice mail plan=1, otherwise plan=0
Number_vmail_message	The number of voice mails

▼Table 3. Some features in the Telecom users dataset

Feature	Meaning
LatestServiceTime	Last service time before collecting information
CallerCount	The number of calling in a month
DataDuration	Time spent in the data business in a month
SmsSendTimesChange	The number of SMS messages sent in a month

telecom users dataset.

5.2 Experimental Results

We mainly conducted a system functional test. In this paper, we use precision, recall, accuracy and F1-score to evaluate the experiment results which are commonly used metrics in machine learning. The results of our experiment are shown in **Tables 4** and **5**.

The churn management dataset is a dataset that has been cleaned and is often used for research on the problem of off-grid. Therefore, we used it to validate that our system and our

▼Table 4. Experimental results based on the churn management dataset

Algorithm	Recall	Precision	Accuracy	F1
Decision tree	0.7907	1.0	0.9913	0.8831
Logistic regression	0.9535	0.4409	0.9477	0.6029
Random forest	0.7442	0.8205	0.9825	0.7805
Parallel gcForest	0.8140	0.8333	0.9855	0.8235

▼Table 5. Experimental results based on the Telecom user dataset

Training data	Test data	Recall	Precision	Accuracy	F1
September	October	0.7895	0.6023	0.9233	0.6833
September	April	0.7186	0.6682	0.9173	0.6925
October	September	0.7147	0.6723	0.9197	0.6926
October	April	0.7239	0.5310	0.9331	0.6126
April	September	0.7545	0.5305	0.9330	0.6125
April	October	0.7844	0.6052	0.9238	0.6833

parallel gcForest are effective. In this experiment, we firstly made feature selection by using the methods in our system, involving correlation calculation and box plots analysis. According to the results of the analysis, we finally selected 12 features as follow: account length, area code, international plan, voice mail plan, number vmail messages, total day minutes, total day charge, total eve minutes, total international minutes, total international calls, total international charge, and number customer service calls. After feature selection, we use the decision tree, logistic regression, and random forest to deal with the prediction problem. In particular, the first three algorithms used selected features and some strategies to train model in the experiment; for example, we used threshold-moving to reduce the bad impact of class-imbalance. Besides, the parallel gcForest use all features of the dataset to do end-to-end training, thereby reducing the use of rules in training. The reason is that gcForest can automatically filter useless features. The final results are shown in Table 4. The result of the decision tree algorithm is the best, followed by the parallel gcForest. The result of logical regression is the worst. The experimental results show that our system and parallel gcForest are effective.

The Telecom user dataset is the raw data produced in real life in recent years. This dataset is very large and of poor quality. The role of our system is to use this data to solve the problem of off-grid prediction. This dataset is mainly to verify the practicality and effectiveness of our system. We selected three months of user data, including June 2016, September 2016, and April 2017. Moreover, the size of every month's data is millions. These data are read from the HDFS and we get a better result to predict whether the user will become an off-grid user through data pre-processing, feature selection, training model, and a series of operations. In the experiment, we use k-fold cross validation to select the final algorithm to train model. The advantage of the system is that it can choose a better model. As shown in Table 5, the best performance is achieved by logical regression.

The experiment results show that, in the data of academic research, the performance of our system is better than the performance in real-world data. The results are normal because the data in the actual application are dirty and the data scale is very large. Table 5 shows that our system could solve the problem of off-grid user prediction problem effectively. In addition, Table 4 shows that the proposed Parallel gcForest is effective.

6 Conclusions

In this paper, we develop a parallel off-grid prediction system, which includes many machine learning algorithms for data mining. Built on the Spark computing framework, the whole system can be applied to large data to solve the problem of off-grid user prediction and used for practical applications. Furthermore, we also use the Spark parallel framework to speed up the gcForest algorithm. The new parallel algorithm enables

gcForest to solve practical problems, such as off‑grid prediction problem and other problems of data mining.

References

[1] WARD J S, BARKER A. Undefined by Data: a Survey of Big Data Definitions [EB/OL]. (2013). https://arXiv preprint arXiv:1309.5821

[2] HAN L X, ONG H Y. Parallel Data Intensive Applications Using MapReduce: A Data Mining Case Study in Biomedical Sciences [J]. Cluster Computing, 2015, 18 (1): 403−418. DOI: 10.1007/s10586-014-0405-9

[3] LU P, DONG Z J, LUO S M, et al. A Parallel Platform for Web Text Mining [J]. ZTE Communications, 2013, 11(3): 56−61. DOI: 10.3969/j.issn.1673-5188.2013. 03.010

[4] PAGANO F, PARODI G, ZUNINO R. Parallel Implementation of Associative Memories for Image Classification [J]. Parallel Computing, 1993, 19(6): 667−684. DOI: 10.1016/0167-8191(93)90014-c

[5] CHU C-T, KIM S K, LIN Y-A, et al. Map-Reduce for Machine Learning on Multi-core [C]//19th International Conference on Neural Information Processing Systems. Vancouver, Canada, 2006: 281−288, 2007.

[6] DEAN J, GHEMAWAT S. Mapreduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1): 107−113, 2008.

[7] LEE K H, LEE Y, CHOI H, et al. Parallel Data Processing with MapReduce: a Survey [J]. ACM SIGMOD Record, 2012, 40(4): 11− 20. DOI: 10.1145/2094114.2094118

[8] KOLIOPOULOS A-K, YIAPANIS P, TEKINER F, et al. A Parallel Distributed Weka Framework for Big Data Mining Using Spark [C]//IEEE International Congress on Big Data. New York, USA, 2015. DOI 10.1109/BigDataCongress.2015.12

[9] COX L A. Data Mining and Causal Modeling of Customer Behaviors [J]. Telecommunication Systems, 2002, 21(2/3/4): 349−381. DOI: 10.1023/A:1020911018130

[10] ROSSET S, NEUMANN E. Integrating Customer Value Considerations into Predictive Modeling [C]//Third IEEE International Conference on Data Mining, Melbourne, USA, 2003: 283−290. DOI: 10.1109/ICDM.2003.1250931

[11] NATH S V, BEHARA R S. Customer Churn Analysis in the Wireless Industry: A Data Mining Approach [C]//Annual Meeting of the Decision Sciences Institute. China, 2003: 505-510

[12] SHVACHKO K, KUANG H R, RADIA S, et al. The Hadoop Distributed File System [C]//IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Incline Village, USA, 2010: 1−10. DOI: 10.1109/MSST.2010.5496972

[13] DEAN J, GHEMAWAT S. Mapreduce: Simplified Data Processing on Large Clusters [J]. Communications of the ACM, 2008, 51(1): 107−113. ACM, 2008

[14] MYERS J L, WELL A D. Research Design and Statistical Analysis [M]. 2nd ed. Mahwah, USA: Lawrence Erlbaum Associates, 2010

[15] BREIMAN L. Random Forests [J]. Machine Learning, 2001, 45(1): 5−32. DOI: 10.1023/A:1010933404324

[16] FENG J, ZHOU Z-H. Deep Forest: Towards an Alternative to Deep Neural Networks [C]//Twenty‑Sixth International Joint Conference on Artificial Intelligence. Melbourne, Australia, 2017: 3553−3559

[17] BA L J, CARUANA R. Do Deep Nets Really Need to be Deep? [C]//Advances in Neural Information Processing Systems. Red Hook, USA: Curran Associates, 2013: 2654−2662

[18] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks [C]//International Conference on Neural Information Processing Systems. Doha, Qatar, 2012: 1097−1105

[19] ZHOU Z H. Ensemble Methods: Foundations and Algorithms [M]. London, UK: Taylor & Francis, 2012

[20] LIU F T, KAI M T, ZHOU Z H. Isolation Forest [C]//Eighth IEEE International Conference on Data Mining. Miami, USA, 2009: 413−422

Biographies

LI Xuebing received the M.S. degree from the College of Information Science and Engineering, Yanshan University, China. His research interests include machine learning and data mining. He is currently a recommended system engineer at Baidu.

SUN Ying is a master candidate at the Institute of Computing Technology, Chinese Academy of Sciences, China. She received the B.S. degree from Beijing Institute of Technology, China in 2017. Her research interests include machine learning and data mining. She has published two research papers in SIGKDD.

ZHUANG Fuzhen (zhuangfuzhen@ict.ac.cn) received the B.S. degree in computer science from Chongqing University, China in 2006, and the Ph.D. degree in computer software and theory from the University of Chinese Academy of Sciences, China in 2011. He is currently an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences, China. His research interests include machine learning, data mining, transfer learning, multi‑task learning and recommendation systems. He has published around 60 papers in various journals and conferences, such as *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Cybernetics*, *IEEE Transactions on Neural Network and Learning System*, KDD, IJCAI, AAAI, ICDE, and WWW.

HE Jia is a Ph.D. candidate at the Institute of Computing Technology, Chinese Academy of Sciences, China. Her research interests include machine learning, Bayesian nonparametric learning, and multi‑view learning. She has published several papers in some relevant research conferences, such as IJCAI, ICDM, ECML, and CIKM.

ZHANG Zhao is a Ph.D. candidate in the Institute of Computing Technology, Chinese Academy of Sciences, China. He received the B.S. degree from Beijing Institute of Technology, China in 2015. His research interests include machine learning, data mining, and relational learning. He has published several papers in some relevant research conferences and journals, such as EMNLP, CIKM, and information systems.

ZHU Shijun received the B.E. degree in management science and engineering from University of Science and Technology of China (USTC) in 2003. Working with the Wireless Big Data R&D Center of ZTE Corporation, he is responsible for the development of smart optimization and planning sytem of wireless networks.

HE Qing is a professor at the Institute of Computing Technology, Chinese Academy of Science (CAS), and he is also a professor at University of Chinese Academy of Sciences, China. He received the B.S. degree from Hebei Normal University, China in 1985, and the M.S. degree from Zhengzhou University, China in 1987, both in mathematics. He received the Ph.D. degree in fuzzy mathematics and artificial intelligence in 2000 from Beijing Normal University, China. He was with Hebei University of Science and Technology from 1987 to 1997. He is currently a doctoral tutor at the Institute of Computing and Technology, CAS. His interests include data mining, machine learning, classification, and fuzzy clustering.

Detecting Abnormal Start-Ups, Unusual Resource Consumptions of the Smart Phone: A Deep Learning Approach

ZHENG Xiaoqing[1], LU Yaping[2], PENG Haoyuan[1], FENG Jiangtao[1], ZHOU Yi[1], JIANG Min[2], MA Li[2], ZHANG Ji[2], and JI Jie[2]

(1. School of Computer Science, Fudan University, Shanghai 201203, China;
2. Software R&D Center/Terminal Business Division, ZTE Corporation, Shanghai 201203, China)

Abstract: The temporal distance between events conveys information essential for many time series tasks such as speech recognition and rhythm detection. While traditional models such as hidden Markov models (HMMs) and discrete symbolic grammars tend to discard such information, recurrent neural networks (RNNs) can in principle learn to make use of it. As an advanced variant of RNNs, long short-term memory (LSTM) has an alternative (arguably better) mechanism for bridging long time lags. We propose a couple of deep neural network-based models to detect abnormal start-ups, unusual CPU and memory consumptions of the application processes running on smart phones. Experiment results showed that the proposed neural networks achieve remarkable performance at some reasonable computational cost. The speed advantage of neural networks makes them even more competitive for the applications requiring real-time response, offering the proposed models the potential for practical systems.

Keywords: deep learning; time series analysis; convolutional neural network; RNN

1 Introduction

Deep learning algorithm emerged as a successful machine learning technique a few years ago. With the deep architectures, it became possible to learn high-level (compact) representations, each of which combines features at lower levels in an exponential and hierarchical way [1]–[3]. A stack of representation layers, learned from the data in order to optimize the given objective, make deep neural networks gain advantages such as generalization to unknown examples [4], discovering disentangling factors of variation and sharing learned representations among multiple tasks [5]. The recent successes of the deep convolutional neural networks (CNNs) are mainly based on such ability to learn hierarchical representation for spatial data [6]. For modeling temporal data, the recent resurgence of recurrent neural networks (RNN) has led to remarkable advances [6]–[11]. Unlike the spatial data, learning both hierarchical and temporal representation is one of the long-standing challenges for RNNs in spite of the fact that hierarchical structures naturally exist in many temporal data [12]–[15].

Forecasting future values of the observed time series in fact plays an important role in nearly all fields of science and engineering, such as economics, finance, business intelligence, and industrial applications. There has been extensive research on using machine learning techniques for time-series forecasting. Several machine learning algorithms were presented to tackle time series forecasting problem, such as multilayer perceptron, Bayesian neural networks, K-nearest neighbor regression, support vector regression, and Gaussian processes [16]. The effectiveness of local learning techniques is explored for dealing with temporal data [17]. In this study, we tried to detect abnormal start-ups, unusual CPU and memory consumptions of the application processes running on smart phones using RNNs, which falls in line with the recent efforts to analyze time series data in order to extract meaningful statistics and other characteristics of the data with the deep learning approach.

The paper is organized as follows. First, we give an overview of the research goals, and then try to convey an intuition of the

Detecting Abnormal Start-ups, Unusual Resource Consumptions of the Smart Phone: A Deep Learning Approach | *Special Topic*

ZHENG Xiaoqing, LU Yaping, PENG Haoyuan, FENG Jiangtao, ZHOU Yi, JIANG Min, MA Li, ZHANG Ji, and JI Jie

key ideas in Section 2. Section 3 presents RNN-based models to detect unusual CPU, memory consumptions and abnormal start-ups of the application process running on the smart phones. Section 4 reports results of a number of experiments on real-life devices and shows the effectiveness of the proposed models. The conclusion and future work are summarized in Section 5.

2 Background

Recurrent neural networks are a class of artificial neural networks that possess internal state or short-term memory due to recurrent feed-back connections that make them suitable for dealing with sequential tasks, such as speech recognition, prediction and generation [18]–[20]. Traditional RNNs trained with stochastic gradient-descent (SGD) have difficulty learning long-term dependencies (i.e. spanning more than ten time-steps lag) encoded in the input sequences due to vanishing gradient [21]. This problem has been partly addressed by using a specially designed neuron structure, or cell, in long short-term memory (LSTM) networks [21], [22] that keeps constant backward flow in the error signal; second-order optimization methods [23] preserve the gradients by approximating their curvature; or using informed random initialization [24] which allows for training the networks with momentum and stochastic gradient-descent only.

In conventional LSTM each gate receives connections from the input units and the outputs of all cells, but there is no direct connection from the Constant Error Carrousel (CEC) it is supposed to control. All it can observe directly is the cell output, which is close to zero as long as the output gate is closed. The same problem occurs for multiple cells in a memory block: when the output gate is closed none of the gates has access to the CECs they control. The resulting lack of essential information may harm network performance. Gers, Schraudolph, and Schmidhuber [25] suggested adding weighted "peephole" connections from the CEC to the gates of the same memory block. The peephole connections allow all gates to inspect the current cell state even when the output gate is closed. The information can be essential for finding well-working network solutions. During learning no error signals are propagated back from gates via peephole connections to the CEC. Peephole connections are treated like regular connections to gates except for updating timing.

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks (a variant of LSTM), introduced by Cho et al. [9]. Their performance on polyphonic music modeling and speech signal modeling was found to be similar to that of long short-term memory. GRUs have been shown to exhibit better performance on smaller datasets because they have fewer parameters than LSTM, as they lack an output gate. There are several variations on the full gated units, with gating done using the previous hidden state and the bias in different combi-

nations.

3 Deep Neural Network-Based Models

Two deep neural network-based models will be described in this section. Both tasks mentioned above are time series forecasting, which uses a model to predict future values based on previously observed values. RNNs and their variants are used to perform those tasks since RNNs can recognize patterns that are defined by temporal distance.

3.1 Unusual CPU and Memory Consumption Detection

The proposed unusual CPU and memory consumption detection model aims at detecting unusual CPU and memory consumption of an application from its resource consumption series and runtime policy in a given time period. Model analyzes the resource consumption data of the application in the specified time period, including a series of its CPU consumption, its memory usage, as well as its runtime policy, and then outputs the probabilities of the existence of unusual CPU consumption and unusual memory consumption if there are any. Since it is a classification problem with time series analysis, we proposed a detection model based on LSTM network to model the time series, and followed by a 3-layer neural network to perform the classification.

3.1.1 Problem Formalization

The proposed model consists of two parts: unusual CPU consumption detection model and unusual memory consumption detection model. For unusual CPU consumption detection model, given an input time series $X = (X_1, X_2, \ldots X_t, \ldots, X_T)$, where X_t represents a sampling point including the current CPU consumption and runtime policy of an application and T is the length of the input series, the model analyzes the time series, predicts the probability of the existence of unusual resource consumption, and finally assigns the input into one of the two classes: unusual resource consumption or non-unusual resource consumption. Unusual memory consumption detection model has the same structure as the unusual CPU consumption detection model, but the sampling points X_t in its input series $X = (X_1, X_2, \ldots X_t, \ldots, X_T)$ only includes the current memory consumption data.

3.1.2 Normalization of Input Series

At each sampling point, the current CPU consumption is expressed as a percentage, ranging from 0 to 100, and the current memory consumption ranges from 0 to 10^6 KB. These values are all positive and too large for a neural network based model. Therefore, we introduced a normalization step before the proposed model processes the input data. We calculated the average and standard variance of the CPU consumption $(avg_{cpu}, stdv_{cpu})$ and the memory consumption $(avg_{mem}, stdv_{mem})$ of all the sampling points in the training set, and then normal-

Special Topic | Detecting Abnormal Start‑ups, Unusual Resource Consumptions of the Smart Phone: A Deep Learning Approach

ZHENG Xiaoqing, LU Yaping, PENG Haoyuan, FENG Jiangtao, ZHOU Yi, JIANG Min, MA Li, ZHANG Ji, and JI Jie

ized the resource consumption values in each sampling point by subtracting the average from the original value, and then dividing the result by the standard variance:

$$X_t[cpu] \leftarrow (X_t[cpu] - avg_{cpu})/stdv_{cpu}, \tag{1}$$

$$X_t[mem] \leftarrow (X_t[mem] - avg_{mem})/stdv_{mem}. \tag{2}$$

3.1.3 Time Series Modelling

For precisely detecting the unusual resource consumption, the amount of consumption is important, as well as the order and trends of changes in the input series. As **Fig. 1** shows, in order to leverage such information, we apply a bi‑directional LSTM network to model the input series and generate a vector representation for it, including a forward LSTM and a backward LSTM.

Both LSTM networks contain a cell state as the representation of its input series, which is initialized as a zero vector before it starts to read the input series (i.e., $C_0 = 0$). The forward LSTM reads the input series X from its left to its right. When it reads a sampling point X_t at timestep t, it first computes its forget gate to decide what information should be forgotten:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f), \tag{3}$$

where h_{t-1} is the output of the last timestep, and $\sigma(\cdot)$ represents the sigmoid function. Then, it computes its input gate to decide what information should be added into its current cell state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i). \tag{4}$$

Then, it updates its cell state and calculates its output at the current timestep:

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, X_t] + b_C), \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{6}$$

$$h_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) * tanh(C_t). \tag{7}$$

By continuously updating the cell state, after the LSTM reads all the sampling points, the cell state vector and the output vector at the last timestep contain the information of the whole input series. Notice that an LSTM only reads the input series from one direction, resulting that the earlier inputs usually have less impact on the final cell state and the output, we adopted a backward LSTM to read the input series from its right to its left, and concatenate the final output vector from the forward LSTM and backward LSTM as the vector representation of the input series:
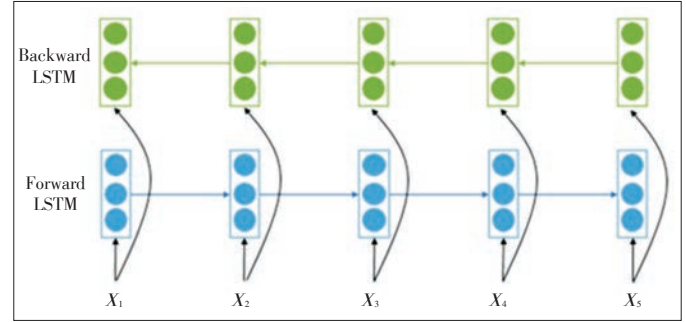
▲Figure 1. The bi-directional LSTM networks used in the model.

$$r = \left[h_T^{fw}, h_T^{bw} \right]. \tag{8}$$

We applied a 3‑layer neural network to compute the probability of the existence of unusual consumption. Preliminary experiments show that 3‑layer neural network achieved a good trade‑off between training speed and performance. The network takes the vector representation of the input series as input, and outputs a 2‑dimension vector $o = (o_1, o_2)$ by a softmax layer, where o_1 represents the probability of the existence of unusual consumption, and o_2 represents the probability of the normal case:

$$o = softmax(U \cdot \sigma(W \cdot r + b) + d). \tag{9}$$

We trained our model by minimizing the negative log‑likelihood of all the samples of the training set, and the model parameters are optimized by Adam [26], with hyper‑parameters recommended by the authors (i.e., learning rate = 0.001, β_1 = 0.9, β_2 = 0.999).

3.2 Abnormal Start-Up Prediction

To save the limited resource of a smart phone, a possible way is to stop application processes when they are abnormally started. By predicting whether a start-up of an application process is abnormal, memory and computational resources could be allocated more efficiently, which further optimizes the performance of the smart phone. The start‑up prediction problem can be viewed as a binary classification problem. We proposed a hybrid system, consisting of a rule‑based model and a deep learning-based model.

In this section, we first formalize the prediction problem and introduce a set of rules to label all start‑ups with either NORMAL or ABNORMAL tag by leveraging the full information contained in the given data set. Then a hybrid system is constructed to solve the prediction problem. Finally, the deep learning model is described in more details.

3.2.1 Problem Formalization

The abnormal start‑up prediction models aim at predicting the probability whether a start-up event or *am_proc_start* event

Detecting Abnormal Start-ups, Unusual Resource Consumptions of the Smart Phone: A Deep Learning Approach | *Special Topic*

ZHENG Xiaoqing, LU Yaping, PENG Haoyuan, FENG Jiangtao, ZHOU Yi, JIANG Min, MA Li, ZHANG Ji, and JI Jie

is abnormal given its previous logs $\{s_1, \ldots, s_i\}$, where each log s consists of an event name $e \in V_E$, a timestamp t and an argument list a. The used event names and their arguments are listed in **Table 1**. The vocabulary of the package and that of the component are denoted by V_P and V_C. The start type vocabulary is denoted by V_{ST}, in which *ACTIVITY* is a special type indicating that this start-up is brought up by an user. To sum it up, the log data set is defined by $D = \{(s_i, y_i) | s_i = \{s_{ij}\}, s_{ij} = (e_{ij}, t_{ij}, a_{ij})\}$.

However, in the raw data set, namely $D_{raw} = \{s_i\}$, the labels are not given. Thus before defining a machine learning model, the data should be labeled based on human knowledge. One possible way is to label those data manually, which takes a lot of human efforts as well as other resources. Alternatively, we proposed to use a set of rules based on human knowledge or engineer experience to label the data automatically, which is very cost-effective.

The data can be labeled with six steps or six rules from *R1* to *R6* (they are omitted here for security reason). Following those rules, the raw data are tagged with binary labels. The rules can be divided into two parts according to whether a rule can be used in the inference. The first part consists of rule {*R1*, *R2*, *R3*}, which infers the label with previous logs without any future information. The rule 4 labels the data with future information, while these data are not available in the usage. Thus the rule *R4* as well as the rules with lower priority form the second part. The first part can be tackled with trivial rules, namely rule {*R1*, *R2*, *R3*}, while the second part should be determined by a probabilistic machine learning model.

3.2.2 Hybrid Model

As shown in the previous section, the abnormal start-up prediction is defined as a binary classification problem, aiming at predicting whether a start-up of an application process is abnormal with a given set of logs. We proposed a hybrid model to solve this problem.

The model consists of two parts, a rule-based part and a deep learning-based part. The targeted start-up, as well as the previous logs, is first fed into the rule-based model, which is defined by the rule {*R1*, *R2*, *R3*}, and then generates one of the three possible results, namely *NORMAL*, *ABNORMAL*, and *UNDETERMINED*. The first two are deterministic ones, and will be directly outputted by the hybrid model. The last result indicates that the rule-based model is incapable of predicting the results, because it is determined by {*R4*, *R5*, *R6*}. Therefore, such *UNDETERMINED* data will be further fed into the deep learning model, and the deep learning model will output a

▼Table 1. Events and arguments

Event	Argument list
am_proc_start	Package; component; start type
am_proc_died	package

probability, reflecting the likelihood of whether the start-up is *NORMAL*.

3.2.3 Deep Learning Model

The deep learning model aims at providing the probability of a *NORMAL* start-up which cannot be determined by rules based on previous logs. In this situation, the deep learning model is designed to predict whether the process about to start-up will die in the future, namely in one second.

Unlike the rule-based model, the deep learning model cannot map a log to a predicted class directly. Because the deep learning model is capable of fitting data and logs without preprocessing, it could be full of noise, which in turn may result in a bad performance since noise could also be learned by the deep learning model. To reduce such noise in the input of the deep learning model, a feature extraction component is introduced to compute a set of features from the original logs by leveraging human priority. The necessity of these features are demonstrated by preliminary experiments. There are five features extracted listed in **Table 2**.

There are three layers in the neural networks. The first is the five parallel embedding layer, each of which transforms a respective feature into its dense vector representation. Then these vectors are combined with an addition operation, and this layer is designed to combine these parallel features into a joint feature. Finally, we use a logistic regression with this joint feature as its input to estimate the probability of whether the start-up is *NORMAL*. The training parameters are defined as all the embedding layers and the parameters of logistic regression. In this training process, the cross-entropy is used to compute the loss function. To learn the parameter of the deep learning model, Adam optimizer is used.

4 Experiments

We conducted experiments on both tasks to evaluate our model. In the following section, we will first describe the datasets we used and then show the performance of the proposed model on both tasks.

4.1 Data Sets and Preprocessing

For unusual CPU and memory consumption detection task, we prepared a dataset including 992 896 series of resource con-

▼Table 2. Features extracted from original logs to be inputted into the deep learning model

Features	Comments
Event	*am_proc_start* / *am_proc_died*
Package	which package to start
Component	which component to start
Start type	the startup mode
R5	whether *R5* is satisfied

Special Topic | Detecting Abnormal Start‑ups, Unusual Resource Consumptions of the Smart Phone: A Deep Learning Approach

ZHENG Xiaoqing, LU Yaping, PENG Haoyuan, FENG Jiangtao, ZHOU Yi, JIANG Min, MA Li, ZHANG Ji, and JI Jie

sumption in the training set, and 653 233 series in the testing set. The length of these series ranges from 1 to 12, and the average length is about 9.

Unusual CPU consumption occurs in 2.0% of the series in the training set, and unusual memory consumption occurs in 1.6% of the series in the training set. The proportion of the positive and negative samples are too large. In order to prevent our model from tilting, we used a sample strategy in the training process to ensure that the model learn an equal number of positive and negative samples. During testing, we normalized the input series with the average and standard variance calculated from the training set.

4.2 Unusual CPU and Memory Consumption Detection

We trained our model on an Nvidia GPU card. The final hyper-parameter configuration is listed in **Table 3**.

We evaluated the performance of our model from three perspectives: the precision, the recall, and the F1-score:

$$precision = \frac{Number\ of\ all\ detected\ unusual\ consumptions}{Number\ of\ all\ cases\ the\ model\ reports\ as\ unusual\ consumption}, \quad (10)$$

$$recall = \frac{Number\ of\ all\ detected\ unusual\ consumptions}{Number\ of\ unusual\ consumptions\ in\ the\ test\ set}, \quad (11)$$

$$F1 - score = \frac{2*precision*recall}{precision + recall}. \quad (12)$$

The performance results of the unusual CPU and memory consumption detection models are shown in **Table 4**.

Experimental results illustrate that our proposed model has achieved a high performance on the testing set (over 90% on all aspects), proving the effectiveness of applying LSTM networks into time series analysis problems in improving the performance of smart phones.

4.3 Abnormal Start-up Prediction

We collected logs generated by smart phones, and labelled

▼Table 3. Model hyper-parameter configuration

Hyper-parameter	Value
Size of the LSTM output	2 × 50
Size of the hidden layer in classifier	50
Learning rate	0.001
Batch size	32

▼Table 4. Performance of unusual CPU and memory consumption detection model

	CPU	Memory
Precision	0.963	0.972
Recall	0.983	0.998
F1-score	0.973	0.985

the logs with rules defined in Section 3.2.1. **Table 5** shows how many logs are determined by each rule. Please note that future information is involved in {$R4$, $R5$, $R6$}, and the logs labelled by these rules are related to the deep learning model. The labelled data set is split into a training set and a test one with a ratio of 4:1.

The deep learning model is implemented with Tensorflow. We set the embedding dimensionality of all features to 20 and the size of mini‑batch to 64. The hyper‑parameters for Adam optimizer were set to their default values. The results of our experiment is listed below in **Table 6**. It shows that our model demonstrated extremely well with high performance on the machine labelled data set.

5 Conclusions

In this paper, we have described a deep neural network‑based model for detecting abnormal application start-ups, and unusual CPU and memory consumptions of the application processes running on Android systems. A variant of recurrent neural network architecture with multiple layers was implemented and tested systematically. The experiment results showed that the proposed neural networks performed reasonably well on the two tasks, offering the potential of the proposed networks for practical time serious analyzing and other similar tasks. The number of parameters in neural network‑based models is usually much less than other competitors, such as conditional random fields (CRFs). Besides, only simple four-arithmetic operations are required to run the neural network‑based models after the models are well trained. So neural network‑based models are clearly run considerably faster and require much less memory than that of other models. The speed advantage of those neural networks makes them even more competitive for the applications requiring real‑time response, especially for the applications deployed on the smart phones.

▼Table 5. The statistics of the labelled logs

Rules	Number of logs labelled by each rule
$R1$	95 138
$R2$	269 348
$R3$	390 203
$R4$	19 557
$R5$	82 912
$R6$	207 881
$R4$, $R5$, $R6$	310 350
Total	1 065 039

▼Table 6. The performance of our models

Data set	Accuracy (%)
Deep learning model (for the part with future information involved)	98.93
Hybrid model	99.69

Detecting Abnormal Start‑ups, Unusual Resource Consumptions of the Smart Phone: A Deep Learning Approach | *Special Topic*

ZHENG Xiaoqing, LU Yaping, PENG Haoyuan, FENG Jiangtao, ZHOU Yi, JIANG Min, MA Li, ZHANG Ji, and JI Jie

References

[1] BENGIO Y. Learning Deep Architectures for AI [J]. Foundations and Trends in Machine Learning, 2009, 2(1): 1–127. DOI: 10.1561/2200000006

[2] LECUN Y, BENGIO Y, HINTON G. Deep Learning [J]. Nature, 2015, 521 (7553): 436–444. DOI: 10.1038/nature14539

[3] SCHMIDHUBER J. Deep Learning in Neural Networks: An Overview [J]. Neural Networks, 2015, 61: 85–117. DOI: 10.1016/j.neunet.2014.09.003

[4] HOFFMAN J, TZENG E, DONAHUE J, et al. One‑Shot Adaptation of Supervised Deep Convolutional Models [EB/OL]. (2013‑12‑21)[2018‑04‑15]. http://arxiv.org/abs/1312.6204

[5] KINGMA D P, WELLING M. Auto-Encoding Variational Bayes [EB/OL]. (2013-12-20)[2018-04-15]. https://arxiv.org/abs/1312.6114

[6] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks [J]. Communications of the ACM, 2017, 60 (6): 84–90. DOI: 10.1145/3065386

[7] MIKOLOV T, SUTSKEVER I, DEORAS A et al. Subword Language Modelling with Neural Networks [EB/OL]. (2012) [2018‑04‑15]. http://www.fit.vutbr.cz/~imikolov/rnnlm/char.pdf

[8] Graves A. Generating Sequences With Recurrent Neural Networks [EB/OL]. (2013-08-04) [2018-04-15]. https://arxiv.org/abs/1308.0850

[9] CHO K, MERRIENBOER B van, GULCEHRE C, et al. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation [C]// Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, 2014. DOI: 10.3115/v1/D14-1179

[10] SUTSKEVER I, VINYALS O L, Le Q V. Sequence to Sequence Learning with Neural Networks [M]//Advances in Neural Information Processing Systems. Cambridge, USA: The MIT Press, 2014: 3104–3112

[11] VINYALS O, TOSHEV A, BENGIO S, et al. Show and Tell: A Neural Image Caption Generator [C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA, 2015: 3156–3164. DOI: 10.1109/CVPR.2015.7298935

[12] MOZER M C. Induction of Multiscale Temporal Structure [C]//Proc. 4th International Conference on Neural Information Processing Systems. San Francisco, USA: Morgan Kaufmann Publishers Inc., 1991: 275–282.

[13] HIHI S E, BENGIO Y. Hierarchical Recurrent Neural Networks for Long-Term Dependencies [C]//Proc. 8th International Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, 1995: 493–499

[14] LIN T, HORNE B G, TINO P, et al. Learning Long‑Term Dependencies in NARX Recurrent Neural Networks [J]. IEEE Transactions on Neural Networks, 1996, 7(6): 1329–1338. DOI: 10.1109/72.548162

[15] KOUTNÍK J, GREFF K, GOMEZ F, et al. A Clockwork RNN [C]//31st International Conference on Machine Learning. Beijing, China, 2014: 1863–1871

[16] AHMED N K, ATIYA A F, GAYAR N E, et al. An Empirical Comparison of Machine Learning Models for Time Series Forecasting [J]. Econometric Reviews, 2010, 29(5/6): 594–621. DOI: 10.1080/07474938.2010.481556

[17] BONTEMPI G, BEN TAIEB S, LE BORGNE Y A. Machine Learning Strategies for Time Series Forecasting [M]//BONTEMPI G, BEN TAIEB S, LE BORGNE Y A. eds. Business Intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013: 62–77. DOI:10.1007/978-3-642-36318-4_3

[18] ROBINSON A J, FALLSIDE F. The Utility Driven Dynamic Error Propagation Network: CUED/FINFENG/TR.1 [R]. Cambridge, UK: Cambridge University, Engineering Department, 1987

[19] WERBOS P J. Generalization of Backpropagation with Application to a Recurrent Gas Market Model [J]. Neural Networks, 1988, 1(4): 339–356. DOI: 10.1016/0893-6080(88)90007-x

[20] Williams R J. Complexity of Exact Gradient Computation Algorithms for Recurrent Neural Networks: NUCCS-89-27 [R]. Boston USA: Northeastern University, College of Computer Science, 1989

[21] Hochreiter S, Bengio Y, Frasconi P, et al. Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies [M]// Kremer S C, Kolen, J F eds. A Field Guide to Dynamical Recurrent Networks. Hoboken, USA: IEEE Press, 2001

[22] HOCHREITER S, SCHMIDHUBER J. Long Short‑Term Memory [J]. Neural Computation, 1997, 9(8): 1735–1780. DOI: 10.1162/neco.1997.9.8.1735

[23] MARTENS J, SUTSKEVER I. Learning Recurrent Neural Networks with Hessian‑Free Optimization [C]//28th International Conference on Machine Learning. Bellevue, USA, 2011: 1033–1040

[24] SUTSKEVER I, MARTENS J, DAHL G E, et al. On the Importance of Initialization and Momentum in Deep Learning [C]//30th International Conference on Machine Learning. Atlanta, USA, 2013: 1139–1147

[25] GERS F A, SCHRAUDOLPH N N, SCHMIDHUBER J. Learning Precise Timing with LSTM Recurrent Networks [J]. Journal of Machine Learning Research, 2002, 3:115–143

[26] KINGMA D, Ba J. Adam: A Method for Stochastic Optimization [EB/OL]. (2014-12-22) [2018-04-15] https://arxiv.org/abs/1412.6980

Biographies

ZHENG Xiaoqing (zhengxq@fudan.edu.cn) received the Ph.D. degree in computer science from Zhejiang University, China in 2007. After that, he joined the faculty of School of Computer Science at Fudan University, China. He did research on semantic technology during his stay at the Information Technology Group, Massachusetts Institute of Technology (MIT), USA as an international faculty fellow from 2010 to 2011. His current research interests include natural language processing, deep learning, data analytics and semantic web. He published more than 30 academic papers in various journals and conferences, including ACL, IJCAI, AAAI, WWW, EMNLP, etc.

LU Yaping received his M.S. degree in industrial management engineering from Shanghai Jiao Tong University, China in 1988 and that in computer science from the University of Texas at Arlington, USA in 1996. He was the chief engineer of the North America R&D Center of ZTE's Terminal Business Division from 2015 to 2018. He worked as a senior staff engineer at Motorola Inc from 1997 to 2006, and a solution architect at i2 Technologies from 2006 to 2009 separately before he joined ZTE (USA) in 2010. His research interests include artificial intelligence, operations research, and their applications.

PENG Haoyuan received the M.S. degree in software engineering from Fudan University, China in 2015. He has been doing research on machine learning and natural language processing, supervised by Professor ZHENG Xiaoqing.

FENG Jiangtao received the M.S. degree in software engineering from Fudan University, China in 2019. He has been doing research on machine learning and natural language processing, supervised by Professor ZHENG Xiaoqing.

ZHOU Yi received the B.S. degree in software engineering from Fudan University, China in 2017. He has been doing research on machine learning and natural language processing, supervised by Professor ZHENG Xiaoqing.

JIANG Min was the system software engineer with the Software R&D Center of ZTE's Terminal Business Division. His research areas include Android system, embedded operating system, virtualization technology and machine learning.

MA Li received his B.S. degree from the University of Electronic Science and Technology, China in 1993. He has been working in ZTE Corporation since 1999, where he has held important software development, design and management positions in the Network Division, Technology Center and Mobile Device Division. He is currently working as the chief software engineer in the ZTE Terminal Business Division. His research interests and development field include terminal system design, AI engine deployment and optimization, algorithm R&D, and other core terminal technologies.

ZHANG Ji received the B.S. degrees in microelectronics and software engineering from Xidian University, China in 2013. He is currently a software engineer with ZTE Corporation. His research areas include Android system, embedded operating system, virtualization technology and machine learning.

JI Jie received his master's engineering degree in computer science software engineering from Xidian University, China in 2008. He is currently a software designer in the Research and Development Team of ZTE Corporation. His research interests include wireless communication technology, AI in embedded systems, and smart phone operating system.

Cooperative Intelligence for Autonomous Driving

CHENG Xiang[1], DUAN Dongliang[2], YANG Liuqing[3], and ZHENG Nanning[4]

(1. State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China;
2. Department of Electrical and Computer Engineering, University of Wyoming, Laramie, WY 82071, USA;
3. Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, USA;
4. Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China)

Abstract: Autonomous driving is an emerging technology attracting interests from various sectors in recent years. Most of existing work treats autonomous vehicles as isolated individuals and has focused on developing separate intelligent modules. In this paper, we attempt to exploit the connectivity among vehicles and propose a systematic framework to develop autonomous driving techniques. We first introduce a general hierarchical information fusion framework for cooperative sensing to obtain global situational awareness for vehicles. Following this, a cooperative intelligence framework is proposed for autonomous driving systems. This general framework can guide the development of data collection, sharing and processing strategies to realize different intelligent functions in autonomous driving.

Keywords: autonomous driving; cooperative intelligence; information fusion; vehicular communications and networking

1 Introduction

As an emerging technology attracting exponentially growing research and development interests from various sectors including academia, industry and government, autonomous driving is expected to bring numerous benefits to our everyday life, including increased safety, alleviation of traffic congestion, improved parking, and more efficient utilization of transportation resources, just to name a few (see e.g., [1]−[3]).

Though research on driverless vehicles dates back to as early as 1920s and the first prototype of autonomous vehicles dates back to 1980s, they have not attracted people's attention until a little more than 10 years ago due to the limitation of hardware techniques. However, In the past decade, there have been huge improvements on many supporting techniques such as sensing technology, high performance computing, artificial intelligence, computer vision, and wireless communications and network. These bring a promising future for driverless vehicles. The realization of driverless vehicles and their common adoption in people's daily life have been put on agenda. Nowadays, many research institutes and companies all over the world have already put the driverless vehicles to road tests, even on public roads. In United States, many states are giving permission to allow driverless vehicles to drive in their public transportation system and many companies such as Uber and Google are putting their driverless cars into operation.

Throughout the recent years of research and development in autonomous driving, however, the starting point has always been the cases with human drivers. More specifically, existing vehicle automation work has been almost exclusively focused on developing modules to assist human driving. These include auto‑parking, off‑lane drift warning, automatic emergency brake, and so on (see e.g. [4]−[6]). Based upon these work, intelligence has been gradually introduced into more and more driving functions, with the hope that one day, with integration of all these individual modules, the vehicle can be fully intelligent and can accomplish autonomous driving.

This strategy enables the developed technologies to be deployed into the current transportation systems and can bring investment return within a short time window. However, this strategy also greatly limits the framework of the autonomous vehicle research. This is because the inherited module-based approach, which was originally introduced to assist human driving, does not bear any systematic vision or plan on achieving the level of overall intelligence to take over human driving. Moreover, such a framework leads to the mentality of treating the autonomous vehicles as isolated individuals, which naturally inherits the human driving mechanism, but is a long way from maximally exploiting the machine intelligence or utilizing the potential of vehicle cooperation (see e.g. [7]−[9]). As a result, by imposing the human driving mentality onto machines, this widely adopted framework results in very high cost, while

achieving very limited reliability [10].

In this paper, instead of treating the autonomous vehicles as isolated individuals, we introduce interconnectivity and hence cooperation among vehicles for the system design. Based upon this capability, we investigate on how the redundancy provided by different types of sensors can be best utilized to improve the reliability in sensing and how the spatial diversity provided by different vehicles can be exploited to extend the vision range in sensing. The heterogeneous vehicular network structure is proposed to support the information sharing during the cooperation. A general hierarchical information fusion framework is then established for the cooperative sensing to achieve global situational awareness in autonomous driving. An example of cooperative simultaneous localization and mapping with multiple object tracking (SLAM‐MOT) is presented to illustrate the framework. Then, this sensing framework is extended to form a cooperative intelligence framework in designing the autonomous driving system. Last but not least, the issues associated with the communications system design are discussed and some important remarks are given.

## 2 Cooperative Sensing Obtains Global Situational Awareness

Sensing is the fundamental task in autonomous vehicles, which provides the necessary information for intelligent driving. Since the first prototype, many sensing techniques have been applied in autonomous vehicles. In existing autonomous vehicles, there are usually many sensing devices of many different types to execute different sensing tasks at different sensing ranges. To enable the intelligence, instead of providing the raw sensing data, the sensing module in autonomous vehicles extracts relevant sensing information via data analytics. So far, lots of efforts have been devoted towards better data analytics to improve the quality of information extracted from the sensing data. Accordingly, to improve the sensing performance, instead of installing more expensive sensing devices, people are now focusing more on designing better data analytical methods to improve the quality of sensing information while using low‐cost sensing devices.

In most existing work, since the starting point is to assist human drivers or to mimic the sensing capability of human drivers, in their design, an autonomous vehicle is treated as an isolated individual. Furthermore, a particular sensing task in a given sensing range is usually accomplished by a very limited number of sensors. As a result, if any of them is not working properly, the entire system does not have any backup for correction. For example, the notorious Tesla accident in Florida in May 7, 2016 occurred since the vehicle only relied on the camera and computer vision techniques to facilitate obstacle detection and unfortunately this technique failed to recognize the truck due to the lighting condition. This framework is somewhat like the situation that a single driver is driving the vehi-

cle and there is no other people to correct his/her conducts when needed. The only advantage of the autonomous vehicle over human driver is that the machine can never get fatigue and it has a lower error rate.

To improve the reliability of the entire system, recently there have been extensive research on extracting the sensing information from the data of multiple on‐board sensors from the same vehicle such as light detection and ranging (LiDAR), various kinds of cameras, radar with different wavelengths and detection ranges, and sonar (see e.g. [11]−[20]). However, due to cost considerations and space limit, the sensors that can be installed on a single vehicle are very limited. This means the data sources are still limited. More importantly, they are still on the same vehicle, and their sensing ranges are limited by the position and vision range of the vehicle and cannot avoid possible blind spots.

On the other hand, in the entire system, there are multiple autonomous vehicles, each equipped with their own sensors. If information can be cooperatively extracted from the sensing data collected from different locations, this can overcome the limitation of individual vehicle and achieve the beyond the‐vision-range awareness. In addition, in the intelligent transportation system, there are many road side units (RSUs) which are constantly monitoring the traffic conditions. If these information can be shared with the autonomous vehicles, further extension of the range of sensing can be achieved and hopefully the global situational awareness can be obtained. Different sensing strategies are illustrated in **Fig. 1**. In Fig. 1a, the white car


a) Single-vehicle sensing

b) Multi-vehicle sensing

c) Multi-vehicle sensing with road side units
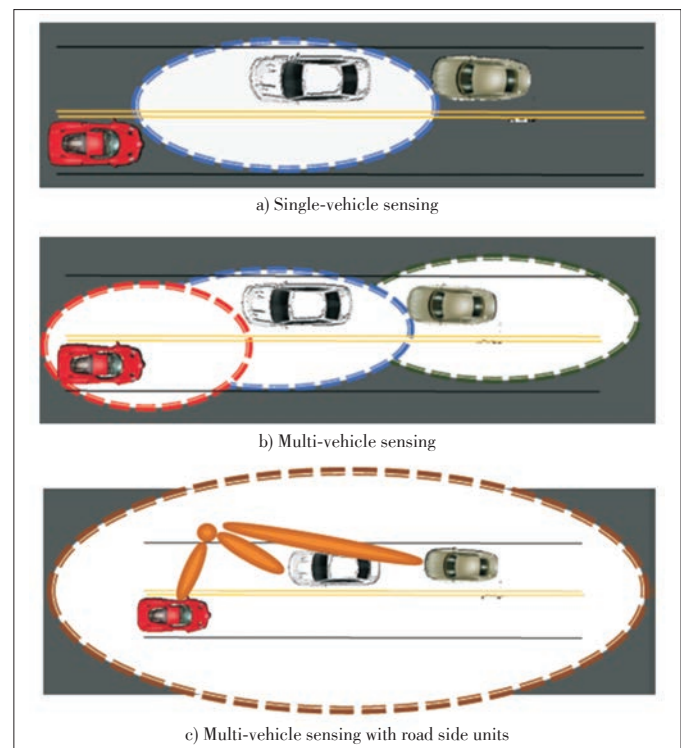
▲Figure 1. Illustration of different sensing strategies.

tries to sense the environment by itself. However, since it is located in a relatively crowded traffic scenario and its vision range is blocked by the red car and the green car, resulting in a very limited sensing range. When communications and information sharing are enabled among the three vehicles in the figure, as shown in Fig. 1b, the sensing range is greatly extended. If an RSU is also in the scene and shares its sensing information with all the vehicles, we can see that the sensing range would be further extended as seen in Fig. 1c. If information sharing among the entire intelligent system is enabled, the global situational awareness would be obtainable. From these illustrations, we see that with cooperative sensing, the troublemakers are now serving as helpers during sensing process.

3 Heterogeneous Vehicular Network Structure

To obtain the global situational awareness as illustrated above, sensing information across the entire system should be collected and shared via a supporting communications and networking infrastructure. We establish a global sensing and processing framework as illustrated in **Fig. 2**. There are mainly three types of entities in the framework:

• Autonomous vehicles are the major participants in the transportation system. They are using the road and changing the traffic conditions. At the same time, they would be able to sense the road and traffic conditions and other participants such as other vehicles, pedestrian, and animals.

• Intelligent transportation infrastructure refers to the auxiliary equipment installed in the transportation system such as road side units (see e.g. [21]–[23]). They usually would not participate in the transportation system and only collects information about the road and traffic conditions. Their original purpose is to report the basic traffic monitoring information to a control center to guide the traffic. However, within our framework, we allow them to share some information such as traffic flow condition and road congestion condition with the autonomous vehicles via vehicle-to-infrastructure (V2I) communications [24] to improve the situational awareness of the autonomous vehicles.

• Cloud processing center refers to a global center that collects all processed sensing information from the entities across
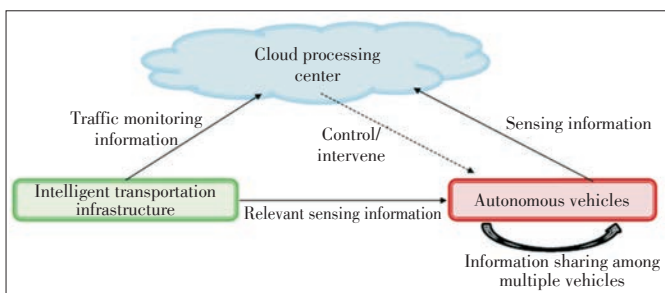
▲Figure 2. The information sharing among different entities.

the entire system and creates a global situational awareness of the traffic conditions in the system (see e.g. [25]). When needed, the center could send intervening control signals to some vehicles to guide their behavior for improved system efficiency or security. For example, if the center learned that a particular road is jammed, it can send control information to re-route some vehicles to avoid further traffics into that road and alleviate the jam condition.

With this proposed framework, the global situational awareness could be constructed by collecting, sharing and processing all sensing information across the entire system, including those provided by onboard sensors of autonomous vehicles and those from the intelligent transportation infrastructure. To support the information sharing within this framework, the network would be heterogeneous with different communication links meeting a wide range of QoS requirements:

• Among autonomous vehicles: To facilitate the cooperative sensing, a large amount of sensing data need to be shared among different autonomous vehicles. Depending on the time sensitivity of the sensing data, different requirements on the communication delay will be needed and depending on the volume of the sensing data, different bandwidth will be demanded. Generally speaking, high bandwidth and low latency communication techniques need to be exploited to support the vehicular network (see e.g. [26]–[31] for some proposed solutions).

• From the intelligent transportation infrastructure to the autonomous vehicles: The sensing information provided by the intelligent transportation units are usually at low rate. But they need to be communicated to the autonomous vehicles with low latency. In addition, the communication range of the intelligent transportation infrastructure is usually limited and the vehicles are usually moving at high speed. As a result, frequent hand-offs between a vehicle and different infrastructure units will be needed. This must be considered in the design of the network infrastructure of the entire system.

• Between autonomous vehicles and the cloud processing center: The communications between autonomous vehicles and the cloud processing center are bi-directional. For the uplink transmissions, the data volume is usually large, but the latency requirement would usually be low since the global road and traffic condition typically do not change that fast. Hence, the bandwidth would be of more concern; for the downlink transmissions, usually the more time sensitive control information with small volume will be involved. As a result, the latency would be the major concern.

• From the intelligent transportation infrastructure to the cloud processing center: This is already included in the current intelligent transportation system and can be directly applied to our proposed framework.

With the heterogeneous vehicular network to support the information sharing among different entities, we would be ready to develop the cooperative sensing and cooperative intelligence

framework.

4 A Hierarchical Information Fusion Framework for Cooperative Sensing

Given the possibility and many advantages of the collaborations among vehicles during sensing, the main challenge against collaborations among different sensors is the heterogeneity in data, even for the relatively simpler case of collaborating sensors located at the same vehicle. There are some papers in the literature concerning the latter case. However, they are all limited to some specific combinations of some specific sensors and the resultant algorithm cannot be applied to general cases (see e.g. [32]–[35]). When sensors' properties change, or as new sensors or sensor types are introduced, one has to completely re-formulate the fusion problem and re-solve it. Moreover, these algorithms cannot be applied to combine sensor information obtained from different vehicles. These issues are all due to the fact that the existing work tried to directly work on the heterogeneous data and there is no general framework to guide how the heterogeneous data should be collected, shared and processed to achieve the desired situational awareness.

To facilitate the design of cooperative sensing, here we first categorize the data based upon how the content of the data is related to the driving tasks:

- Data: It refers to the raw sensor data. Given that there are many different kinds of sensors in the system, the data take different forms and are reported at different rates. They also contain different information. In general, most raw sensor data cannot be directly used in the intelligent modules and have to be processed to provide driving-related guidance, e.g. the point cloud provided by LiDAR or the pictures captured by camera. In the context of cooperative sensing, sharing these types of data is also an unrealistic option due to their sizes.

- Information: It refers to the data describing the general driving environment that can be used in driving tasks. Examples would be the mapping for the area around the vehicle or identification of objects in the traffic scene. The "information" is usually a result of some preliminary processings of the raw sensor data.

- Knowledge: It refers to the data containing specific information that can readily guide the intelligent decisions in autonomous driving. Examples would be the driving status of other vehicles, the intentions of other objects in the traffic scene, the road condition along a particular route, and so on. One can treat the knowledge as the further processed information. The major difference between "information" and "knowledge" is that while the information can usually

be extracted from the raw data provided by a single sensor, the knowledge usually is the result of processing data from multiple sensors or even multiple vehicles.

It should be noted that though we are introducing these different categories, they are still simply the data during the fusion process and there are no clear distinction line among them. They are introduced here to describe the readiness of the data for intelligent driving tasks and also indicate the conciseness of data.

With this categorization of data, we propose a hierarchical information fusion procedure for cooperative sensing to obtain the global situation awareness in autonomous driving (**Fig. 3**). As seen in the figure, there are multiple levels of data processing and information fusion depending on the nature of data and how to obtain the required information/knowledge from the data. At the lowest level, data from different sensors are processed to extract information about the environment. Then, the extracted information can be shared among different vehicles to provide a more comprehensive knowledge about the driving situation.

4.1 An Illustrative Example: Cooperative SLAM-MOT

To better explain our proposed framework, here we use one important task of autonomous driving, simultaneous localization and mapping with moving objects tracking (SLAM-MOT) [36] as an illustrative example. The typical data used for localization would be the GPS and inertial measurement unit (IMU) data and the typical data for mapping and moving objects tracking are LiDAR, camera, sonar, radar, and so on. To achieve the best performance in cooperation, one would expect all data shared among all users. However, this creates huge communication burden, especially for those high-volume data such as cloud points from LiDAR and images from cameras. So, they have to be processed locally to some form of information before they can be shared among different users. However, at the same time, the vehicles are distributed in space and the vision ranges and views of the vehicles are quite different. This im-
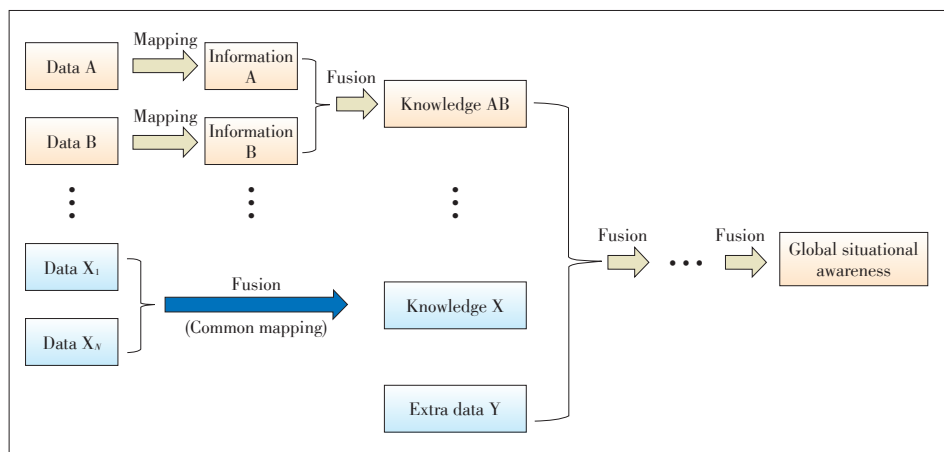
▲Figure 3. The hierarchical information fusion procedure to obtain the global situational awareness.

plies that there must be some schemes to handle the heterogeneity in space of these data. In addition, different autonomous vehicles might be equipped with different types of sensors. Therefore, there are also heterogeneity in information of these data. To implement the hierarchical information fusion framework we propose, one is facing the challenges introduced by the heterogeneity.

One possible solution to the data heterogeneity is to find a uniform representation of these data via local processing. For example, the local sensor data can be processed to generate the occupancy grid map (OGM) [37] for the area around a vehicle (e.g. our work in [38]). With the OGM representation, all forms of raw sensor data are converted to the uniform information about the occupancy states of map grids in space, which solves the problem of heterogeneity in information. At the same time, the location and vision range of different vehicles are reflected on the range that the OGM generated by each individual vehicle covers, which solves the issue of heterogeneity in space. Moreover, the quantified occupancy probabilities reflects the quality and confidence levels of the sensor data, which can guide the data fusion process. With these local processing and mapping, instead of the raw sensor data, the local maps can be shared among different vehicles to provide a global map, collecting the spatial diversity provided by multiple vehicles at different locations and achieving the beyond-the-vision-range situational awareness. This also greatly alleviate the challenges for the communication burden using our proposed fusion framework. Notice that one important feature of our proposed hierarchical information fusion framework is that one could flexibly adjust the fusion strategy based upon the data volume and the supporting communication network. Generally speaking, for high-volume data, they would be processed locally at lower levels to avoid high communication burden. On the other hand, when communication network is capable of supporting high-volume data, one can send more unprocessed raw data for less loss of information during the fusion process.

Then, the information can be further processed to obtain other information. For example, one can analyze the series of occupancy maps obtained over a period and then identify objects based upon the dynamics of the occupancy grids on the map. Large occupancy area corresponds to a large object such as cars or surrounding buildings, and small occupancy area corresponds to a small object such as motorcycles or pedestrians; fast-moving occupancy area corresponds to high-speed targets such as cars or motorcycles, and small-moving occupancy area corresponds to low-speed targets such as pedestrians or surrounding buildings. With this multiple levels of data processing, traffic information is roughly constructed. On the other hand, some data can directly provide the knowledge that is needed for driving. Examples would be the GPS data that provides the localization of the ego vehicle and the IMU data that provides the speed and acceleration of the ego vehicle. The self-awareness of these data can also then be combined with the ve-

hicles' observations on other objects using other sensors to provide the dynamic localization and tracking on other objects (e.g. our work in [39]). The SLAM-MOT system for the examples described above using our hierarchical information fusion framework is summarized in **Fig. 4**.

Then, to facilitate the driving optimization and decisions, one need to construct models to describe the dynamics of the traffic participants, including vehicles, pedestrians, animals, and so on. The behavior pattern of a single participant as well as the interactions among multiple participants in the system need to be considered in constructing the dynamic models. Based upon past historical data, models can be selected and model parameters can be estimated. Based upon the model and current observed data, future behavior or dynamics of the participants can then be predicted. To provide accurate prediction, the model must be composite and probabilistic. For being composite, multiple single models should be constructed to describe the possible dynamics of the participants working in different patterns (such as normal, abnormal, conservative, aggressive, fatigue drivers, and so on. For being probabilistic, probabilities should be assigned to the behavior patterns and also the intentions of the participants to cover all possibilities. In this way, intelligent decisions would then be developed. The intelligent decision process will in turn provide feedback to the information collection and fusion process to indicate what kind of information is needed at what level of quality. The intelligent framework is illustrated in **Fig. 5**.

5 A Cooperative Intelligence Framework for Autonomous Driving

With the global situational awareness provided by cooperative sensing, we can further include the collaborations among different vehicles in the decision process to obtain a coopera-
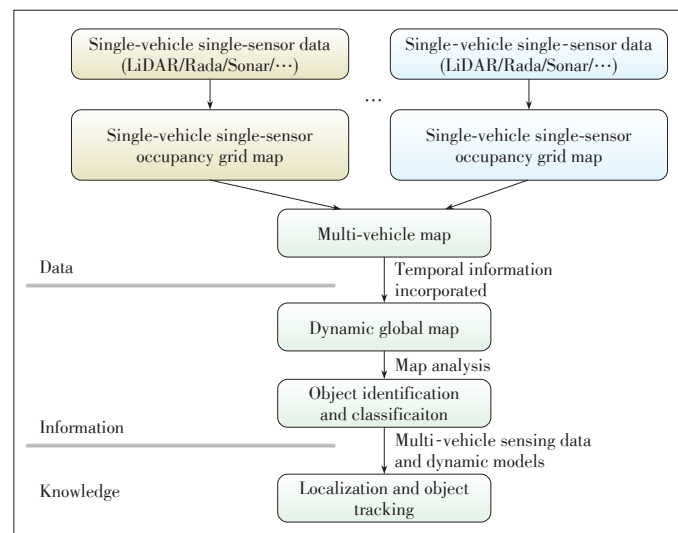
▲ Figure 4. Cooperative simultaneous localization and mapping with moving objects tracking (SLAM-MOT) using our proposed framework.
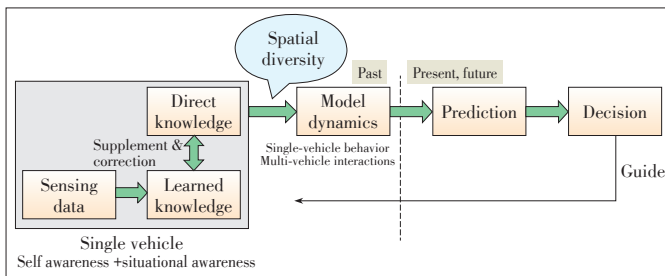
▲Figure 5. The illustration of our proposed cooperative intelligence framework.

tive intelligence framework. With this framework, we focus on how the information is collected, shared and involved during the intelligent driving process. The starting point of the information is the self-awareness, i.e. the driving decision or intention of the vehicle itself. For fully autonomous vehicles, this will be simply directly provided by the decision module; for half-autonomous vehicles where a driver is still controlling vehicle but the vehicle has the capability to observe and analyze the behavior of the driver, this can be provided by the vehicle's intelligent driver behavior analysis module. The self-awareness of multiple cooperating intelligent vehicles can be shared among each other. However, for other non-intelligent vehicles, cooperative sensing must be conducted to obtain the knowledge about them supplementing the self-awareness of the intelligent vehicles. With the proposed cooperative sensing framework, spatial diversity will be introduced to provide a more comprehensive picture on the traffic environment and overcome the vision ranges of individual vehicles. During this process, it is important to determine which kind of information or knowledge should be collected, processed, and shared among the vehicles.

6 Concluding Remarks and Prospects

In this paper, we exploit the collaborations among vehicles for autonomous driving. We proposed a hierarchical information fusion framework for cooperative sensing to provide global situational awareness for autonomous driving and a cooperative intelligence framework to encourage the collaborations among vehicles in their driving decision processes for improved system efficiency and security. The proposed frameworks are general and can provide valuable guidance to design the individual sensing and decision modules in autonomous driving.

One important issue for the proposed hierarchical information fusion and cooperative intelligence framework is the data sharing among different entities. This highly depends on the V2X communications and networking techniques. One could tackle this issue from two aspects: (1) the design and management of the information fusion procedure, considering the quality of service provided by the current vehicular communication infrastructure; and (2) the design of the vehicular communication infrastructure to satisfy the data sharing requirement in

the desired cooperative sensing and intelligence framework. On the one hand, the performance of the cooperative sensing and intelligence would be limited by the communications. On the other hand, the cooperative sensing and intelligence provides good motivation and guidance to the design of a better vehicular communications and networking structure.

References
[1] MAURER M, GERDES J C, LENZ B, et al. Autonomous Driving. Berlin, Heidelberg, Germany: Springer, 2016
[2] YANG J, COUGHLIN J F. In-Vehicle Technology for Self-Driving Cars: Advantages and Challenges for Aging Drivers [J]. International Journal of Automotive Technology, 2014, 15(2): 333−340. DOI: 10.1007/s12239-014-0034-6
[3] Bimbraw K. Autonomous Cars: Past, Present and Future a Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology [C]//12th International Conference on in Informatics in Control, Automation and Robotics (ICINCO). Colmar, France, 2015: 191−198
[4] GAIKWAD V, LOKHANDE S. Lane Departure Identification for Advanced Driver Assistance [J]. IEEE Transactions on Intelligent Transportation Systems, 2014: 1−9. DOI: 10.1109/tits.2014.2347400
[5] LIANG Y, REYES M L, LEE J D. Real-Time Detection of Driver Cognitive Distraction Using Support Vector Machines [J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(2): 340−350. DOI:10.1109/TITS.2007.895298
[6] CHENG S Y, TRIVEDI M M. Turn-Intent Analysis Using Body Pose for Intelligent Driver Assistance [J]. IEEE Pervasive Computing, 2006, 5(4): 28−37. DOI: 10.1109/MPRV.2006.88
[7] OKUDA R, KAJIWARA Y, TERASHIMA K. A Survey of Technical Trend of ADAS and Autonomous Driving [C]//2014 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA), Taiwan, China, 2014: 1−4. DOI: 10.1109/VLSI-TSA.2014.6839646
[8] LIN S-C, ZHANG Y, HSU C-H, et al. The Architectural Implications of Autonomous Driving: Constraints and Acceleration [C]//Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems. Williamsburg, USA, 2018: 751−766. DOI: 10.1145/3173162.3173191
[9] LIU S S, TANG J, ZHANG Z, et al. Computer Architectures for Autonomous Driving [J]. Computer, 2017, 50(8): 18−25. DOI: 10.1109/mc.2017.3001256
[10] DAZIANO R A, SARRIAS M, LEARD B. Are Consumers Willing to Pay to Let Cars Drive for Them? Analyzing Response to Autonomous Vehicles [J]. Transportation Research Part C: Emerging Technologies, 2017, 78: 150−164. DOI: 10.1016/j.trc.2017.03.003
[11] CHO H, SEO Y W, KUMAR B V K V, et al. A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments [C]// IEEE International Conference on Robotics and Automation (ICRA). Hong Kong, China, 2014: 1836−1843. DOI: 10.1109/ICRA.2014.6907100
[12] PONZ A, RODRÍGUEZ-GARAVITO C H, GARCÍA F, et al. Laser Scanner and Camera Fusion for Automatic Obstacle Classification in ADAS Application [M]//PONZ A, RODRÍGUEZ-GARAVITO C H, GARCÍA F, et al. eds. Communications in Computer and Information Science. Cham, Switzerland: Springer International Publishing, 2015: 237−249. DOI: 10.1007/978-3-319-27753-0_13
[13] ZIEBINSKI A, CUPEK R, ERDOGAN H, et al. A Survey of ADAS Technologies for the Future Perspective of Sensor Fusion [M]//ZIEBINSKI A, CUPEK R, ERDOGAN H, et al. eds. Computational Collective Intelligence. Cham, Switzerland: Springer International Publishing, 2016: 135−146. DOI: 10.1007/978-3-319-45246-3_13
[14] ASVADI A, PREMEBIDA C, PEIXOTO P, et al. 3D Lidar-Based Static and Moving Obstacle Detection in Driving Environments: An Approach Based on Voxels and Multi-Region Ground Planes [J]. Robotics and Autonomous Systems, 83: 299−311, 2016
[15] DE SILVA V, ROCHE J, KONDOZ A. Fusion of LiDar and Camera Sensor Data for Environment Sensing in Driverless Vehicles [EB/OL]. (2018-03-29)[2019-03-01]. https://arxiv.org/abs/1710.06230v2
[16] AUFRÈRE R, GOWDY J, MERTZ C, et al. Perception for Collision Avoidance and Autonomous Driving [J]. Mechatronics, 2003, 13(10): 1149−1161. DOI: 10.1016/s0957-4158(03)00047-3
[17] SHIMONI M, TOLT G, PERNEEL C, et al. Detection of Vehicles in Shadow Areas Using Combined Hyperspectral and Lidar Data[C]//IEEE International Geoscience and Remote Sensing Symposium. Vancouver, Canada, 2011: 4427−

4430. DOI: 10.1109/IGARSS.2011.6050214

[18] CHAVEZ-GARCIA R O, AYCARD O. Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking [J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(2): 525–534. DOI: 10.1109/tits.2015.2479925

[19] GAO H B, CHENG B, WANG J Q, et al. Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment [J]. IEEE Transactions on Industrial Informatics, 2018, 14(9): 4224–4231. DOI: 10.1109/tii.2018.2822828

[20] ASVADI A, GARROTE L, PREMEBIDA C, et al. Multimodal Vehicle Detection: Fusing 3D-LIDAR and Color Camera Data [J]. Pattern Recognition Letters, 2018, 115: 20–29. DOI: 10.1016/j.patrec.2017.09.038

[21] SOU S I, TONGUZ O K. Enhancing VANET Connectivity through Roadside Units on Highways [J]. IEEE Transactions on Vehicular Technology, 2011, 60 (8): 3586–3602. DOI: 10.1109/tvt.2011.2165739

[22] BARRACHINA J, GARRIDO P, FOGUE M, et al. Road Side Unit Deployment: A Density-Based Approach [J]. IEEE Intelligent Transportation Systems Magazine, 2013, 5(3): 30–39. DOI: 10.1109/mits.2013.2253159

[23] REIS A B, SARGENTO S, NEVES F, et al. Deploying Roadside Units in Sparse Vehicular Networks: What Really Works and What does not [J]. IEEE Transactions on Vehicular Technology, 2014, 63(6): 2794–2806. DOI: 10.1109/tvt.2013.2292519

[24] MILANES V, VILLAGRA J, GODOY J, et al. An Intelligent V2I-Based Traffic Management System [J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(1): 49–58. DOI: 10.1109/tits.2011.2178839

[25] WANG J, JIANG C, HAN Z. Internet of Vehicles: Sensing-Aided Transportation Information Collection and Diffusion [J]. IEEE Transactions on Vehicular Technology, 2018, 67(5): 3813–3825. DOI: 10.1109/tvt.2018.2796443

[26] CHEN S Z, HU J L, SHI Y, et al. Vehicle-To-Everything (v2x) Services Supported by LTE-Based Systems and 5G [J]. IEEE Communications Standards Magazine, 2017, 1(2): 70–76. DOI: 10.1109/mcomstd.2017.1700015

[27] ZHANG R, CHENG X, YAO Q, et al. Interference Graph-Based Resource-Sharing Schemes for Vehicular Networks [J]. IEEE Transactions on Vehicular Technology, 2013, 62(8): 4028–4039. DOI: 10.1109/TVT.2013.2245156

[28] CHENG X, YANG L, SHEN X. D2D for Intelligent Transportation Systems: A Feasibility Study [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(4): 1784–1793. DOI: 10.1109/tits.2014.2377074

[29] CHENG X, ZHANG R, YANG L. 5G-Enabled Vehicular Communications and Networking [M]. Cham, Switzerland: Springer International Publishing, 2018.

[30] CHENG X, CHEN C, ZHANG W X, et al. 5G-Enabled Cooperative Intelligent Vehicular (5GenCIV) Framework: When Benz Meets Marconi [J]. IEEE Intelligent Systems, 2017, 32(3): 53–59. DOI: 10.1109/mis.2017.53

[31] CHENG X, ZHANG R Q, YANG L Q. Wireless Toward the Era of Intelligent Vehicles [J]. IEEE Internet of Things Journal, 2019, 6(1): 188–202. DOI: 10.1109/jiot.2018.2884200

[32] KIM J K, KIM J W, KIM J H, et al. Experimental Studies of Autonomous Driving of a Vehicle on the Road Using LiDAR and DGPS[C]//15th International Conference on Control, Automation and Systems (ICCAS). Busan, South Korea, 2015: 1366–1369. DOI: 10.1109/ICCAS.2015.7364852

[33] JAIN A, SINGH A, KOPPULA H S, et al. Recurrent Neural Networks for Driver Activity Anticipation via Sensory-Fusion Architecture [C]//IEEE International Conference on Robotics and Automation (ICRA). Stockholm, Sweden, 2016: 3118–3125. DOI: 10.1109/ICRA.2016.7487478

[34] XUE J R, WANG D, DU S Y, et al. A Vision-Centered Multi-Sensor Fusing Approach to Self-Localization and Obstacle Perception for Robotic Cars [J]. Frontiers of Information Technology & Electronic Engineering, 2017, 18(1): 122–138. DOI: 10.1631/fitee.1601873

[35] XIAO L, WANG R L, DAI B, et al. Hybrid Conditional Random Field Based Camera-LIDAR Fusion for Road Detection [J]. Information Sciences, 2018, 432: 543–558. DOI: 10.1016/j.ins.2017.04.048

[36] WANG C C, THORPE C, THRUN S, et al. Simultaneous Localization, Mapping and Moving Object Tracking [J]. The International Journal of Robotics Research, 2007, 26(9): 889–916. DOI: 10.1177/0278364907081229

[37] BOUZOURAA M E, HOFMANN U. Fusion of Occupancy Grid Mapping and Model Based Object Tracking for Driver Assistance Systems Using Laser and Radar Sensors [C]//IEEE Intelligent Vehicles Symposium, San Diego, USA, 2010: 294–300. DOI: 10.1109/IVS.2010.5548106

[38] LI Y R, DUAN D L, CHEN C, et al. Occupancy Grid Map Formation and Fusion in Cooperative Autonomous Vehicle Sensing (Invited Paper) [C]//IEEE International Conference on Communication Systems (ICCS). Chengdu, China, 2018: 204–209. DOI: 10.1109/ICCS.2018.8689254

[39] YANG P T, DUAN D L, CHEN C, et al. Optimal Multi-Sensor Multi-Vehicle

(MSMV) Localization and Mobility Tracking [C]//IEEE Global Conference on Signal and Information Processing (GlobalSIP). Anaheim, USA, 2018: 1223–1227. DOI: 10.1109/GlobalSIP.2018.8646626

Biographies

CHENG Xiang (xiangcheng@pku.edu.cn) is currently a professor at Peking University, China. His general research interests are in areas of channel modeling, wireless communications and data analytics, subjects on which he has published more than 200 journal and conference papers, 5 books and 6 patents. Dr. CHENG was the recipient of the IEEE Asia Pacific (AP) Outstanding Young Researcher Award in 2015, the co-recipient for the 2016 IEEE JSAC Best Paper Award: Leonard G. Abraham Prize, the NSFC Outstanding Young Investigator Award, the both First-Rank and Second-Rank Award in Natural Science, Ministry of Education in China. He has also received the Best Paper Awards at IEEE ITST' 12, ICCC' 13, ITSC' 14, ICC' 16, ICNC' 17, GLOBECOM' 18, and ICCS' 18. He has served as Symposium Leading-Chair, Co-Chair, and a Member of the Technical Program Committee for several international conferences. He is currently an Associate Editor for IEEE Transactions on Intelligent Transportation Systems and Journal of Communications and Information Networks, and an IEEE Distinguished Lecturer. He is a senior member of IEEE.

DUAN Dongliang received the B.S. degree from Huazhong University of Science and Technology, China in 2006, the M.S. degree from the University of Florida, USA in 2009, and the Ph.D. degree from the Colorado State University, USA in 2012, all in electrical engineering. Since graduation, he joined the Department of Electrical and Computer Engineering at the University of Wyoming, USA and is now an associate professor. His research interests include signal processing techniques for wireless communications and power systems, estimation and detection theory, and energy resource management in wireless communication systems. He is a member of IEEE.

YANG Liuqing received the Ph.D. degree from the University of Minnesota, USA in 2004. Her main research interests include communications and signal processing. Dr. YANG has been actively serving in the technical community, including the organization of many IEEE international conferences, and on the editorial boards of a number of journals, including the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Intelligent Transportation Systems*, and the *IEEE Transactions on Signal Processing*. She received the Office of Naval Research Young Investigator Program Award in 2007, the National Science Foundation Career Award in 2009, the IEEE GLOBECOM Outstanding Service Award in 2010, the George T. Abell Outstanding Mid-Career Faculty Award and the Art Corey Outstanding International Contributions Award at CSU in 2012 and 2016 respectively, and Best Paper Awards at IEEE ICUWB' 06, ICCC' 13, ITSC' 14, GLOBECOM' 14, ICC' 16, WCSP' 16, GLOBECOM' 18 and ICCS' 18. She is a Fellow of IEEE.

ZHENG Nanning received his B.S. degree from the Department of Electrical Engineering, Xi' an Jiaotong University, China in 1975, and M.S. degree in information and control engineering from Xi' an Jiaotong University in 1981, and Ph.D. degree in electrical engineering from Keio University, Japan in 1985. In 1975, he jointed Xi' an Jiaotong University, where he is currently a professor and the Director of the Institute of Artificial Intelligence and Robotics. His research interests include computer vision, pattern recognition and image processing, and hardware implementation of intelligent systems. Prof. ZHENG became a member of the Chinese Academy of Engineering in 1999, and he is the Chinese Representative on the Governing Board of the International Association for Pattern Recognition. He also is the President of the Chinese Association of Automation. He is a Fellow of IEEE.

Standardization of Fieldbus and Industrial Ethernet

CHEN Jinghe and ZHANG Hesheng

(Beijing Jiaotong University, Beijing 100044, China)

Abstract: Fieldbus and industrial Ethernet standards can guide the specification and coordinate bus optimization. The standards are the basis for the development of fieldbus and industrial Ethernet. In this paper, we review complex standard systems all over the world. We discuss 18 fieldbus standards, including the International Electrotechnical Commission (IEC) 61158, the IEC 61784 standard matched with IEC 61158, the controller and device interface standard IEC 62026 for low voltage distribution and control devices, and the International Organization for Standardization (ISO) 11898 and ISO 11519 standards related to the controller area network (CAN) bus. We also introduce the standards of China, Europe, Japan and America. This paper provides a reference to develop fieldbus and industrial Ethernet products for Chinese enterprises.

Keywords: fieldbus; industrial Ethernet; standard

1 Introduction

Technical standards are a set of established norms about a technical system; they are usually developed and approved by recognized organizations. In today's digital technology era, a standard is more than a product specification; it also guides the development of the relevant high-tech field [1]. The standardization of fieldbus and industrial Ethernet is convenient for users to use products and regulates the market; it also promotes the development and optimization of bus technology.

The International Electrotechnical Commission (IEC) was committed to develop the one and only comprehensive international fieldbus standard and it began to formulate the fieldbus standards in 1985 [2]. However, the development of the fieldbus standards was not smoothly sailing due to the limitations of the industry, the history of geographical development, the driving of commercial interests, as well as the complex economic and social causes. After many years of debating and coordination, the first edition of the fieldbus standard IEC 61158 was published in March 1999 as a technical specification, following which the second edition with eight fieldbuses, the third edition with 10 fieldbuses, and the fourth edition with 18 fieldbuses were published in 2000, 2003 and 2007 respectively. After the fourth edition, new bus technology has no longer been added to IEC 61158; however, the revisions of the fourth edition are continued.

In recent years, Ethernet with unique advantages has gradually been applied into the field of industrial control. The Pub-

licly Available Specifications (PAS) documents based on 11 real-time Ethernets were published in the IEC 61784-2, and IEC 61784 was the "Communication Profile Family" matched with IEC 61158. Furthermore, IEC has developed the IEC 62026 that is control equipment interface standard for low voltage switchgears and control devices.

In addition to the IEC, the International Organization for Standardization (ISO), an international standard−setting body, has developed the standards for the controller area network (CAN) bus. They are ISO 11898 and ISO 11519.

Several countries, such as China, and regions have also developed fieldbus standards to promote the development of fieldbus technology.

At present, the fieldbus and industrial Ethernet are the mainstream technology in industrial control field, which are the key to intelligent manufacturing. Fieldbus and Industrial Ethernet standards are critical; standard formulation marks the fact that fieldbus and industrial Ethernet are moving towards harmonization and maturity and it is also directly related to the economic interests of major manufacturers. It is especially necessary for Chinese enterprises to understand and master the fieldbus and industrial Ethernet international standards and related regional and national standards, if they want to develop products with strong applicability and seize the larger world market.

This paper describes the international, regional and national standards of fieldbus and industrial Ethernet. It reviews the development process, content and standard types of the international standards IEC 61158, IEC 61784 and IEC 62026 in detail, the content of the international standards ISO 11898 and ISO 11519, and the types of Chinese standards corresponding to IEC62026 and IEC 61158. It briefly introduces European,

Japanese, American and other standards. Finally, the summary of the research significance of fieldbus and industrial Ethernet standards is presented.

2 International Fieldbus Standards

At present, recognized standardization organizations of fieldbus and industrial Ethernet are the IEC and ISO.

2.1 IEC Fieldbus Standards

2.1.1 IEC 61158

In 1985, IEC established the working group IEC/TC65/SC65C/WG6 for drafting fieldbus standards. The Digital Data Communications Subcommittee SC65C of the IEC and the Instrument Society of American (ISA) jointly developed fieldbus international standards called IEC 61158 for industrial control systems [3]. IEC 61158 has released four versions so far, and its long-time development and the number of voting reflect the diversity of the development of fieldbus and industrial Ethernet.

(1) The First Edition of IEC 61158:

The IEC began to develop fieldbus standards in 1985, but the progress had been slow due to inconsistent views from different countries. After years of debating, the first edition of IEC 61158 fieldbus standard (Ed1.0) was finally published as a technical specification in March 1999. The first edition is based on foundation fieldbus (FF). It consists of the Introductory Guide, Physical Layer Specification, Date Link Service Definition, Date Link Protocol Specification, Application Layer Specification, Application Layer Protocol Specification, and System Management.

(2) The Second Edition of IEC 61158:

After the release of IEC 61158 first edition, the major companies rushed to put forward all kinds of suggestions on amendments about the first edition of IEC 61158 specification in order to make their own fieldbuses into international standards. To coordinate, the IEC defined a matrix of protocol modules (called the type) [4], recognized by the parties. The second edition of IEC 61158 Standard (Ed2.0) was released in January 2000, including eight types of fieldbus. The types are shown in **Table 1** [5]. The Ed2.0 consists of the following five parts.

- IEC 61158-2: Fieldbus standard for use in industrial control systems—Part 2: Physical layer specification and service definition
- IEC 61158-3: Digital data communications for measurement and control—Fieldbus for use in industrial control systems—Part 3: Date link service definition
- IEC 61158-4: Digital data communications for measurement and control—Fieldbus for use in industrial control systems—Part 4: Date link protocol specification
- IEC 61158-5: Digital data communications for measurement and control—Fieldbus for use in industrial control sys-

tems—Part 5: Application layer service definition
- IEC 61158-6: Digital data communications for measurement and control—Fieldbus for use in industrial control systems—Part 6: Application layer protocol specification.

The new IEC 61158 retained the original IEC technical report as Type 1, and the other buses entered IEC 61158 as a Type 2-Type 8 in accordance with the IEC Technical Report format [6].

(3) The Third Edition of IEC 61158

After the release of IEC 61158 second Edition (Ed2.0), fieldbus and industrial Ethernet technology had developed rapidly. In April 2003, the third Edition (Ed3.0) was officially released. **Table 2** shows the types of IEC 61158 Ed3.0. The title of the third edition of IEC 61158-2 was amended as "IEC 61158-2: Digital Data Communication Systems for Measurement and Control—Fieldbus for Industrial Control Systems—Part 2: Physical Layer Specification" [7]. The titles of IEC 61158-3 to IEC 61158-6 in IEC 61158 version 3 were the same as the corresponding parts of version 2.

In particular, IEC 61158 Ed3.0 was amended by MT9. The MT9 Maintenance group develops amendments to IEC 61158 [8].

▼Table 1. The types of IEC 61158 Ed2.0

Type	Technical name
Type 1	Foundation fieldbus
Type 2	ControlNet
Type 3	ProfiBus
Type 4	P-Net
Type 5	HSE
Type 6	SwiftNet
Type 7	WorldFIP
Type 8	Interbus

HSE: High Speed Ethernet   WorldFIP: World Factory Instrument Protocol
P-Net: Process automation net

▼Table 2. The types of IEC 61158 Ed3.0

Type	Technical name
Type 1	Foundation fieldbus
Type 2	ControlNet
Type 3	ProfiBus
Type 4	P-Net
Type 5	HSE
Type 6	SwiftNet
Type 7	WorldFIP
Type 8	Interbus
Type 9	FF H1
Type 10	ProfiNet

FF: Foundation Fieldbus   P-Net: Process automation net
HSE: High Speed Ethernet   WorldFIP: World Factory Instrument Protocol

(4) The Fourth Edition of IEC 61158

The Fieldbus Maintenance Working Group IEC/SC65C/MT9 and the Realtime Ethernet Working Group WG11 held a joint working group meeting at Phoenix, USA, from December 5 to 9, 2005. A total of 32 technical experts from the world's major industrial companies and bus organizations and related IEC officials attended the meeting. They worked together for drafting IEC 61158 fieldbus (Ed4.0) and IEC61784-2 real-time Ethernet Committee Draft with Vote (CDV) [9].

IEC 61158 Ed4.0 consists of the following six parts under the title "Industrial Communication Networks—Fieldbus Specifications":

- IEC 61158-1: Overview and guidance for the IEC 61158 and IEC 61784 series
- IEC 61158-2: Physical layer specification and service definition
- IEC 61158-3: Data link layer service definition
- IEC 61158-4: Data link layer protocol specification
- IEC 61158-5: Application layer service definition
- IEC 61158-6: Application layer protocol specification.

The 20 fieldbus types of the IEC 61158 Ed4.0 are shown in **Table 3**.

The fieldbus and industrial Ethernet standards basically keep stable after the release of IEC 61158 Ed4.0, and their development and application are onto the track of continuous development.

From the viewpoint of the development of IEC 61158, the IEC eventually have to use a variety of fieldbuses as a standard to meet the requirements of different parties, although its original intention was to develop a standard for a single fieldbus. There are two main reasons. One is the technical reason. There is no best fieldbus that could be used for all the application areas yet; actually, each fieldbus had its own application scope. The other reason is the commercial interest. The bright development prospects of fieldbus have attracted major companies, especially large-scale multinational companies, to invest much in developing their own fieldbus products without taking the related established standards into account. These major companies and bus organizations would then push their own fieldbuses to become the international standards for the purpose of protecting their own investment and interests. They bargained and competed with one another, eventually leading to a variety of buses in IEC 61158.

2.1.2 IEC 61784

The series of IEC 61158 standards are conceptual specifications that do not involve the specific implementation of fieldbus. And there is only fieldbus type number in the IEC 61158 standard, and the specific fieldbus technical name and commercial trade name are not allowed to appear. So IEC/SC65C developed IEC 61784 matched with the IEC 61158 standard, in order to make developers and users easily carry out product design and application selection.

IEC 61784 is the Communication Profile Family (CPF) in continuous and decentralized manufacturing systems related to fieldbus in industrial control systems. IEC 61784 describes a subset of the communication used by a particular fieldbus system. The communication profiles of the different fieldbuses and the bus type of IEC 61158 corresponding to them are shown in this standard. The IEC 61784 standard under the title "Industrial Communication Networks—Profiles" consists of the following parts [10]:

- IEC 61784-1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems
- IEC 61784-2: Additional profiles for ISO/IEC 8802.3 based communication networks in real-time applications
- IEC 61784-3: Profiles for functional safety communications in industrial networks
- IEC 61784-4: Profiles for secure communications in industrial networks
- IEC 61784-5: Installation profiles for communication networks in industrial control systems
- IEC 61784-6: Time sensitive networking profile for industri-

▼Table 3. The types of IEC 61158 Ed4.0

Type	Technical name
Type 1	Foundation fieldbus
Type 2	CIP
Type 3	ProfiBus
Type 4	P-Net
Type 5	HSE
Type 6	SwiftNet (be revoked)
Type 7	WorldFIP
Type 8	Interbus
Type 9	FF H1
Type 10	ProfiNet
Type 11	TCnet
Type 12	EtherCAT
Type 13	Ethernet Powerlink
Type 14	EPA
Type 15	Modbus-RTPS
Type 16	SERCOS I, SERCOS II
Type 17	Vnet/IP
Type 18	CC-Link
Type 19	SERCOS III
Type 20	HART

CC-Link: Control & Communication Link
CIP: Common Industrial Protocol
EPA: Ethernet for Plant Automation
EtherCAT: Ethernet for Control Automation Technology
FF: Foundation Fieldbus
HART: Highway Addressable Remote Transducer

HSE: High Speed Ethernet
P-Net: Process automation net
RTPS: Real-time Publish/Subscribe
SERCOS: Serial Real Time Communication Specification
TCnet: Time-Critical Control network
WorldFIP: World Factory Instrument Protocol

al use based on IEEE 802.1 and IEEE 802.3 (The document has not been officially published).

CPF for fieldbuses in IEC 61784-1 [11] are shown in **Table 4** and real-time Ethernet CPF in IEC 61784-2 [10], [11] are shown in **Table 5**.

The above communication profiles provide a detailed description of the interoperability features and options in fieldbus devices, and specify the capabilities and detailed communication function of equipment in the network communication, and give the specific performance indicators of equipment communications. IEC61784-2 communication profiles specify delivery time, number of end nodes, basic network topology, number of switches between end nodes, throughput RTE, RTE broadband, time synchronization accuracy, and redundancy re-

▼Table 4. CPF for fieldbuses in IEC 61784-1

Family	Technology name	The corresponding type in IEC 61158
CPF 1	Foundation fieldbus	Type 1, 9
CPF 2	CIP	Type 2
CPF 3	ProfiBus	Type 3
CPF 4	P-Net	Type 4
CPF 5	WorldFIP	Type 7
CPF 6	Interbus	Type 8
CPF 8	CC-Link	Type 18
CPF 9	HART	Type 20
CPF 16	SERCOS I, SERCOS II	Type 16

CC-Link: Control & Communication Link
CIP: Common Industrial Protocol
FIP: Factory Instrument Protocol
HART: Highway Addressable Remote Transducer
IEC: International Electrotechnical Commission
P-Net: Process automation net
SERCOS: Serial Real Time Communication Specification

▼Table 5. CPF for real-time Ethernet in IEC 61784-2

Family	Technology name	IEC/PAS	The corresponding type in IEC 61158
CPF 2	EtherNet/IP	IEC/PAS62413	Type 5
CPF 3	Profinet	IEC/PAS62411	Type 10
CPF 4	P-Net	IEC/PAS62412	Type 4
CPF 6	Interbus	—	Type 8
CPF 10	VNET/IP	IEC/PAS62405	Type 17
CPF 11	TCnet	IEC/PAS62406	Type 11
CPF 12	EtherCAT	IEC/PAS62407	Type 12
CPF 13	Ethernet powerlink	IEC/PAS62408	Type 13
CPF 14	EPA	IEC/PAS62409	Type 14
CPF 15	Modbus-RTPS	IEC/PAS62030	Type 15
CPF 16	SERCOS III	IEC/PAS62410	Type 19

EPA: Ethernet for Plant Automation
EtherCAT: Ethernet for Control Automation Technology
IEC: International Electrotechnical Commission
IP: Industry Protocol
PAS: Publicly Available Specification
P-Net: Process automation net
RTPS: Real-time Publish/Subscribe
SERCOS: Serial Real Time Communication Specification
TCnet: Time-Critical Control network

covery time of real-time Ethernet [12]. These profiles help to correctly describe the consistency of ISO/IEC 8802.3 of the Real-Time Ethernet (RTE) communication network and avoid bias to prevent its understanding and use.

2.1.3 IEC 62026

IEC 62026 standards (**Table 6**) developed by IEC/TC17/SC17B/WG3 relate to the fieldbus for low-voltage switchgears and controlgear.

Among these standards, Actuator sensor interface (AS-I) was launched by the German Pepperl and Fuchs; DeviceNet and Smart distributed system (SDS) were respectively launched by the original AB company (Rockwell Automation Company now) and Honeywell in America.

The main contents of the original IEC 62026 are as follows [13]:
- Part 1: General rules. They specify some contents of the special interface standards and some common requirements
- Part 2: AS-I. AS-I is originally the European standard (EN50295), which is particularly suitable for connecting sensors and actuators with switching characteristics
- Part 3: DeviceNet. It is an open communication network based on CAN and it is easy to connect the low voltage switch device to the main control device through DeviceNet. In addition, DeviceNet has become one of European standards EN50325
- Part 4: Lontalk. It is not suitable as a general agreement of device layer and removed later
- Part 5: SDS. It is a CAN-based control network technology introduced in 1993. This network can connect low-voltage electrical appliances and the main control device through a trunk line to achieve data exchange, processing and transmission in automation systems. SDS has also become one of European standards EN50325
- Part 6: Serial multiplex control bus (SMCB). It is used for connection with switching devices, control equipment, sensors, switches, etc.

With the development of technology and the choice of the market, IEC deleted IEC62026-4 and IEC62026-6 in June

▼Table 6. IEC 62026 standards

Standard number	Standard title	Published date	Status
IEC62026-1	Part 1: General rules	2000-07	In force
IEC62026-2	Part 2: Actuator sensor interface (AS-I)	2000-07	In force
IEC62026-3	Part 3: DeviceNet	2000-07	In force
IEC62026-4	Part 4: LonTalk	2000-07	Cancelled
IEC62026-5	Part 5: Smart distributed system (SDS)	2000-07	Cancelled
IEC62026-6	Part 6: Serial multiplex control bus (SMCB)	2001-11	Cancelled
IEC62026-7	Part 7: CompoNet	2010-12	In force

2000, IEC62026 - 5 in 2006; updated IEC62026 - 2and IEC62026-3 in June 2007; and added IEC62026-7 in December 2010.

2.2 ISO Fieldbus Standard

CAN is one of the earliest international fieldbus standards before IEC 61158 and IEC 62026 in the fieldbus field. CAN is approved by Technical Committee ISO/TC22 as international standards for ISO 11898 (communication rate < 1 Mbit/s) and ISO 11519 (communication rate < l25 kbit/s).

2.2.1 ISO 11898

IS0 11898 was a controller area network for high-speed communication developed by Technical Committee ISO/TC22 - Road Vehicles and its subcommittee SC 31-Data Communication in November 1993.

ISO 11898 conforms to the Open System Interconnection (OSI) reference model specified in ISO 7489, and describes the general structure of the CAN in a hierarchical form, including the detailed technical specifications of the CAN physical layer and the data link layer. It specifies the various characteristics of digital information exchange between the electronic control units of road vehicles equipped with CAN at a transmission rate of 125 kb/s to 1 Mb/s.

ISO 11898 consists of the following parts, under the general title "Road Vehicles—Controller area network (CAN)" [14]:
- ISO 11898-1: Data link layer and physical signaling
- ISO 11898-2: High-speed medium access unit
- ISO 11898-3: Low-speed, fault-tolerant, medium-dependent interface
- ISO 11898-4: Time-triggered communication
- ISO 11898-5: High-speed medium access unit with low-power mode
- ISO 11898-6: High-speed medium access unit with selective wake-up functionality.

Specially, CAN Technical Specification Standard 2.0 was released in 1991, included 2.0A and 2.0B. 2.0A describes the CAN message format defined in the CAN specification version 1.2; 2.0B describes both standard and extended message formats.

2.2.2 ISO 11519

The Technical Committee ISO/TC22-Road Vehicles and its subcommittee SC 31-Data Communication also developed ISO 11519, a controller area network for low speed serial data communication. The contents of ISO 11519 include Part 1: General and Definitions; Part 2: Low Speed Controller Area Network (CAN); Part 3: Vehicle Area Network (VAN) [15].

ISO 11519-1 (Part 1) describes the general definition of low-speed serial data communication for road vehicles at a rate of no more than 125 kb/s, stipulates the general structure of the communication network for information transmission between different types of electronic modules on road vehicles and the main contents of the data link layer and the physical layer. ISO 11519-2, 3 (Parts 2 and 3) respectively specify the data link layer and physical layer of the CAN and VAN (communication networks up to 125 kb/s) for road vehicle application and illustrate the general structure of the networks.

In particular, CAN is the physical layer of the IEC 62026-3 DeviceNet and IEC 62026-5 Smart Distribution System, so it is the most important technical basis for IEC 62026.

3 Chinese Fieldbus Standard

The basic principle of standardization of relevant agreements in China is to adopt IEC standards equivalently, and the research of Chinese fieldbus standards corresponding to IEC 62026 and IEC 61158 has achieved some success.

3.1 Chinese Fieldbus Standard Corresponding to IEC 62026

The third meeting of the second National Low Voltage Electrical Apparatus Standardization Technical Committee of China was held in Qingdao, China, September 2001. The fieldbus standard GB/T 18858 corresponding to IEC 62026 was reviewed and adopted at this meeting [16]. It consists of the following three sections:
- GB/T 18858.1—2002: Low-voltage switchgear and controlgear—Device interface (CDI)—Part 1:General rules
- GB/T 18858.2—2002: Low-voltage switchgear and controlgear—Device interface (CDI)—Part 2: Actuator sensor (AS-I)
- GB/T 18858.3—2002: Low-voltage switchgear and controlgear—Device interface (CDI)—Part 3: DeviceNet.

With the development of technology and improvement of IEC 62026, the National Standard Committee of China approved to release the revised GB/T18858.1, GB/T18858.2, GB/T18858.3 in November 2012, followed by the release of the GB/T 18858.7 in June 2014. The revised China Fieldbus Standard GB/T 18858 includes the following sections:
- GB/T 18858.1—2012: Low-voltage switchgear and controlgear—Device interface (CDI)—Part 1:General rules
- GB/T 18858.2—2012: Low-voltage switchgear and controlgear—Device interface (CDI)—Part 2: Actuator sensor (AS-I)
- GB/T 18858.3—2012: Low-voltage switchgear and controlgear—Device interface (CDI)—Part 3: DeviceNet
- GB/T 18858.7—2014: Low-voltage switchgear and controlgear—Device interface (CDI)—Part 7: CompoNet.

3.2 Chinese Fieldbus Standard Corresponding to IEC 61158

3.2.1 Recommended National Standards

In China, recommended standards are a type of standards that is voluntarily adopted by means of economic adjustment in

the aspects of production, exchange, use, and so on, also known as voluntary or non-mandatory standards. Any unit has the right to decide whether or not to adopt these standard, and does not bear economic or legal responsibility in violation of such standards. However, once a recommended standard is accepted and adopted, or agreed to be included in an economic contract, the parties have to comply with the technical basis, with legal binding.

Table 7 shows the recommended national fieldbus standards.

3.2.2 Guidance National Standards

The guidance standard refers to the national standard that is voluntarily adopted by the organization (enterprise) in the aspects of production, exchange, use and so on. It is not compulsory and does not have the legal restriction, but only the technical basis for the related parties. **Table 8** shows the guidance national fieldbus standards in China.

As can be seen from the above standards, China has not copied the IEC 61158 international standard. There are two reasons. First, IEC 61158 standards issued separately according to the general, physical layer service definition, protocol specification, data link layer service definition and data link Layer protocol specification, etc. It is very inconvenient to choose from more than 4000 pages of IEC 61158 with 18 kinds of standards when users need to use a standard. Second, performance, application scope, development difficulty and the level of openness vary in 18 kinds of fieldbuses, but generally, we only select buses with the best performance in these buses as

▼Table 7. The recommended national fieldbus standards

Name	National standard number	The title of standard
EPA	GB/T 20171—2006	System structure and communication specification of EPA for industrial measurement and control systems
ProfiBus	GB/T 20540—2006	Fieldbus for measurement and control—digital data communication-Industrial control systems type 3: ProfiBus specification
Modbus	GB/T 19582—2008	Industrial automation network specification based on Modbus Protocol
CC-Link	GB/T 19760—2008	CC-Link control and communication network specification
HART	GB/T 29910—2013	Industrial communication network fieldbus specification type 20: HART
EtherCAT	GB/T 31230—2014	Industrial Ethernet EtherCAT
Profinet-IO	GB/T 25105—2014	Industrial communication network fieldbus specification type 10: Profinet-IO
Profisafe	GB/T 20830—2015	Functional security communication profiles based on ProfiBus-DP and ProfiBus-IO—Profisafe
CC-Link IE	GB/T 33537—2017	Industrial communication network fieldbus specification type 23:CC-Link IE

CC-Link: Control & Communication Link   HART: Highway Addressable Remote
DP: Decentralized Periphery   Transducer
EPA: Ethernet for Plant Automation   IE: Industrial Ethernet
EtherCAT: Ethernet for Control Automation Technology   IO: Input Output

▼Table 8. The guidance national fieldbus standards

Name	National standard number	The title of standard
LonWorks	GB/Z 20177—2006	Control network LonWorks technical specification
Profinet	GB/Z 20541—2006	Fieldbus for measurement and control -digital data communication-Industrial control systems type 10: Profinet
ControlNet, EtherNet/IP	GB/Z 26157—2010	Fieldbus for measurement and control -digital data communication-Industrial control systems type 2:ControlNet and EtherNet/IP
Interbus	GB/Z 29619—2013	Fieldbus for measurement and control -digital data communication-Industrial control systems type 8:Interbus
CC-Link safety	GB/Z 29496—2013	Control and communication network CC-Link safety specification

CC: Control & Communication   IP: Industry Protocol

national standards instead of using all of them. The methods of selecting Chinese national standards include:

(1) Some excellent and promising buses with great influence are chosen as the standard of China. These buses need to be truly open. Relevant international organizations need to support and help China´s standardization work so that Chinese enterprises can better develop Fieldbus products. (2) The standard text is in the form of the European Fieldbus standard and a bus corresponds to a text, which is user-friendly.

4 Other Fieldbus Standards

In addition to international standards and Chinese standards, there are some fieldbus standards in Europe, Japan, America and other regions. We select some for a brief introduction here.

4.1 European Fieldbus Standards

(1) EN 50170:

EN 50170 is an open international fieldbus standard. In 1996, the European Committee for Electrotechnical Standardization (CENELEC) issued a non-single European standard EN 50170 containing three incompatible protocols: Volume 1: P-Net; Volume 2: ProfiBus; Volume 3: WorldFIP [17]. P-Net is the Danish standard DSF 21906, ProfiBus is the German standard DIN 19245, and WorldFIP is the French standard FIP.

(2) EN 50254:

EN 50254 was prepared by the Technical Committee CENELEC TC 65CX in 1998, including Volume 1: Introduction; Volume 2: Interbus; Volume 3: ProfiBus-DP Profile; Volume 4: WorldFIP Profile 1. Among them, Interbus is the German national standard DIN 19258.

(3) EN 50295:

EN 50295 was prepared by the Technical Committee CENELEC TC 17B in 1999. The title of EN 50295 is "Low voltage switchgear and controlgear—controller and device interface systems actuator sensor interface AS-i".

(4) EN 50325:

EN 50325: Industrial communication subsystem based on ISO 11898(CAN) for controller device interfaces includes:
- EN 50325-1-2001: General requirements
- EN 50325-2-2001: DeviceNet
- EN 50325-3-2001: SDS
- EN 50325-4-2003: CANopen
- EN 50325-5-2010: Functional safety communication based on EN 50325-4 (CANopen Safety).

Among them, CANopen is an application layer protocol based on the CAN. In Europe, CANopen is considered to be a leading standard in CAN-based industrial systems.

4.2 Japanese Fieldbus Standards

(1) CC-Link:

Japan Mitsubishi Electric released the CC - Link fieldbus standard in 1996. It was approved by the International Semiconductor Manufacturers as one of the fieldbus standards used in semiconductor manufacturing (SEMI E54.12) in 2001, and was into the IEC61158 standard and ISO15745-5 standard in 2007.

(2) Vnet/IP:

The real-time Ethernet Vnet/IP is a real-time factory network system developed by Yokogawa Corporation in 2004 for process automation. It has become the IEC/PAS 62405 and IEC 61158 standards.

(3) TCnet:

Tcnet is a real-time Ethernet developed by Toshiba Corporation, which is mainly used in Toshiba´s Toshiba 3000 industrial automation control system. And it is widely used in high speed areas such as drive devices and steel rolling. It has become the IEC/PAS 62406 and IEC 61158 international standards.

4.3 American Fieldbus Standards

American National Standards Institute/National Electrical Manufacturers Association (ANSI/NEMA) supported International Society of Automation (ISA)/IEC standards in the same way:
- ISA/IEC 61158-1 General rules
- ISA/IEC 61158-2 Physical layer specification
- ISA/IEC 61158-3 Link layer service definition
- ISA/IEC 61158-4 Link layer specification
- ISA/IEC 61158-5 Application layer service definition
- ISA/IEC 61158-6 Application layer specification
- ISA/IEC 61158-7 Management system
- ISA/IEC 61158-8 Consistency test.

4.4 Other Fieldbus Standards

In addition to the above standardized fieldbuses, there are other fieldbuses, such as Lonworks, Sensoplex 2, Attached Resource Computer Network (ARCNET), and Dupline.

Lonworks—Local Operating Network was published in 1990 [18]. It was developed by Echelon, and jointly advocated by Echelon, Motorola, and Toshiba. Lonworks has become the recognized international standard for construction industry, and was also used as a China's national standard (GB/Z) in 2006.

Sensoplex 2 is a control network designed by TURCK in Germany for heavy industry automation [8]. It has strong anti-jamming capability and is widely used in the world's major automobile factories.

ARCNET is a widely installed local area network (LAN) technology developed by Datapoint in 1977 [20]. The initial application target of the ARCNET network was office automation. However, as the demand for office network systems shifts to Ethernet, ARCNET networks are gaining new applications in real-time control.

Dupline is a field and installation bus designed by Carlo Gavazzi [21]; it offers unique solutions for building automation and other fields.

5 Conclusions

Fieldbus and Industrial Ethernet are the hotspots of automation technology in the 21st century. There are following benefits to understand and master international, national, regional and other related standards of fieldbus and industrial Ethernet. First, tracking the trend of fieldbus and industrial Ethernet technology in the world promotes the application and development of the fieldbus and industrial Ethernet technology in China. Second, it will provides reference to develop fieldbus and industrial Ethernet standards and improve the standard systems in China. Third, it can guide the development of enterprise products in China, make enterprises develop fieldbus products in accordance with the international mainstream fieldbuses and try to take a larger market share in the country and abroad.

References
[1] MIAO X Q. The Latest Developments in Fieldbus Technology [J]. Process Automation Instrument, 2000, 21(7): pp. 1–4. DOI: 10.3969/j.issn.1000-0380.2000. 07.001
[2] FELSER M, SAUTER T. The Fieldbus War: History or Short Break Between Battles? [C]//4th IEEE International Workshop on Factory Communication Systems, Vasteras, Sweden, 2002: 73–80. DOI:10.1109/WFCS.2002.1159702
[3] THOMESSE J P. Fieldbus Technology in Industrial Automation [J]. Proceedings of the IEEE, 2005, 93(6): 1073–1101. DOI:10.1109/jproc.2005.849724
[4] FELSER M. Real-Time Ethernet—Industry Prospective [J]. Proceedings of the IEEE, 2005, 93(6): 1118-1129. DOI:10.1109/jproc.2005.849720
[5] WOOD G. State of Play [J]. IEE Review, 2000, 46(4): pp. 26–28. DOI: 10.1049/ir:20000404
[6] MIAO X Q. 20 Types of Fieldbuses into the International Standards IEC61158 Fourth Edition [J]. Process Automation Instrument, 2007, 28(z1): pp. 25–29. DOI: 10.3969/j.issn.1000-0380.2007.z1.008

[7] IEC. Digital Data Communications for Measurement and Control—Fieldbus for Use in Industrial Control Systems: IEC 61158 [S]. 2003

[8] TONG W M, MU M, LIN J-B. Fieldbus Standard [J]. Low Voltage Apparatus, 2003(2): pp. 32−36. DOI: 10.3969/j.issn.1001-5531.2003.02.009

[9] IEC/TC Delegation of China. IEC SC65C MT9, WG11 Fieldbus and Real-time Ethernet Working Group Meeting (USA) Minutes [J]. Instrument Standardization and Metrology, 2006(1). DOI:10.3969/j.issn.1672-5611.2006.01.003

[10] WINKEL L. Real-Time Ethernet in IEC 61784-2 and IEC 61158 Series [C]// 4th IEEE International Conference on Industrial Informatics, Singapore, Singapore, 2006: 246−250. DOI:10.1109/INDIN.2006.275788

[11] IEC. Industrial Communication Networks—Fieldbus Specifications—Part 1: Overview and Guidance for the IEC 61158 and IEC 61784 Series: IEC 61158 [S]. 2014

[12] MIAO X Q. The Latest Development in Real-Time Ethernet Technology [J]. Electrical Age, 2005(6): pp. 64− 68. DOI: 10.3969/j.issn.1000 - 453X.2005.06.018

[13] IEC. Low Voltage Switchgear and Controlgear—Controller Device Interfaces (CDIs): IEC62026 [S]. 2008

[14] ISO. Road Vehicles—Controller Area Network (CAN): ISO 11898 [S]. 2015

[15] ISO. Road Vehicles—Low-Speed Serial Data Communication—Part 3:Vehicle Area Network: ISO 11519 [S]. 1995

[16] GUO Q Y, HUANG S Z, XUE J. The Applications of Fieldbus and Industrial Ethernet [M]. Beijing, China: Science Press, 2016.

[17] DEMARTINI C, VALENZANO A. The EN50170 Standard for a European Fieldbus [J]. Computer Standards & Interfaces, 1998, 19(5/6): 257−273. DOI: 10.1016/s0920-5489(98)00027-0

[18] TIAN M, GAO A B. New Development of "LonWorks" Fieldbus Technology [J]. Journal of Harbin University of Science and Technology, 2010, 15(1): 33−39. DOI: 10.3969/j.issn.1007-2683.2010.01.008

[19] NIE X B, WANG L D, SHEN P, et al. Real-Time Performance Analysis and Re-search of ARCNET Network System [J]. Journal of the China Railway Society, 2011, 33(1): pp. 58−62. DOI: 10.3969/j.issn.1001-8360.2011.01.010

[20] GAO Z B, XU G J, WANG B. Implementation of Signal Acquisition System Based on Siemens PLC and Dupline Bus [J]. Nonferrous Metals: Mineral Processing, 2017, 206: 204−206. DOI: 10.3969/j.issn.1671-9492.2017.z1.045

Biographies

CHEN Jinghe (16121422@bjtu.edu.cn) received her Bachelor's degree in electrical engineering from Jinan University, GuangDong in 2016. Now she is a master in electrical engineering of Beijing Jiaotong University, China, and she engages in research of fieldbus and industrial Ethernet.

ZHANG Hesheng received his B.S.E.E. and M.S. degrees in electrical engineering from Northern Jiaotong University of China in 1992 and 1995, respectively. He received his Ph.D. degree in automation science and technology from Tsinghua University, China in 2006. He is now a professor in School of Electrical Engineering, Beijing Jiaotong University, China. His research interests include fieldbus, sensor network, network communication performance, and intelligent control. He is a senior member of the CES and CCF. He has published more than 60 papers in the journals and conferences as the first or second author, and has applied for nine national invention patents as the first inventor.

10.1109/mcom.2017.1600422cm

[47] CARROLL R J, RUPPERT D, CRAINICEANU C M, et al. Measurement Error in Nonlinear Models: A Modern Perspective [M]. London, UK: Chapman and Hall/CRC, 2006

[48] BATTISTELLI G, CHISCI L, FANTACCI C, et al. Networked Target Tracking with Doppler Sensors [J]. IEEE Transactions on Aerospace and Electronic Systems, 2015, 51(4): 3294−3306. DOI: 10.1109/taes.2015.140340

[49] HASSIBI B, SAYED A H, KAILATH T. Indefinite-Quadratic Estimation and Control: A Unified Approach to H2 and H-infinity Theories [M]. Philadelphia, USA: SIAM, 1999: vol 16

[50] KAILATH T, SAYED A H, HASSIBI B. Linear Estimation [M]. Upper Saddle River, USA: Prentice Hall, 2000

Biography

Erich Zöchmann (ezoechma@gmail.com) received all his degrees (B.Sc., Dipl.-Ing, Dr.techn) in electrical engineering from TU Wien, Austria. From 2013 to 2015, he was a project assistant at the Institute of Telecommunications where he co-developed the Vienna LTE-A uplink link level simulator and conducted research on physical layer signal processing for 4G mobile communication systems. From 2015 to 2018 he was involved in experimental characterization and modelling of millimeter wave propagation. From November 2017 until February 2018, he was a visiting scholar at the University of Texas at Austin, USA. Besides wireless propagation, his research interests include physical layer signal processing, array signal processing, compressed sensing, and convex optimization.

SRSC: Improving Restore Performance for Deduplication‑Based Storage Systems

ZUO Chunxue[1], WANG Fang[1], TANG Xiaolan[2], ZHANG Yucheng[1], and FENG Dan[1]

(1. Key Laboratory of Information Storage Systems, Engineering Research Center of Data Storage Systems and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China;
2. 5G Application Product Line, ZTE Corporation, Shenzhen, Guangdong 518057, China)

Abstract: Modern backup systems exploit data deduplication technology to save storage space whereas suffering from the fragmentation problem caused by deduplication. Fragmentation degrades the restore performance because of restoring the chunks that are scattered all over different containers. To improve the restore performance, the state-of-the-art History Aware Rewriting Algorithm (HAR) is proposed to collect fragmented chunks in the last backup and rewrite them in the next backup. However, due to rewriting fragmented chunks in the next backup, HAR fails to eliminate internal fragmentation caused by self-referenced chunks (that exist more than two times in a backup) in the current backup, thus degrading the restore performance. In this paper, we propose Selectively Rewriting Self-Referenced Chunks (SRSC), a scheme that designs a buffer to simulate a restore cache, identify internal fragmentation in the cache and selectively rewrite them. Our experimental results based on two real-world datasets show that SRSC improves the restore performance by 45% with an acceptable sacrifice of the deduplication ratio.

Keywords: data deduplication; fragmentation; restore performance

1 Introduction

With a flood of data from companies, networks, emails systems, individual stations and other devices, data deduplication as a type of compression technology is widely employed in the backup systems for saving storage space [1]–[4], [5]. A recent study shows the amount of digital data in the world will exceed 44 Zettabytes in 2020 [6], while EMC reports that more than 90% data are redundant in backup systems [7]. Therefore, data deduplication is an efficient way in space to eliminate redundant data in modern backup systems. Data deduplication splits the file into fixed-size [8] or variable-sized chunks [1], computes the fingerprint of each chunk with hash algorithms (e.g. SHA-1, MD5) and determines whether a chunk is duplicated by looking up a fingerprint index, which records all the fingerprints of non-duplicate chunks in a backup system. If the fingerprint of a chunk has no identical records in the fingerprint index, the chunk is unique and will be written to a read/write storage unit called "container" with fixed size (e.g. 4 MB [9]).

Restore process is of great importance as backup process for the occurrence of disasters such as earthquakes, tsunami and hurricanes [10]. Higher restore performance satisfies the requirements of higher availability of the system. Restoring a backup stream is based on a backup recipe, which includes the metadata information of data chunks, such as chunk fingerprints, chunk ID and chunk size. From the recipe, a container that the chunk locates is read from the disk to a restore cache. When a restore cache is full, it will evict an entire container with a cache replacement algorithm.

Since duplicate chunks are removed from multiple backups, the chunks of a backup stream are physically scattered across different containers, which introduces two types of fragmentation. One is inter-version fragmentation caused by duplicate chunks among multiple versions of the same backup; the other is internal fragmentation caused by duplicate chunks (often called self-referenced chunks [11]) in a single backup. In order to restore a backup stream with fragmentation, multiple containers are read from the disk. Because of the poor random-access performance of hard disk drives (HDDs), fragmentation results in the severely degradation of the restore performance.

To address the fragmentation problem, the state-of-the-art History Aware Rewriting Algorithm (HAR) is proposed to iden-

tify and collect the duplicate but fragmented chunks in the last backup, and then rewrite them in the next backup. However, for the internal fragmentation, HAR fails to eliminate them because identified internal fragmentation are rewritten in the next backup, rather than being handled immediately. In this paper, we propose an efficient approach called Selectively Rewriting Self‑Referenced Chunks (SRSC). The main idea behind SRSC is to maintain a fixed‑size buffer, identify whether the self‑referenced chunks are fragmented in a buffer and rewrite them in the backup.

The main contributions of this paper include:

(1) As HAR collects fragmented chunks in the last backup and rewrites them in the next backup, we observe that HAR cannot address the internal fragmented chunks caused by self‑referenced chunks in a backup.

(2) To reduce internal fragmented chunks in the deduplication‑based backup systems, we propose a SRSC scheme to maintain a buffer to simulate a restore cache, identify internal fragmented chunks in the buffer and rewrite them in the current backup.

(3) We implement our scheme in the deduplication‑based system. Compared with HAR, the experimental results based on two datasets show that SRSC is more efficient in eliminating internal fragmentation and significantly improving the restore performance.

The rest of paper is organized as follows. In Section 2, we review the background for our research. In Section 3, we present our observations to motivate our research. We describes the detailed architecture overview and implementation of our algorithm in Section 4. We present the experimental evaluations with two datasets in Section 5. In Section 6, we present the related work. We conclude our paper in Section 7.

2 Background

2.1 Fragementation Problem

Fragmentation problem has been received a broad attention in the deduplication‑based backup systems. **Fig. 1** shows the fragmentation arises between two backups. Both two backups have 12 chunks. After backup 1, 12 unique chunks of backup
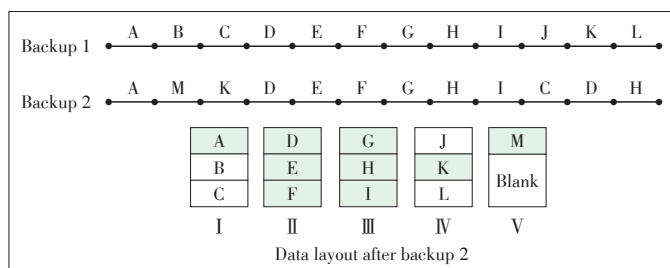
▲Figure 1. Fragmentation appears between two backup streams. The shaded chunks represent the chunks referenced by the second back–up in each container. The blank areas offer extra space of the half‑full containers when a backup completes.

1 are stored in the containers I, II, III, and IV. Since 11 chunks of backup 1 are identical to that chunks of backup 2, only chunk M is stored in the container V after backup 2. The chunks of backup 1 are aggregated in the first 4 containers. However, the chunks of backup 2 are scatter across 5 different containers, which is called fragmentation problem.

With a 3‑container‑size Least Recently Used Algorithm (LRU), restoring backup 1 obviously needs to read 3 containers. However, restoring backup 2 needs to read 9 containers. Thus, the restore performance of backup 2 is worse than that of backup 1. This is because reading containers I and II is not efficient and each of the container only includes one useful chunk for backup 2. And then, for restoring a chunk A required by backup 2, we need to read an entire container I. Thus, chunk A is fragmented chunks for backup 2.

2.2 History‑Aware Rewriting Algorithm

To address the fragmentation problem and improve the restore performance, HAR is proposed. HAR is based on a key observation that two consecutive backups are very similar, and thus history information collection during the last backup is useful for the performance of the next backup.

Specifically, HAR first defines a container's utilization for a backup stream as the fraction of its chunks referenced by the backup. If the utilization of a container is smaller than 50%, the container is regarded as a sparse container that will amplify the read performance. Next, HAR observes that sparse containers of the current backup remain sparse in the next backup. Based on the observation, HAR loads the IDs of sparse containers identified by the last backup, and identifies the sparse containers and rewrites fragmented chunks in the sparse containers. Lastly, HAR collects the IDs of sparse containers as the history information and exploits it for identifying and rewriting fragmented chunks in the next backup. Thus, HAR accurately identifies the fragmented chunks and improves the restore performance.

3 Motivation

Deduplication is important in backup systems for saving storage space, but introduces fragmented chunks. As described in Section 2.1, the chunks of a backup are scattered in different containers because of the removel of duplicate chunks, which causes the inter‑version fragmentation, thus reducing the restore performance. Similarly, for a single backup stream, self‑referenced chunks, such as chunk D in backup 2, can also reduce the restore performance. This is because, with a 3‑container‑size LRU, restoring chunk D two times requires to read the container II two times, which severely reduces the restore performance. Thus, an existing rewriting algorithm such as HAR has been proposed to rewrite inter‑version fragmented chunks identified by the last backup to improve the restore performance. However, the design of HAR ignores that internal

fragmentation caused by self‑referenced chunks also has a negative impact on the restore performance.

Specifically, at the beginning of a backup, HAR firstly loads the fragmented chunks of the last backup and rewrites them. When the backup finishes, HAR collects information of fragmentation in the sparse containers prepared for the next backup. Obviously, HAR ignores to identify internal fragmentation stemming from self‑referenced chunks in a backup, and thus fails to reduce internal fragmentation. In other word, HAR collects fragmented chunks of each backup, but these fragmented chunks come from duplicate chunks between multiple versions of the same backup. Hence, HAR does not consider the impact of internal fragmentation on the restore performance, which hinders HAR from further improving the restore performance.

Fig. 2 shows an example of the HAR working process, in which all the backup streams have 12 chunks. After backup 1, 12 unique chunks are stored in four containers. Since 10 chunks in backup 2 are duplicate with backup 1, only chunks M and N are written to the containers IV and V. After backup 2, containers' utilization are computed, thus chunks C and D are fragmented in the containers I and II and collected by HAR. When the third backup stream arrives, chunks C and D that have been recorded in backup 2 are rewritten to the container VI in backup 3 (Fig. 2f). We observe that chunks I and J as self‑referenced chunks are fragmented because they will lead containers III and IV to be read more than once in a limit
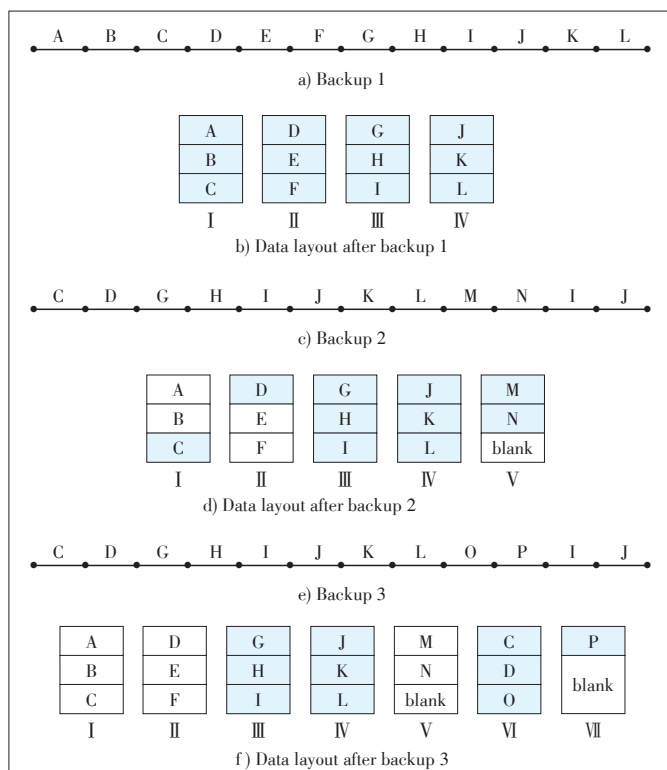
restore cache. However, after HAR algorithm finishes, chunks I and J are still fragmented chunks and not be eliminated. Assume a backup stream has many self‑referenced chunks, multiple containers will be accessed during a restore. Thus, HAR increases the number of accessed containers and is an inefficient way to reduce internal fragmented chunks caused by self‑referenced chunks.

The above observation motivates us to propose a scheme to reduce internal fragmented chunks. We have proposed a SRSC scheme that maintains a fixed‑size buffer to simulate a restore cache, identifies self‑referenced chunks whether fragmented in the buffer, and selectively rewrite self‑referenced chunks based on computing their containers' utilizations. In such a scheme, internal fragmentation could be accurately identified. Meanwhile, in order to save storage space, we selectively rewrite a part of self‑referenced chunks by choosing the low containers' utilization. Therefore, SRSC not only reduces the number of containers, but also improves the restore performance.

4 Selectively Rewriting Self-Referenced Chunks

4.1 Overview of SRSC

To reduce internal fragmented chunks caused by self‑referenced chunks, our SRSC Scheme is proposed to identify self‑referenced chunks and selectively rewrite them. As our observation in Section 3, with the increase of backup versions, HAR rewrites fragmented chunks from the last backup but cannot deal with internal fragmented chunks in the current backup. This is because the distance between two self‑referenced chunks exceeds the restore cache size during a restore, which causes multiple containers to be read and the restore performance to decline. Thus, a buffer is designed for SRSC to simulate a restore cache. In this buffer, SRSC identifies whether self‑referenced chunks are fragmented and then rewrite these fragmented chunks. However, there are at least 20% self‑referenced chunks in a backup stream. Rewriting a number of fragmented chunks caused by self‑referenced chunks will slow down the backup time and occupy much storage space, which motivates us to rewrite only a part of self‑referenced chunks of a backup stream. Thus, SRSC selectively rewrites self‑referenced chunks based on the containers' utilization. Except for addressing internal fragmented chunks, we also add HAR rewriting algorithm to the SRSC scheme. This is because HAR takes better advantage of heritability of fragmentation, which is efficient in eliminating inter-version fragmented chunks among multiple versions of backups. Therefore, SRSC can reduce fragmented chunks no matter whether the fragmentation comes from a backup or multiple versions of backups.

▲Figure 2. An example of three consecutive backups with the HAR rewriting scheme. The shaded chunks represent the chunks referenced by the current backup in each container.

4.2 Design of SRSC

SRSC is implemented by filter rewriting and selective rewrit-

ing. **Fig. 3** shows the SRSC architecture.

(1) Rewriting Filter

As shown in Fig. 3, the rewriting filter module, stemming from HAR rewriting algorithm, includes two data structures: the conditional container IDs and history information collection. During a backup, the IDs of fragmentation's containers in the last backup are loaded into the conditional container IDs. The rewriting filter module then checks whether the container's IDs of duplicate chunks exist in the conditional container IDs. Before finishing the backup, the history information collection is responsible for collecting the information of fragmented chunks of the current backup, such as containers' IDs of chunks and preparing for the rewriting phase of the next backup.

(2) Selective Rewriting

Whether self‑referenced chunks negatively impact containers' replacement in a restore cache depends on the distance between two self‑referenced chunks, so selective rewriting is designed to identify internal fragmented chunks and selectively rewrite identified fragmentation. Selective rewriting scheme includes two data structures: fragmentation identification and selectively rewriting self-referenced chunks.

Specifically, fragmentation identification first creates a fixed-size buffer to simulate a restore cache, about 256 MB for experiments. In the buffer, the scheme of selective rewriting identifies self‑referenced chunks by checking all the duplicate chunks whether the container ID of the chunks is larger than the total number of the containers stored in the last backup. If

so, the chunk is self‑referenced, and then fragmentation identification will match the container ID of the chunk with that in the buffer. Finding the identical container's ID means that this self‑referenced chunk will not have an impact on the restore performance. Otherwise, the self‑referenced chunk is fragmented and needs to be rewritten to the disk. However, considering the storage overheads and deduplication efficiency, the data structure of selectively rewriting self‑referenced chunks is used for limiting the number of self‑referenced chunks rewritten according to the container's utilization. Thus, the container's ID of this self‑referenced chunk is sent to SRSC and is determined whether its chunk needs to be written.

We first define a container's utilization threshold as 50%. Next, SRSC computes the utilization of the container that identified fragmentation belongs to. If the container's utilization is smaller than 50%, this self‑referenced chunk will be rewritten to the container. Hence, the scheme of selective rewriting not only identifies fragmented chunks by stimulating a restore cache to check if the container ID of the self-referenced chunk can be hold, but also writes a part of the backup stream by using a container's utilization threshold. The work flow of SRSC is shown in **Algorithm 1**.

Algorithm 1. SRSC Algorithm

1: Initialize a buffer S and define the total number of container N;
2: **while** the backup is not completed do
3: **if** the container ID of the chunks is larger than N then
4:    the chunk is a self-referenced chunk
5:    **if** container ID that the self-referenced chunk locates is in S **then**
6:      Compute the container's utilization C_{uti}.
7:      **if** $C_{uti} < 50\%$ then
8:        the self-referenced chunk will be rewritten to the container.
9:      **end if**
10:   **else**
11:     Insert the container ID to S.
12:   **end if**
13: **end if**
14: **end while**
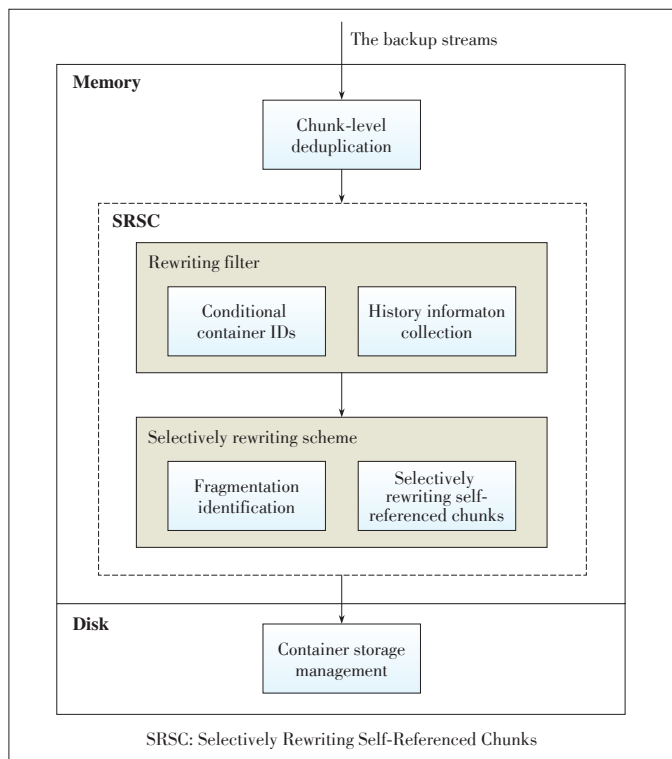
▲Figure 3. Modules and main data structure of SRSC.

5 Performance Evaluation

5.1 Experiment Setup

(1) Platform

We implement the SRSC algorithm on an opensource deduplication system called Destor [12], which is running on the Ubuntu12.04.2 operating system. The operating system is con-

figured as follows: a 16 GB RAM, a quad core Intel i7‐4770 processor at 3.4 GHz. We evaluate SRSC in terms of deduplication ratio and restore performance. Since our paper aims to improve the restore performance of HAR by reducing internal fragmentation, HAR is the baseline of our evaluation, and HAR and SRSC are compared.

(2) Datasets

WEBS, VMDK and FSLHomes datasets are used for evaluations (**Table 1**). Specifically, the WEBS dataset is from the backups of the Sina web with using a tool "wget" to grasp every day. VMDK is a set of virtual machines, including 30 versions. Each backup is 20 GB on average and has 20% self-referenced chunks. The total size of the datasets is about 369 GB. FSLHomes is an open source of traces and snapshots that are collected by file systems and the storage lab and its collaborators, which can be downloaded from the website http://tracer. filesystems.org [13].

(3) Configuration

The backup system divides each dataset into variable‐size chunks by using the Rabin chunking algorithm and computes the fingerprints of each chunk with the MD5 hash algorithm. The fingerprint index is stored in memory by default. And the restore cache prefetches and evicts the containers with optimal replacement algorithm [11]. We use the deduplication ratio to evaluate the deduplication efficiency and the speed factor to evaluate the restore performance of SRSC.

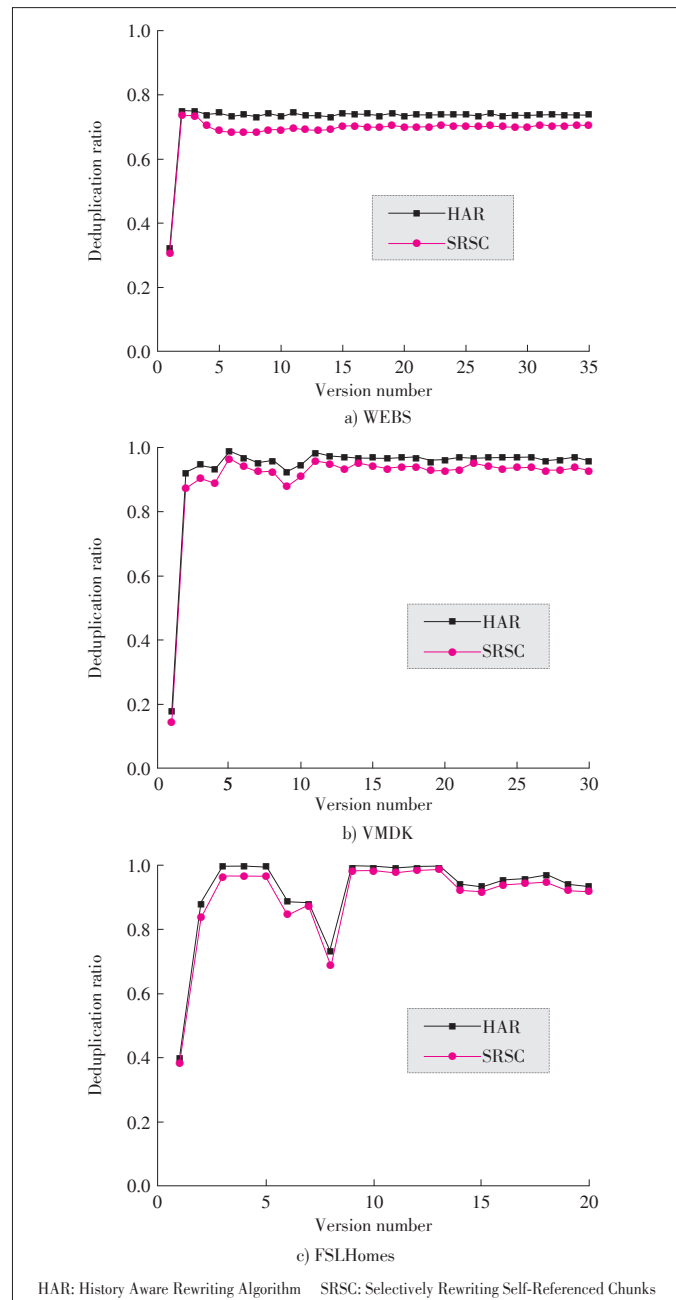## 5.2 Experimental Results and Analysis

5.2.1 Deduplication Ratio

In this evaluation, we use the deduplication ratio as a metric to evaluate the deduplication efficiency. Deduplication ratio is defined as the ratio of the size of duplicate chunks and a file size. **Fig. 4** show the deduplication ratio of HAR and SRSC based on the three datasets. In general, the deduplication ratios of SRSC are lower than HAR. This is because HAR addresses the inter‐version fragmented chunks, and thus rewrite fragmented chunks of the last backup. However, compared with HAR, our SRSC scheme rewrites not only inter‐version fragmented chunks of the last backup, but also writes a part of self‐referenced chunks for addressing the new fragmented chunks caused by selfreferenced chunks in a backup stream. Meanwhile, rewriting fragmented chunks suggests that less duplicate chunks lead to the reduction of the deduplication ratio.

Specifically, for the WEBS dataset, the deduplication ratio of HAR is 73% on average. The deduplication ratio of SRSC is

▼Table 1. Characteristics of three datasets

Dataset name	Total size (GB)	Version number	Deduplication ratio
WEBS	105	35	73%
VMDK	550	30	92%
FSLHomes	860	20	91%

a) WEBS

b) VMDK

c) FSLHomes

HAR: History Aware Rewriting Algorithm   SRSC: Selectively Rewriting Self-Referenced Chunks

▲Figure 4. The comparisons between SRSC and HAR in terms of deduplication ratio based on the three datasets.

69% on average, 4% lower than HAR. For the VMDK dataset, the deduplication ratio of HAR is 92% on average. And the deduplication ratio of SRSC is 88% on average, 4% lower than HAR. Similarly, the deduplication ratio of SRSC is 3% lower than HAR for the FSLHomes dataset. Although SRSC's deduplication ratio is a little lower than HAR regardless of the dataset's type, it is reasonable and acceptable.
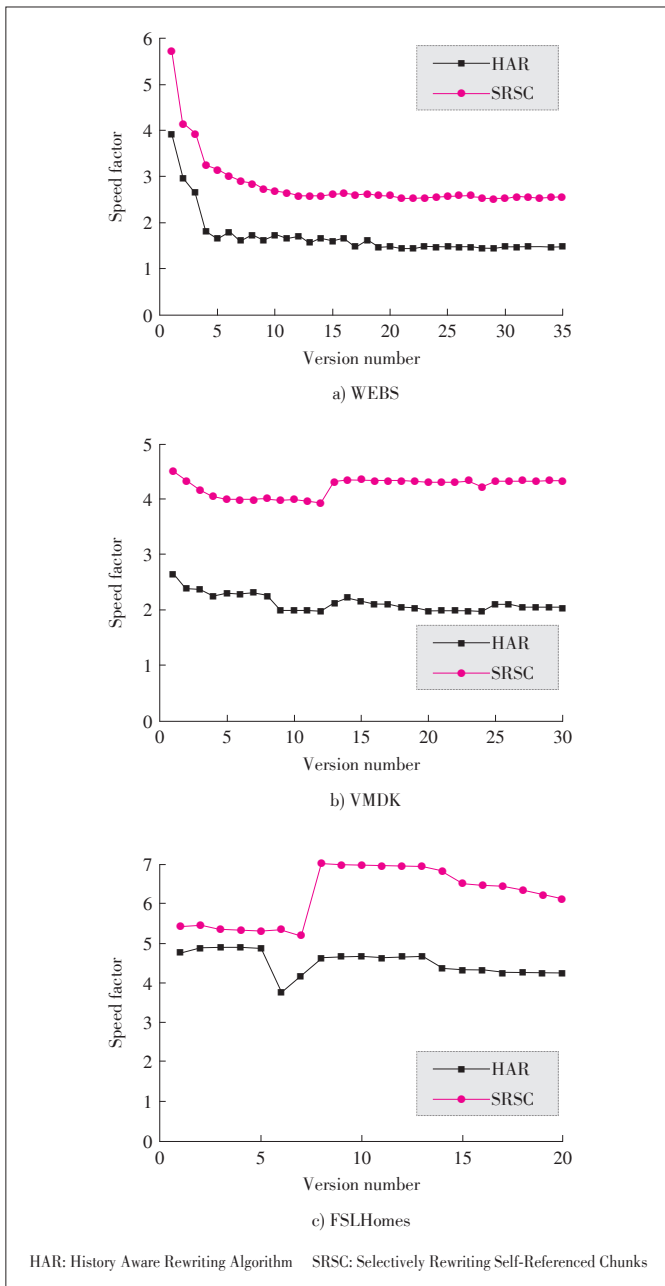
5.2.2 Restore Performance

We use the speed factor [11] as a metric to evaluate the re-

store performance. Speed factor represents 1/mean containers read per MB of restored data. **Fig. 5** shows the restore performance between HAR and SRSC based on the three different datasets, which are widely used in the deduplicationbased backup systems [11]. In the figure, SRSC achieves better restore performance than HAR in three data sets. For the WEBS dataset, SRSC outperforms HAR by 40%. For the VMDK dataset, SRSC achieves better restore performance by 48%. For the FSLHomes dataset, SRSC outperforms HAR by 27%. This is because SRSC not only addresses the inter‑version fragmenta-

tion but also deals with the internal fragmentation. Fragmented chunks are eliminated, which means that the utilization of each container increases. In addition, it also suggests that self‑referenced chunks have a negative impact on the restore performance. In general, SRSC achieves higher restore performance than HAR.

Fig. 6 shows the restore performance between HAR and SRSC under different cache sizes. In WEBS, SRSC significantly improves the restore performance than HAR. Because both inter‑version fragmentation and internal fragmentation can de-

HAR: History Aware Rewriting Algorithm   SRSC: Selectively Rewriting Self-Referenced Chunks

▲Figure 5. The comparisons between SRSC and HAR in terms of restore performance. The cache is 256‑, 128‑, and 256‑container‑sized in WEBS, VMDK and FSLHomes respectively.

HAR: History Aware Rewriting Algorithm   SRSC: Selectively Rewriting Self-Referenced Chunks
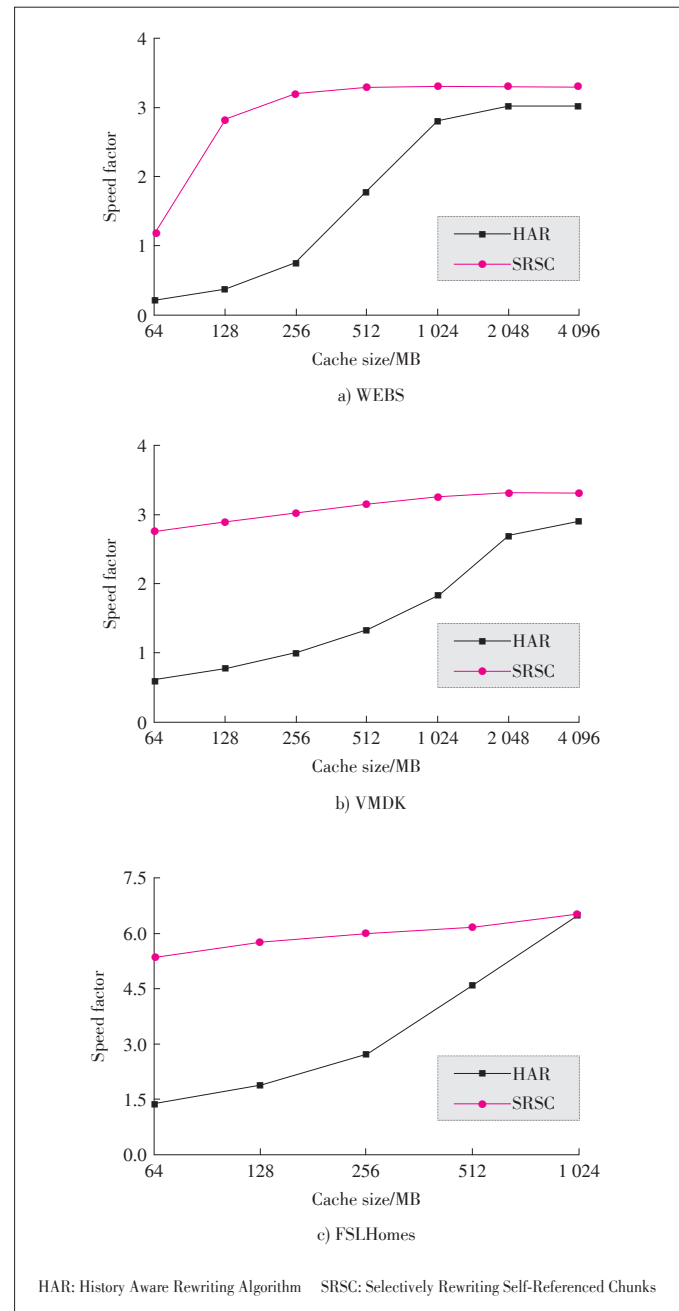
▲Figure 6. The comparisons between SRSC and HAR in terms of restore performance under different cache sizes.

grade the restore performance, SRSC reduces both types of fragmentation.

Note that, with the increase of the cache size, the restore performance becomes higher. This is because a large cache can hold more containers for reducing the number of the containers evicted by the cache replacement algorithm. Moreover, we observe that the restore performance increases more slowly when the speed factor reaches a certain value. This is because the cache size of 1 024 MB can restore a backup stream of WEBS. That is to say, the cache size bigger than 1 024 MB is enough to hold a whole dataset, and thus the speed factor becomes slowly.

5.3 Simulated Buffer Size

In this paper, the simulated buffer size is important for the fragmentation identification. This is because a large buffer (e. g., 1 GB size) can hold more containers than a small one at the expense of more memory overheads. However, a small buffer (e. g., 8 MB) identifies more self-referenced chunks as fragmented chunks than a large one. These fragmented chunks will be rewritten, which reduces the deduplication ratio. Thus, SRSC uses a 256 MB buffer, which is reasonable. We find from Fig. 6 that the restore performance keeps relatively stable when the cache size reaches up to 256 MB. Even though the cache size continues to increase, the restore performance improves sightly about 1%−2%. In addition, as shown in Fig. 4, the deduplication ratio does not drop much with using a 256 MB buffer. Based on the observation, a large cache size (more than 256 MB size) is not necessary. Thus, the 256 MB buffer is sufficient for simulating a restore cache.

6 Related Work

6.1 Data Deduplication

Deduplication is an indispensable component for the backup systems to achieve the goal of saving storage space [15]−[17]. Recent studies for deduplication technology mainly focus on the problem of fast content-defined chunking and large fingerprint indexing. To improve the speed of chunking, some solutions [18]−[21] have been proposed to find an appropriate cut point to achieve a low computation overhead and high chunking throughput for accelerating the process of chunking. To address the large fingerprint indexing, Extreme Binning [22], Silo [23], and Sparse Index [2] exploit the locality of files, the similarity of files, or combines the locality and similarity of the files to improve the performance of fingerprint indexing. Different from above studies, our SRSC algorithm mainly focuses on addressing the fragmentation problem for improving the restore performance.

6.2 Restore

The fragmentation problem caused by in-line deduplication

has a negative impact on the restore performance. Rewriting algorithms such as context-based rewriting (CBR), capping (CAP), and HAR are proposed to address the fragmentation problem in the deduplication-based backup systems.

CBR [23] uses a sliding window to buffer a small part of the backup stream for identifying fragmented chunks. For each of backup streams, CBR defines the rewrite utility, which is the quotient of the size of chunks that are in the disk not in the backup stream and the total size of the disk, to decide whether it is fragmented. If the rewrite utility of the chunk exceeds the rewrite utility threshold (e.g., 0.5), the chunk is the fragmentation. CAP [24] directly divides the backup stream into fixed-size segments (e.g., 20 MB). In each limited segment, CAP counts the number of referenced containers (N) to recognize fragmented chunks. Assume that N is higher than the threshold value T, the chunks in the containers that hold the least referenced number are regarded as fragmented chunks. HAR [11] is based on the observation that fragmented chunks in the current backup remain fragmented in the subsequent backup. Hence, HAR rewrites the fragmented chunks identified in the last backup, records fragmentation information of the current backup and rewrites them in the next backup.

Rewriting algorithms improve the restore performance with the sacrifice of the deduplication ratio. Some studies take advantage of such characteristic that the read sequence during the restore is identical to the write sequence during the backup to leverage future knowledge for improving the restore performance. The forward assembly area (FAA) [24] maintains an assembly buffer to be filled out the chunks based on the information of a recipe, and thus ensures that the container is read only once in a small space. Limited Forward Knowledge (LFK) uses a backup recipe to improve the restore performance. Park et al. [25] propose a new cache design to evict the container that has the smallest future access by using a lookahead window and maintain a small log to hold evicted containers in order to maximize the cache hit.

In addition to above algorithms, some other schemes are proposed to improve the restore performance. iDeDup [26] exploits spatial locality to selectively eliminate sequential blocks and reduce fragmentation for primary storage systems. RevDedup [27] puts forward a mixed inline and offline deduplication scheme in order to keep the layout of the most up-todate backup as sequential as possible. Mao et al. [28] propose to maintain the chunks with a high reference count on solid state disks (SSDs) of the cloud server to improve the restore performance. CABdedup [29] improves the restore performance by identifying and removing unmodified data transmissions from successive versions of the backup datasets between the client and the cloud.

7 Conclusions

In this paper, we propose SRSC, a scheme that simulates a

restore cache to identify whether self‑referenced chunks are fragmented and then selectively rewrite them. SRSC not only eliminates internal fragmentation that HAR cannot address, but also further improves the restore performance. Experimental results based on two datasets demonstrate that SRSC improve the restore performance by 45% at an acceptable cost in deduplication ratio.

References

[1] ZHU B, LI K, PATTERSON H. Avoiding the Disk Bottleneck in the Data Domain Deduplication File System [C]//6th USENIX Conference on File and Storage Technologies, San Jose, USA, 2008. DOI: 10.1126/science.1164390

[2] LILLIBRIDGE M, ESHGHI K, BHAGWAT D, et al. Sparse Indexing: Large Scale, Inline Deduplication Using Sampling and Locality [C]//7th USENIX Conference on File and Storage Technologies, San Francisco, USA, 2009. DOI: 10.1145/2187836.2187884

[3] DUBNICKI C, GRYZ L, HELDT L, et al. Hydrastor: A Scalable Secondary Storage [C]//7th USENIX Conference on File and Storage Technologies, San Francisco, USA, 2009

[4] FU M, FENG D, HUA Y, et al. Design Tradeoffs for Data Deduplication Performance in Backup Workloads [C]//13th USENIX Conference on File and Storage Technologies, Santa Clara, USA, 2015: 331–344

[5] MEYER D T, BOLOSKY W J. A Study of Practical Deduplication [J]. ACM Transactions on Storage, 2012, 7(4): 1–20. DOI: 10.1145/2078861.2078864

[6] IDC. The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things [EB/OL]. (2014‑04). http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

[7] WALLACE G, DOUGLIS F, QIAN H, et al. Characteristics of Backup Workloads in Production Systems [C]//10th USENIX Conference on File and Storage Technologies, San Jose, USA, 2012

[8] QUINLAN S, DORWARD S. Venti: A New Approach to Archival Storage [C]//USENIX Symposium on Networked Systems Design and Implementation, Monterey, USA, 2002: 89–102

[9] GUO F, EFSTATHOPOULOS P. Building a High‑Performance Deduplication System [C]//USENIX Annual Technical Conference, Portland, USA, 2011

[10] PRESTON W C. Backup and Recovery [M]. Sebastopol, USA: O'Reilly Media, 2006

[11] FU M, FENG D, HUA Y, et al. Accelerating Restore and Garbage Collection in Deduplication‑Based Backup Systems via Exploiting Historical Information [C]//USENIX Annual Technical Conference, Philadelphia, USA, 2014

[12] FU M. Destor: An Experimental Platform for Chunk‑Level Data Deduplication [EB/OL]. (2014). https://github.com/fomy/destor

[13] TARASOV V, MUDRANKIT A, BUIK W, et al. Generating Realistic Datasets for Deduplication Analysis [C]//USENIX Annual Technical Conference, Boston, USA, 2012

[14] BIGGAR H. Experiencing Data De‑Duplication: Improving Efficiency and Reducing Capacity Requirements[R]. The Enterprise Strategy Group, 2007

[15] ASARO T, BIGGAR H. Data De‑Duplication and Disk‑To‑Disk Backup Systems: Technical and Business Considerations [R]. The Enterprise Strategy Group, 2007

[16] XIA W, JIANG H, FENG D, et al. A Comprehensive Study of the Past, Present, and Future of Data Deduplication [J]. Proceedings of the IEEE, 2016, 104(9): 1681–1710. DOI: 10.1109/JPROC.2016.2571298

[17] RABIN M O. Fingerprinting by Random Polynomials [R]. Center for Research in Computing Tech., Aiken Computation Laboratory, Univ., 1981

[18] KRUUS E, UNGUREANU C, DUBNICKI C. Bimodal Content Defined Chunking for Backup Streams [C]//8th USENIX Conference on File and Storage Technologies, San Jose, USA, 2010

[19] AGARWAL B, AKELLA A, ANAND A, et al. Endre: An End‑System Redundancy Elimination Service for Enterprises [C]//7th USENIX Symposium on Networked Systems Design and Implementation, San Jose, USA, 2010

[20] ZHANG Y C, JIANG H, FENG D, et al. AE: An Asymmetric Extremum Content Defined Chunking Algorithm for Fast and Bandwidth‑Efficient Data Deduplication [C]//IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 2015: 1337–1345. DOI: 10.1109/INFOCOM.2015.7218510

[21] BHAGWAT D, ESHGHI K, LONG D D E, et al. Extreme Binning: Scalable, Parallel Deduplication for Chunk‑Based File Backup [C]//IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, London, UK, 2009: 1–9. DOI: 10.1109/MASCOT.2009.5366623

[22] XIA W, JIANG H, FENG, et al. Silo: A Similarity‑Locality Based Near‑Exact Deduplication Scheme with Low RAM Overhead and High Throughput [C]//USENIX Annual Technical Conference, Portland, USA, 2011

[23] KACZMARCZYK M, BARCZYNSKI M, KILIAN W, et al. Reducing Impact of Data Fragmentation Caused by In‑Line Deduplication [C]//ACM SYSTOR, Haifa, Israel, 2012. DOI: 10.1145/2367589.2367600

[24] LILLIBRIDGE M, ESHGHI K, BHAGWAT D, Improving Restore Speed for Backup Systems that Use Inline Chunk‑Based Deduplication [C]//11th USENIX Conference on File and Storage Technologies, San Jose, USA, 2013. DOI: 10.1145/2385603.2385607

[25] PARK D, FAN Z Q, NAM Y J, et al. A Lookahead Read Cache: Improving Read Performance for Deduplication Backup Storage [J]. Journal of Computer Science and Technology, 2017, 32(1): 26–40. DOI: 10.1007/s11390-017-1680-8

[26] SRINIVASAN K, BISSON T, GOODSON G, et al. IDedup: Latency‑Aware, Inline Data Deduplication for Primary Storage [C]//10th USENIX Conference on File and Storage Technologies, San Jose, USA, 2012: 299–312. DOI: 10.1111/j.1360-0443.2007.01823.x

[27] NG C‑H, LEE P P. Revdedup: A Reverse Deduplication Storage System Optimized for Reads to Latest Backups [C]//ACM Asia‑Pacific Workshop on Systems, Singapore, Singapore, 2013. DOI: 10.1145/2500727.2500731

[28] MAO B, JIANG H, WU S Z, et al. SAR: SSD Assisted Restore Optimization for Deduplication‑Based Storage Systems in the Cloud [C]//IEEE Seventh International Conference on Networking, Architecture, and Storage, Xiamen, China, 2012: 328-337. DOI: 10.1109/NAS.2012.48

[29] TAN Y, JIANG H, FENG D, TIAN L, YAN Z. Cabdedupe: A Causality‑Based Deduplication Performance Booster for Cloud Backup Services [C]//IEEE International Parallel & Distributed Processing Symposium, Anchorage, USA, 2011. DOI: 10.1109/IPDPS.2011.76

Biographies

ZUO Chunxue (cxzuo@hust.edu.cn) is currently working toward the Ph.D. degree in computer architecture at Huazhong University of Science and Technology, China. Her research interest is data deduplication and restore performance.

WANG Fang received the B.E., M.E., and Ph.D. degrees in computer science and technology from Huazhong University of Science and Technology (HUST), China in 1994, 1997, and 2001, respectively. She is a professor of the School of Computer Science and Technology, HUST. Her research interests include computer architecture, massive storage systems, and parallel file systems. She has more than 40 publications to her credit in journals and international conferences including ACM TACO, SC, MSST, ICPP, ICA3PP, HPDC, and ICDCS.

TANG Xiaolan received the M.E. degree in physical electronics from Huazhong University of Science and Technology, China. She is currently a project manager with ZTE Corporation. Her research interests include core network, cloud storage, and 5G technologies.

ZHANG Yucheng is currently a Ph.D. student majoring in computer architecture at Huazhong University of Science and Technology, China. His research interests include data deduplication, storage systems, etc. He has several papers in refereed journals and conferences including IEEE-TC, INFOCOM, etc.

FENG Dan received the B.E., M.E., and Ph.D. degrees in computer science and technology from Huazhong University of Science and Technology (HUST), China in 1991, 1994, and 1997, respectively. She is a professor and the dean of the School of Computer, HUST. Her research interests include computer architecture, and massive storage systems. She has many publications in major journals and international conferences, including IEEE-TC, IEEETPDS, FAST, USENIX ATC, and MSST.

ZTE Communications Guidelines for Authors

Remit of Journal
ZTE Communications publishes original theoretical papers, research findings, and surveys on a broad range of communications topics, including communications and information system design, optical fiber and electro−optical engineering, microwave technology, radio wave propagation, antenna engineering, electromagnetics, signal and image processing, and power engineering. The journal is designed to be an integrated forum for university academics and industry researchers from around the world.

Manuscript Preparation
Manuscripts must be typed in English and submitted electronically in MS Word (or compatible) format. The word length is approximately 3000 to 8000, and no more than 8 figures or tables should be included. Authors are requested to submit mathematical material and graphics in an editable format.

Abstract and Keywords
Each manuscript must include an abstract of approximately 150 words written as a single paragraph. The abstract should not include mathematics or references and should not be repeated verbatim in the introduction. The abstract should be a self−contained overview of the aims, methods, experimental results, and significance of research outlined in the paper. Five carefully chosen keywords must be provided with the abstract.

References
Manuscripts must be referenced at a level that conforms to international academic standards. All references must be numbered sequentially in−text and listed in corresponding order at the end of the paper. References that are not cited in−text should not be included in the reference list. References must be complete and formatted according to ZTE Communications Editorial Style. A minimum of 10 references should be provided. Footnotes should be avoided or kept to a minimum.

Copyright and Declaration
Authors are responsible for obtaining permission to reproduce any material for which they do not hold copyright. Permission to reproduce any part of this publication for commercial use must be obtained in advance from the editorial office of *ZTE Communications*. Authors agree that a) the manuscript is a product of research conducted by themselves and the stated co−authors, b) the manuscript has not been published elsewhere in its submitted form, c) the manuscript is not currently being considered for publication elsewhere. If the paper is an adaptation of a speech or presentation, acknowledgement of this is required within the paper. The number of co−authors should not exceed five.

Content and Structure
ZTE Communications seeks to publish original content that may build on existing literature in any field of communications. Authors should not dedicate a disproportionate amount of a paper to fundamental background, historical overviews, or chronologies that may be sufficiently dealt with by references. Authors are also requested to avoid the overuse of bullet points when structuring papers. The conclusion should include a commentary on the significance/future implications of the research as well as an overview of the material presented.

Peer Review and Editing
All manuscripts will be subject to a two−stage anonymous peer review as well as copyediting, and formatting. Authors may be asked to revise parts of a manuscript prior to publication.

Biographical Information
All authors are requested to provide a brief biography (approx. 100 words) that includes email address, educational background, career experience, research interests, awards, and publications.

Acknowledgements and Funding
A manuscript based on funded research must clearly state the program name, funding body, and grant number. Individuals who contributed to the manuscript should be acknowledged in a brief statement.

Address for Submission
http://mc03.manuscriptcentral.com/ztecom

ZTE COMMUNICATIONS
中兴通讯技术（英文版）