SVM-based online learning for interference-aware multi-cell mmWave vehicular communications

Dapeng Li1,2 | Jiangpei Zhu1 | Haitao Zhao1 | Xiaoming Wang1 | Rui Jiang1

1 College of Telecommunications, and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
2 Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education, Nanjing 210003, China

Abstract

This paper proposes a data-driven method of mmWave beam selection in multi-cell systems to achieve a near-optimal fast beam allocation with low complexity. In particular, an online learning algorithm based on support vector machine (SVM) equipped with the radial basis function kernel, namely SVM-based online beam selection (SBOS) algorithm is proposed. The proposed algorithm starts with an adaptive beam selection process for certain traffic patterns that uses an SVM learning model to adaptively refine the beam selection strategy. Specifically, SVM-based model labels the feedback (the average information rate) from the cellular system, then learns from samples, and makes the scheme space smaller by maximising samples’ minimum distances to all labelled samples in the sample space constrained by newly learned boundaries. Then, according to the aggregated data about the traffic patterns and the performance of corresponding beam selection strategy, SBOS algorithm exploits beam selection schemes recorded in the database or explores new schemes for unknown situations, respectively, and how to tune the hyperparameters for the SBOS algorithm is discussed. Furthermore, the extensive simulation results show that the proposed algorithm achieves a better performance versus upper confidence bound and Random methods.

1 | INTRODUCTION

MmWave cellular systems are expected to be densely deployed to provide multi-Gbps links to support high data rate by using the small-size antennas [1–3]. Especially, multi-Gbps links will be necessary for future vehicle-to-everything (V2X) communications to support the intensive sensory data transmission for (semi)-autonomous driving or the augmented reality (AR) and virtual reality (VR) services for passengers [4, 5]. However, mmWave signals are prone to the environment blockages due to high severe path loss. Such losses can be compensated with the elaborate beamforming which can be enabled by large antenna arrays to provide extra transmit gains. One promising solution is to let base stations (BSs) or transmission points (TPs) in different geographical cells coordinate in transmission [6].

Choosing suitable beams for users is a high complexity-cost procedure, especially for high-speed vehicle communications, since the time-varying channel characteristics will lead to extra channel estimation overhead and access delay, which may cause serious problems for vehicle time-sensitive services [3, 7–9]. Therefore, developing a low-complexity method is motivated to mitigate interference via the adaptively adjusting of spatial beam patterns.

Machine learning (ML) technologies about extracting hidden insights from training data has been regarded as the basic discipline for cellular networks. The deep neural network (DNN) learning as model-free and data-driven approaches have been studied extensively to reduce the computational complexity with available training. For example, the autoencoders are used in [10] to model an end-to-end communication system comprising the encoding, channel and decoding blocks for MIMO interference channels. The adoption of feedforward deep neural networks [11] and convolutional neural networks (CNN) [12] are studied for joint channel estimation and sensing, respectively. However,
the DNN models generally require generating optimal labelled data for offline training, which limits the application in real-time systems. On the other hand, the reinforcement learning (RL) as another main branch of ML can be a feasible option for real-time decision-making tasks, such as resource allocation in heterogeneous networks [13] and IoT systems [14].

The machine learning algorithms have also been investigated to enhance the beamforming techniques. In [15], the authors leveraged offline learning to identify the vehicle’s optimal beam pair index based on the past beam training data by using support vector machine (SVM) classification. The selection of codewords for analogue beamforming is modelled as an offline SVM multiclass-classification problem in [16], where the training dataset consists of a large number of samples of the millimetre-wave channel. In [17], the authors examine the interaction between codebook design and the performance exhibited by the DNN-based analogue beam selection methods through offline angle-of-arrival (AoA) data. The authors in [18] propose a deep learning scheme for the downlink beamforming optimization to solve the SINR balancing and power minimisation problems. By assuming the channel impulse response (CIR) model, the work [19] proposes an adaptive recursive least squares (RLS) CIR prediction approach to predict future orthogonal frequency division multiplexing (OFDM) block CIR coefficients. In [20], the authors utilize reinforcement learning to achieve a mean-field equilibrium which is intended to optimize the hybrid precoding matrix for the unmanned aerial vehicles (UAV) to ground user links. In [21], deep reinforcement learning (DRL) technique was covered, a DRL-based method called PrecoderNet is proposed to design the digital precoder and analogue combiner for beamforming. Although the precoding techniques have been well studied in the above works, significant beam misalignments will occur frequently due to the mobility. This problem becomes particularly challenging for mmWave vehicle communications, especially with multiple users and inter-cell interference.

The Multi-Armed Bandit (MAB) online learning algorithms are used in [22, 23] for beam alignment for the mmWave link with a transmitter and a receiver. Beam switching architectures for mmWave vehicle communications have been studied in [24] to reduce the amount of re-pointing required by leveraging position prediction. The authors in [25] proposed a MAB online learning scheme that allows the base station/access point to explore, learn from, and adapt to their surroundings autonomously to select multiple beams for multiple users simultaneously based on the available contextual information. The above works focus on the single-cell situation. For multi-cell mmWave vehicle communications, the increased complexity will stem from the additional learning required for inter-cell interference management. The supervised SVM can be employed to efficiently handle interference classification, but the offline learning may not be able to adapt to environmental change.

The main contribution of this paper is to introduce SVM-based active learning for interference-aware multi-cell mmWave vehicle communications. In particular, the proposed method starts with aggregating the traffic patterns data, followed by an adaptive beam selection process that uses a fuzzy support vector machine active learning algorithm to adaptively refine the selection scheme in each cell. Our empirical and synthetic studies show that the proposed method performs better than the existing MAB learning algorithm, where the beams are greedily picked with the highest upper confidence bound. In addition, we demonstrate that the proposed method provides robust performance for initialization condition in adaptive learning. Note that, the desirable property of the proposed SVM-based active learning algorithm comes from the fact that the adaptive beam selection problem in each step can be transformed to a dual convex optimization problem and is primarily scaled by the size of the training data, rather than the dimensionality of the data vector. Therefore, the proposed algorithm not only uses the traffic pattern information but also offers an explicitly defined unique optimum that is easily solvable by most software. Consequently, the SVM-based active learning method is particularly suitable for the multi-cell mmWave beam selection with interference awareness.

In order to use the aggregated data from the historic traffic patterns, we introduce the exploration-or-exploitation learning strategy to accelerate the beam selection process. Particularly, if two traffic patterns share some commonality, they can be treated to use the identical beam selection strategies. In this paper, the similarity is measured by the cosine similarity of the vectorized traffic pattern matrices. The basic intuition is that the SVM-based active learning algorithm is performed for every similar traffic patterns. We label these results according to their relative performance and store the data into the database. Consequently, the exploration-or-exploitation learning scheme can exploit the beam selection schemes when with a certain similarity or explores new schemes for unrecorded patterns, respectively. We evaluate several hyperparameter tuning (HT) methods for the proposed model and employ the cross-validation (CV) technique to select the optimal parameters for the proposed SVM-based active learning model. Finally, numerical simulations are performed to evaluate the performance of the proposed algorithm and the impacts of hyperparameters. We find that the proposed algorithm with Bayesian hyperparameter tuning can achieve near-optimal performance with low computational complexity.

The rest of the paper is organized as follows: In Section 2, the system model is described. In Section 3, we introduce the SVM-based adaptive beam selection method. In Section 4, we present the online learning algorithm by considering the similarity of dynamic traffic patterns. Section 5 discusses the hyperparameters tuning methods. In Section 6, the simulations are constructed to evaluate the performance and the impacts of hyperparameters. Finally, we conclude the paper in Section 7.

2 SYSTEM MODEL AND PROBLEM FORMULATION

2.1 MmWave vehicle communications networks

Consider a mmWave vehicle communications network where the passing vehicle wants to communicate with the
Problem formulation

millimetre-wave base stations (mmBS) over mmWave links. MMWave requires beamforming to compensate for the high path loss. As illustrated in Figure 1, the mmBS is equipped with the steerable directional antenna array and can use a finite set of distinct, orthogonal beams to cover the road in the cell [26]. For instance, the similar beam switching architectures for mmWave vehicle communications have been studied in [24] to reduce the amount of repointing required by leveraging position prediction. Also, the MAC layer of IEEE 802.11ad introduces the virtual antenna sectors, discretizing the azimuth plane based on the antenna beam-width [27]. The mobile vehicles are equipped with both LTE and mmWave interfaces [28] to guarantee the connection to the mmBSs and obtain the possible high-speed transmission services in a millimetre wave base station (mmBS) cell. For instance, the large file intermittent download service is considered. The GPS information can be gathered through the LTE interface to get the direction of arrival (DoA) information [28]. The inter-cell interference is considered for the mmWave vehicle communications.

Assuming there are $C$ adjacent mmWave cells in a certain region and the $c$-th cell is denoted as $c \in C = \{1, ..., C\}$, the mmBS can use a limited set $B$ of beams. The transmit power of the mmBS is denoted as $P$, but at a time, the mmBS just can use a subset of $K$ beams, where $K \in \mathbb{N}$, $K < |B|$ is a fixed value which is related to hardware constraint. This limitation could be imposed by the mmWave beamforming technique, the hardware characteristics and cost constraints etc. Then, a mmBS can serve at most $K$ vehicles/user equipments (UEs) simultaneously. The mmBS updates its beam selection in regular time periods. We use $U^c_t$ to represent the current number of UEs in cell $c$ at time $t$.

Hence, at time $t$, there are in total $U = \sum_{c=1}^{C} U^c_t$ UEs/vehicles in $C$ cells. Let $\hat{u}_{c,m}, c \in C, m \in U$, where $U = \{1, ..., U\}$ represent the $m$-th UE in cell $c$ at time $t$. Define $M_t = \max_{c \in C} U^c_t$ as the largest number of users of all cells at time period $t$.

We construct a traffic status/pattern matrix $S_t = [s_1^c, ..., s_C^c]$, where every column $s_i^c = [t_{c,m}^1, ..., t_{c,m}^C]^T$, $c \in C$ for $c \in C$ at time $t$. We use $\hat{t}_{c,m}^c \in C, m \in M_t$ to represent location information of UE $\hat{u}_{c,m}$. At time $t$, for the cells whose number of UEs is smaller than $M_t$, the corresponding elements in $S$ is set to 0.

In each period $t \in \{1, ..., T\}$, the purpose of all mmBSs is to maximise the amount of expected received data at the vehicles over all periods. We construct a beam selection indication (BSI) matrix as $B_t = [b_1^c, ..., b_C^c]^T$ where every column $b_i^c = [b_{c,m}^1, ..., b_{c,m}^C]^T, c \in C$ represents the beam selection scheme for cell $c \in C$ at time $t$. The element of $b_i^c$ takes the value of 1 or 0, which means the user is allocated for the beam or not. Note that, $b_{c,m}^i = 0$ if $u_{c,m}^i < 1 \leq M_t$. The matrix $B_t$ is introduced for convenient calculation in SVM learning. Define $\Omega_{c}$ as the space of all $B_t$.

Given a BSI matrix, the path loss of link between cell $r$ and UE $\hat{u}_{c,m}^r$ is denoted as $\alpha_{c,m}^r$ and the antenna gain is denoted as $\beta_{c,m}^r$. Thus, the interference power for user $\hat{u}_{c,m}^r$ can be calculated as

$$I_{\hat{u}_{c,m}^r} = \sum_{c \in C, c \neq r} P \cdot \beta_{c,m}^r \cdot \alpha_{c,m}^r \cdot \hat{u}_{c,m}^r.$$

The noise power is denoted as $N = n_0 W$ where $W$ represents the bandwidth and $n_0$ is the power spectral density of the noise. Then, we can get the $\text{SNR}$ of UE $\hat{u}_{c,m}^r$:

$$\text{SNR}_{c,m}^r = \frac{P \cdot \beta_{c,m}^r \cdot \alpha_{c,m}^r \cdot \hat{u}_{c,m}^r}{I_{\hat{u}_{c,m}^r} + N} = \frac{P \cdot \beta_{c,m}^r \cdot \alpha_{c,m}^r \cdot \hat{u}_{c,m}^r}{\sum_{c \in C, c \neq r} P \cdot \beta_{c,m}^r \cdot \alpha_{c,m}^r \cdot \hat{u}_{c,m}^r + n_0 W}.$$

In general, the $\text{SNR}$ of a vehicle using a beam during one period is a random variable that depends on the environment of the mmBS (e.g. blockages etc) and the inter-and-intra cell interference. Then, the average data rate of the vehicle in period $t$ is

$$r_{c,m}^t = \frac{W \cdot \log_2(1 + \text{SNR}_{c,m}^r)}{U_t}.$$

In the above equation, the expectation is taken with respect to the randomness of blockages and the interference. We call the random variable $r_{c,m}^t$ as the beam performance. The main notations of this paper are summarized in Table 1.

### 2.2 Problem formulation

Given an arbitrary sequence of vehicle arrivals with beam selection strategies $B_t, t \in T$, the total expected data rate of the
vehicles in period $t$ can be calculated as,

$$
R(T) = \sum_{i \in T} \sum_{c \in C} \sum_{m \in M_{c}} B_{i,c}^{c} R_{c,\text{vec}}^{*}
$$

(1)

The mmBSs aims at selecting a subset of beams which maximises the total expected data rate for all vehicles, that is, maximises the sum of the expected beam performances. We assume that the mmBSs are unaware of its surroundings, and interference. Hence, the mmBSs should learn over time the best subset of beams through aggregated data. This will significantly reduce the channel estimation overhead and access delay for highly time-varying traffics, thus, will be more suitable to the future time-sensitive vehicle services.

In the following, we will propose a data-driven beam selection method for mmWave base stations to allow the base stations to perform the beam selection autonomously by ensuring a dynamic balance between exploration and exploitation through SVM-based adaptively learning, that is, in each round, the algorithm can pick the action based on the observed information rate, and the learning model for the possible interference and blockages. The logic to solve the problem is illustrated in Figure 2.

### TABLE 1 Notations used in the system model

| Notation | Description |
|----------|-------------|
| $B_{i}$  | Beam selection matrix at time $t$ |
| $\theta_{i,m}$ | The beam selection indicator for cell $i$ at time $t$ |
| $M_{c}$  | The maximum number of UEs in a single cell at time $t$ |
| $U_{m}$  | The number of UEs in all cells at time $t$ |
| $U_{i}^{c}$ | The number of UEs in cell $i$ at time $t$ |
| $\alpha_{i,m}^{c}$ | $m$-th UE in cell $i$ and at time $t$ |
| $\beta_{i,m}^{c}$ | The path loss of link between cell $i$ and UE $u_{i,m}$ |
| $\gamma_{i,m}^{c}$ | Antenna gain of link between cell $i$ and UE $u_{i,m}$ |
| $I_{tc}$  | Interference for UE $u_{i,m}$ |
| $\xi_{i,m}$ | Interference for UE $u_{i,m}$ |
| $\gamma$  | Parameter $\gamma$ of RBF kernel |
| $C$      | Parameter $C$ of RBF kernel |
| $D_{i}$  | Beam allocation scheme space at time $t$, $D_{i} = \text{vec} (\Omega_{i})$ |
| $\text{iter}$ | The iteration count of SBOS Algorithm |
| $\text{ITER}$ | Total learning iterations/periods |

3 | ADAPTIVELY SVM-BASED MULTI-CELL INTERFERENCE CLASSIFIER

In this section, we propose an SVM-based learning model with multi-cell interference awareness to reduce the beam selection space gradually for a certain traffic pattern. Specifically, each subsequent beam selection strategy is customized to a certain traffic pattern, while accounting for possible interference. The resulting cost function turns out to be equivalent to an SVM for regression. The performances of the beam selection schemes for all traffic patterns will be stored into the database for the next exploration-or-exploitation learning phase.

3.1 The data gathering and samples labelling

Given a BSI matrix or a beam selection strategy, the vehicle is able to measure beam performance then report it back to the associated mmBS. Also, the LTE BSs periodically record the locations of the vehicles/users into the database. Recall that, we have denoted the beam selection scheme for a cell $i$ at time $t$ as a vector

$$
\mathbf{b}_{i}^{t} = [b_{i,1}^{t}, ..., b_{i,M_{i}}^{t}]'
$$

where $b_{i,m}^{t} \in \{0, 1\}$, $m \in M_{i}$. For example, $\mathbf{b}_{i}^{t} = [1, 1, 0, 1]'$ means that there are four vehicle/UEs in cell $i$ at time $t$ and the mmBS selects the beam to serve UEs 1, 2 and 4.

We define $\text{vec}()$ as a zero-complementation and vectorization function, which means that $\text{vec}(B_{j})$ complements the matrix $B_{j}$ with a size of $C \times M_{j}$ to a $C \times M_{j}$ matrix with zeroes and then vectorize it, where $M_{i}$ is the greatest value of $M_{i}$ in the database. Further, let $B_{j}$ be a BSI matrix and $\mathbf{x}_{i}^{j} = \text{vec}(B_{j}), i \in I = \{1, \ldots, I\}$ for $i$-th beam selection scheme sample, where $I$ is the number of scheme samples. Define $\mathbf{y}_{i}^{j}$ as the label of the sample scheme vector $\mathbf{x}_{i}^{j}$. For simplicity, we denote $\{\mathbf{x}_{i}^{j}, \mathbf{y}_{i}^{j}\}_{i=1}^{I}$ as $\{\mathbf{x}_{i}^{j}, \mathbf{y}_{i}^{j}\}_{i=1}^{I}$.

At time $t$, let $D_{t}$ be the beam allocation scheme space of all $\text{vec}(B_{j})$. Then, any $\mathbf{x}_{i}^{j} \in D_{t}$ represents a beam selection scheme sample. In particular, in the first iteration, the algorithm will generate $l$ samples to form the initial scheme set $D_{0}$ and label these $l$ schemes according to their individual performance, that is, “1” for those schemes whose performance is above the average performance and “-1” vice versa. One major advantage of labelling the sample strategy schemes is that such information can be leveraged to construct the pool of candidate strategy profiles to be evaluated for the consideration of exploitation or exploration in future stages. The rationale for having the majority of profiles in the candidate pool labelling with “1” and “-1”
3.2 SVM-based learning model and its solution

The formulation of SVM model can be viewed as a classification problem that results from the Vapnik Chervonenkis (VC) theory which states that to minimise the predict error, a balance should be taken between minimising the training error and the model complexity [29]. Based on the beam allocation schemes \( \{ x'_i \} \) and the corresponding labels \( \{ y'_i \} \), \( y'_i \in \{-1,1\} \) respectively, the formulation of an SVM classifier is,

\[
\min \frac{1}{2} \| w \|^2 + C \sum \xi_i, \quad \text{s.t.} \quad y'_i (\langle w, x'_i \rangle + b) \geq 1 - \xi_i, \quad i = 1, 2, ..., l, \quad \xi_i \geq 0, \quad i = 1, 2, ..., l.
\]

(2)

where \( w \) is the weight vector, \( b \) is the bias value and \( \xi \) is the slack variable. \( C \) is a hyperparameter to control the relative weight of maximising the margin, and ensuring that most samples have a least margin to the hyperplane. Generally, the training data may not be totally separable, thus, the slack variables \( \xi_i \geq 0, \forall i = 1, 2, ..., l \) is introduced in the formulation.

The training data samples that are nearest to the decision boundary are called support vectors and the distance is given by \( \frac{1}{\| w \|} \). The idea behind the SVM formulation is to maximise the margin which equals to \( \frac{2}{\| w \|} \). Obviously, such an optimization problem is equivalent to minimising \( \| w \| \). Generally, the solution to equation (2) will construct a hyperplane

\[
\theta(x) = w^T x + b = 0,
\]

which separates the data into two half-spaces with the widest margin. A classification rule induced by \( \theta(x) \) for samples \( x'_i \) is denoted as

\[
\text{sign}(w^T x'_i + b).
\]

However, for the problems where the dimension of vector \( x \) is high, the classifier problem (2) may become complicated to be solved. Toward this issue, we can introduce the dual problem of (2) in problem (3), and by applying the kernel trick, the original feature space can be mapped into a higher-dimensional feature space where the training set could become more separable [30].

\[
\max \sum \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y'_i y'_j \langle \phi(x'_i), \phi(x'_j) \rangle
\]

(3)

\[ s.t. \quad 0 \leq \alpha_i \leq C, i = 1, ..., l, \quad \sum_{i=1}^l \alpha_i y'_i = 0. \]

where \( \alpha_i \), \( i \in \{1, ..., l\} \) are the Lagrange multipliers.

In Equation (3), the linear kernel function \( \mathcal{K}(x'_i, x'_j) = \langle \phi(x'_i), \phi(x'_j) \rangle = x'_i^T x'_j \) is employed by default. However, various types of kernel functions can also be adopted, for instance, the Polynomial Kernel Function and the Gaussian Kernel Function (Radial Basis Function), which are given, respectively, in the following:

\[
\mathcal{K}(x'_i, x'_j) = (1 + x'_i^T x'_j)^p,
\]

(4)

\[
\mathcal{K}(x'_i, x'_j) = e^{-\frac{\|x'_i - x'_j\|^2}{\sigma^2}} = e^{-\gamma \|x_i - x_j\|^2},
\]

where \( p \geq 2 \) is the polynomial power and \( \sigma^2 \) is the variance.

In this paper, we consider to use the Gaussian kernel (Equation (4)), and then, the hyperplane achieved by solving the dual problem (3) can be obtained as,

\[
b(x) = \sum_{i=1}^l \alpha_i y'_i e^{-\gamma \|x_i - x'_i\|^2} + b = 0,
\]

(5)

where \( \alpha_i \) is the solution to problem (3).

Based on the boundaries identified by the hyperplane, we propose the beam selection scheme pick algorithm. For the sake of simplicity, the algorithm is named as Picker algorithm and is presented in Algorithm 1. Let the sample set \( X' \) denote the set containing all the labelled samples; \( b_{\text{set}}' \) denote the boundary set. Then, the sample space \( D_i \) can be reduced by Picker algorithm because it picks out the samples with the maximum-minimal distance to \( X' \) in \( D_i \) constrained by the boundary set.
In other words, the observed information rate provides constraint(s) that makes the feasible space of beam selection schemes smaller.

We denote the process of the SVM-base learning as a function $\text{SVM}(\{\mathbf{x}_j\}, \{y_j\}, \text{parameters})$ where $\{\mathbf{x}_j\}, \{y_j\}$ are beam selection schemes and the corresponding labels. Particularly, parameters are the well tuned hyperparameters which will be discussed in detail in Section 5. By the way, the classical SVM can be easily solved by many softwares. For example, the libsvm package [31] for MATLAB and the MATLAB built-in fitcsvm package can be used to solve the dual problem (3) to get the boundaries or hyperplanes.

4 SVM-BASED BEAM ONLINE SELECTION ALGORITHM WITH MULTI-CELL INTERFERENCES

Conditional on the candidate pool of beam selection strategy profiles obtained for certain traffic pattern, this section proposes the SVM-based beam online selection (SBOS) algorithm which accounts for all possible traffic patterns. We first define the similarity of traffic pattern $\mathbf{S}_t$ and $\mathbf{S}_{t'}$ as $s(t, t')$ in the following equation:

$$s(t, t') = \frac{\text{vec}(\mathbf{S}_t) \cdot \text{vec}(\mathbf{S}_{t'})}{\|\text{vec}(\mathbf{S}_t)\| \cdot \|\text{vec}(\mathbf{S}_{t'})\|}$$

(6)

where $\mathbf{S}_t$ and $\mathbf{S}_{t'}$ are status matrices as defined in 2.2, $\text{vec}(\mathbf{S}_t)$ represents the complemented and vectorized $\mathbf{S}_t$. Equation (6) measures the degree of the cosine similarity between $\text{vec}(\mathbf{S}_t)$ and $\text{vec}(\mathbf{S}_{t'})$. We set $\epsilon$ as the threshold to judge traffic pattern similarity.

Recall that we need to gather $l$ samples and their labels in SVM-based learning model. Before going into the feedback gathering stage, we set a flag $\text{gath}$ as True and a flag $\text{gath_f}$ as False. Every time we encounter the similar traffic pattern identified by the similarity threshold $\rho$, we can apply one of these $l$ beam selection schemes generated by Picker algorithm, and then record the average information rate. Until all of these schemes are applied, the algorithm will set $\text{gath}$ as False and $\text{gath_f}$ as True.

1. For the traffic pattern generated at time $t$, we record the traffic pattern information by $\mathbf{S}_t$. Next, we check whether there is a record for the similar traffic pattern in the database. If there is, we reload the last learning data (line 2-3 in Algorithm 2). If the traffic pattern in the database is not the same, we initialize the learning (line 4-8 in Algorithm 2) and enter the exploration or exploitation stage depending on the learning records (line 11-15 in Algorithm 2). In either of these two stages, the algorithm will generate $l$ samples to form the initial scheme set $\mathbf{S}_{t0}$.

2. Gather the performance of the schemes and label these schemes in order to enable the SVM model. For example, in the second iteration, after the feedback gathering stage (line 16-17 in Algorithm 2), the labelling function is called to label these samples (line 18-19 in Algorithm 2) based on the cellular system feedback as introduced in 3.1. Then, after hyperparameter tuning, cross-validation and training, the SVM model learns from these samples. A decision boundary $h_{\text{iter}}(x) = 0$ will be generated from the data set.

**Algorithm 2: SBOS algorithm**

| Line | Description |
|------|-------------|
| 1 | Input: $T, \mathbf{S}_t$, $l \in T, \text{ITER}, l$ |
| 2 | for each $t \in T$ do |
| 3 | if $\mathbf{S}_t$ makes $s(t, t') \geq \rho$ (6) then |
| 4 | otherwise |
| 5 | $\text{iter} = 1, \text{rec} = 0; // set counters$ |
| 6 | $\text{hset}_t \leftarrow \{\}; // clear boundary set$ |
| 7 | $D_t \leftarrow \mathbf{S}_t; // build scheme space from $ |
| 8 | $\text{gath}, \text{gath_f} = 0 // flags$ |
| 9 | if $\text{iter} = 1$ then |
| 10 | if $\text{gath} = 0$ and $\text{gath_f} = 0$ then |
| 11 | /* Exploration. */ |
| 12 | randomly generate $\{x'_i\}_l$ from the preferred scheme set of $\mathbf{S}_t$; |
| 13 | else |
| 14 | /* Exploration. */ |
| 15 | $\{x'_i\}_l = \text{Picker}(\text{hset}, D_t, l, \{\});$ |
| 16 | $\text{gath} = 1, \text{gath_f} = 0$ |
| 17 | else if $\text{gath} = 1$ and $\text{gath_f} = 0$ then |
| 18 | /* get information rate; */ |
| 19 | $\{y'_i\}_l = \text{Label}(\{x'_i\}_l);$ |
| 20 | $\text{iter} = \text{iter} + 1;$ |
| 21 | $\text{gath}, \text{gath_f} = 0$ |
| 22 | else if $\text{iter} \leq \text{ITER}$ then |
| 23 | if $\text{gath} = 0$ and $\text{gath_f} = 0$ then |
| 24 | /* EXPLOITATION */ |
| 25 | $\text{hset}_t = h_{\text{iter} - 1}(x);$ |
| 26 | $\text{rec} = 0;$ |
| 27 | else |
| 28 | $\text{rec} = \text{rec} + 1;$ |
| 29 | /* HT & CV; */ |
| 30 | $h_{\text{iter}}(x) = \text{SVM}(\{x'_i\}_l, \{y'_i\}_l, \text{parameters});$ |
| 31 | $\text{Picker}(\text{hset}_t, h_{\text{iter}}(x), D_t, l - m, X^t);$ |
| 32 | /* Check termination criteria; */ |
| 33 | $\text{gath} = 1, \text{gath_f} = 0$ |
| 34 | else if $\text{gath} = 1$ then |
| 35 | /* get information rate; */ |
| 36 | $\{y'_i\}_l = \text{Label}(\{x'_i\}_l);$ |
| 37 | $\text{iter} = \text{iter} + 1;$ |
| 38 | $\text{gath}, \text{gath_f} = 0$ |
SBOS algorithm is mainly based on SVM algorithm; therefore, its computational complexity is mainly dominated by the computational complexity of the SVM algorithm. In this paper, SVM algorithm is implemented by libsvm which consists of two parts (svm_train and svm_predict) and procedures including svm_save_model and svm_load_model etc. According to [32], both svm_train and svm_predict have the highest complexity of \( O(l^3) \), so the complexity of SBOS algorithm is \( O(l^3) \) where \( l \) is the number of input data in problem (2), that is, pairs of beam allocation schemes and corresponding labels. Recall that the termination criteria of SBOS algorithm are that the Picker (Algorithm 1) cannot pick out new samples or reaches the maximum iteration count ITER. So, in the worst case, SBOS algorithm will execute ITER times for a specific traffic pattern. Because ITER is a constant, so the computational complex of SBOS algorithm is \( O(l^3) \). Note that the SVM algorithm is performed only for different traffic patterns, and \( l \) will be not large and can be set in the region of [6,12] in our setting.

5 | HYPERPARAMETERS ADJUSTMENT FOR SBOS ALGORITHM

Note that there are two hyperparameters that need to be tuned in the algorithm, that is, the \( C \) and \( \gamma \), which will impact the performance of the algorithm a lot. In machine learning, hyperparameter optimization or hyperparameter tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm [33]. Generally, the same kind of machine learning model can generalize different results when adopting different hyperparameters. In this section, we will discuss how to tune the hyperparameters for the algorithm. In particular, we first introduce the widely used hyperparameter adjustment methods, Grid search, Random search, Bayesian optimization, and then, discuss how to use cross-validation to tuning the parameters.

5.1 | Parameter tuning methods

According to problem (3) and Equation (4) in the SVM model, let \( \gamma = (C, \gamma) \). Define a hyperparameter optimization problem in the following:

\[
    f(\gamma) = \max_w \mathcal{T}(w | \{(x_{t}, y_t)\}_{t}, \gamma),
\]

where \( \mathcal{T} \) is the cross-validation accuracy of SVM mode when applying hyperparameters \( \gamma \), \( w \) is the weight vector introduced in problem (2) and \( \{(x_{t}, y_t)\}_{t} \) is the cross-validation beam selection scheme sample set, that is, \( X_t \) in the Algorithm 2. Each specific pair of these two hyperparameter is denoted as \( \gamma_i = (C_i, \gamma_i) \), where \( i \) denotes the hyperparameter adjustment step.

The objective of the iterative global optimization of a function \( f : \mathcal{D} \to \mathbb{R} \) is to find the sequence of points

\[
    \gamma_1, \ldots, \gamma_a \in \mathcal{D} = [C_{\min}, C_{\max}] \times [y_{\min}, y_{\max}]
\]
that converges to the optimal \( f^* \), \( f^*(\hat{c}) = \sup_{x \in X} f^*(x) \). In other words, we can approximate the value of \( f \) from previous steps [34].

### 5.1.1 Grid search

Grid search is simply an exhaustive searching through a manually specified subset of the hyperparameter space. Generally, a grid search algorithm must be evaluated according to certain performance metrics, that is, the cross-validation on the training dataset [35] or on a held-out validation dataset.

### 5.1.2 Random search

Random search selects the hyperparameters randomly in the parameter space instead of exhaustive enumeration of all possible values. Sometimes, it can outperform the Grid search, especially when only a small number of hyperparameters affects the final performance of the machine learning algorithm. In this case, the optimization problem is said to have a low intrinsic dimensionality. Additionally, Random search can use the prior knowledge by specifying the distribution of the samples [36].

### 5.1.3 Bayesian optimization

Bayesian optimization builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set. Bayesian optimization tries to gather observations revealing as much information as possible about this function and in particular, the location of the optimum by iteratively evaluating a promising hyperparameter configuration based on the current model, and then updating it.

The main idea behind Bayesian optimization is to use all of the information gathered in previous iterations for performing the next step in order to minimise the number of classifier trainings. In other words, we can solve the optimization problem directly by performing an adaptive process that on one hand tries to maximise the objective function and on the other hand samples the possible \( \hat{c} \) space.

For Bayesian optimization, if the exact form of \( f \) is known, then the optimization procedure would be much simpler. Unfortunately, \( f \) is a black-box function with a very complex structure, expensive even to evaluate. However, some simplifying assumptions for \( f \) might make a problem solvable. Assume that \( f \) can be represented as a sample from a probability distribution over a family of functions \( f \sim P(f), f \in \mathcal{F} \). The expectation over the loss function \( \delta_n \) can be expressed as,

\[
\min_{\hat{f}} \mathbb{E}_{P(f)}[\delta_n(\hat{f})] = \min_{\hat{f}} \int_{\mathcal{F}} \delta_n(\hat{f})dP(f),
\]

where \( \delta_n(\hat{f}) = f^*(\hat{c}) - f^*(\hat{c}_n) \), and \( \hat{c} \) is the optimal hyperparameters \( \{C^*, \gamma^*\} \).

Given additional assumptions about the prior distribution of \( \mathbf{P} \), very efficient solutions for the entire process can be provided. For example, assuming that the target function \( f \) is a sample of the Gaussian stochastic process. A crucial advantage of such a simplification is that manipulating the distribution over such functions will be relatively easy, in particular, the posterior distribution is also a Gaussian process. Consequently, in each iteration a calculated posterior can be used as an informative prior for the next iteration, creating a relatively simple iterative procedure.

### 5.2 Implementation of hyperparameters tuning

For the SVM-based learning, by adopting different kernel functions, we can apply the “kernel trick” to map the original feature space to a higher-dimensional feature space to reduce the complexity of the SVM problem. Here, we choose kernel function \( K(x_i, x_j) \) (Equation (4)). A soft-margin SVM classifier with a RBF kernel has at least two hyperparameters that need to be tuned, that is, a regularization constant \( C \) and a kernel hyperparameter \( \gamma \). Here, \( \gamma \) is a critical parameter. A large \( \gamma \) leads to a high bias and low variance model, and vice versa. Another parameter \( C \) is used for the soft margin cost function, which controls the influence of each individual support vector. A large \( C \) may cause overfitting and a small \( C \) would cause underfitting. In a word, tuning hyperparameters \( \gamma \) and \( C \) is important for achieving the good performance of SBOS algorithm.

According to empirical studies, both \( \gamma \) and \( C \) can be found within the set \( \{2^{-10}, ..., 2^{10}\} \) and we will find the best pair of \( \gamma \) and \( C \) by cross-validation. Cross-validation is one way to ensure the robustness of the model. The basic idea is that a portion of data (called a holdout sample) can be held back. The bulk of the data is trained and the holdout sample is used to test the model. This is different from the classical method of model testing, which uses all of the data to test the model. By using the cross-validation, we can make full use of the limited training data set. After performing the parameter search on training data set, we finally determine the best pair of parameters denoted as \( \bar{\gamma}, \bar{C} \).

### 6 SIMULATION AND RESULTS

In this section, we evaluate the performance of proposed SVM-based beam online selection algorithm with multi-cell interferences. First, the system parameters and channel model for simulation settings are given. Then, we consider various factors that may affect the performance of the proposed algorithm and compare it with some typical schemes.

#### 6.1 Simulation setup

This study uses the powerful SVM tool package libsvm [31] and the fitcsvm package of MATLAB. The platform we used is an
TABLE 2  Major parameters used in simulation

| Parameter          | Value                  |
|--------------------|------------------------|
| Carrier frequency  | 28 GHz                 |
| System bandwidth   | 1 GHz                  |
| Transmit power     | 30 dBm                 |
| Thermal noise      | $-174 \text{ dBm/Hz}$  |
| Path loss model (dB)| $32.4 + 17.3 \log_{10} d(m) + 20 \log_{10} f_c(\text{GHz}) + \zeta$  |
|                    | $\zeta \sim \mathcal{N}(0, \sigma)$ |

HP Laptop 14s-dp0xxx, AMD 3700U 4-Core 8-Thread, 16GB RAM and the software is Windows 10 amd64 Home Ed. v2004, MATLAB 2020a. In this simulation, we deploy 4 mmWave cells, and we set $l = 5$, $\text{ITER} = 14$. The channel parameters used in this simulation are shown in Table 2. We consider two kinds of blockages: temporary blockage and permanent blockage.

6.2  Performance analysis

In the simulations, we provide a thorough performance analysis by comparing the proposed SBOS Algorithm to the following methods:

- Optimal. The optimal method is assumed to have the full prior knowledge of the multi-cell interference for all traffic patterns. Thus, it is able to choose the best subset of beams and provide the upper bounder of the beam performance for the current traffic pattern.
- UCB. The MAB model has been widely adopted for studying many network problems with unknown parameters, that is, resource allocation, crowdsourcing etc. The classical version of the Multi-armed bandit problem is formulated as a system of $n$ arms (or actions), each having an unknown distribution of rewards. One kind of purpose of MAB is to maximize the reward accumulated by the agent while playing the arms. This objective highlights the balance between staying with the arm that gave highest reward in the past and exploring new arms that might give higher reward in the future. UCB is one of the popular algorithms that can balance the exploitation and exploration in such bandit problems [37, 38]. Specifically, in our simulation, UCB selects the arms/beams, with the highest estimated upper confidence bounds based on the history data. However, when UCB has no experience data, or in other words, it has no idea of the environment and the settings of arms, it will choose the arm randomly.
- Random. This algorithm selects random beams based on a uniform random distribution.

6.2.1  Performance comparison with a certain traffic pattern

Figure 5 shows the average information rate at each iteration with the different parameter REC. The REC is set to 3 and 5, respectively. We can see from the figure that by using the proposed algorithm, we can get near-optimal average rate of the whole system after 12 iterations. Additionally, it can be observed that when REC = 3, the algorithm is more stable but achieves a lower performance compared with the situation of REC = 5. This is due to the reason that when REC = 3, Picker algorithm uses the historical boundary set every 3 iterations and is more likely to exploit the existing information to generate new scheme set. While for REC = 5, Picker algorithm trends to explore the new generated boundary set to generate new scheme set.

Figure 6 illustrates the performance when adopting different beam allocation methods. In the figure, REC is set to 3. It can be observed from the figure that the proposed algorithm would always obtain a better performance in terms of the average information rate compared with the Random and UCB methods. Specifically, the gap between the proposed algorithm and UCB methods could be around 0.3 Gbit/s/vehicle. For instance, at $\text{iter} = 11$, the proposed algorithm obtains the
average information rate of 4.25 Gbit/s/vehicle. While the UCB method obtains the average information rate of 3.9 Gbit/s/vehicle. The gap is about 0.35 Gbit/s/vehicle. Also, the performance of the random algorithm fluctuates rapidly, and the performance of UCB fluctuates slowly. Thus, the proposed algorithm is more stable than the other two methods. Here, we present an intuitive idea to enhance the fairness among different users by using a more complex labelling strategy. For example, when labelling beam selection schemes, the ratio of the worst and best information rates can be used as extra labels. However, theoretical study for the learning scheme to balance the fairness and the system performance is an interesting issue, goes beyond the scope of this article and be left for future work.

Table 3 gives an example to show the change of the information rate of a user/vehicle. We select a user with the lowest information rate initially, and present the rates of the user in several iterations. It is observed from the figure that although the user has the lowest rate at the beginning, its rate will rise quickly due to the movement.

### 6.2.2 Impact of the hyperparameter tuning methods

Figure 7 shows the performance of the proposed algorithm with different HT methods. REC is set to 3 and we apply three HT methods by using two different SVM-related software packages, that is, libsvm and fitcsvm. The figure shows that the proposed algorithm using libsvm package achieves the highest performance after the 12th iteration, and the proposed algorithm using the other three HT schemes terminate at the 8-th iteration, but obtain a lower performance. It also reveals that the proposed algorithm using the Bayesian optimization HT could obtain the relatively good performance after just 8 iterations.

Table 4 lists the results of the proposed algorithm with different HT methods. It shows the number of iterations for the algorithm converges, the average and maximum information rate of schemes achieved by the algorithm, and the running time of the algorithm. We can observe from the table that the Grid Search outperforms the Random Search and runs slower. The reason is that the Grid Search will try every possible pair of hyperparameters, while the Random Search just chooses them randomly. The table also illustrates that the Bayesian HT method outperforms the Random Search and the Grid Search, since the Bayesian HT method can make use of the information gathered in previous steps.

Figure 8 reveals the hyperparameter adjustment process of the Grid Search, the Random Search and the Bayesian optimization method. The figure shows that the Bayesian method outperforms the other two methods in minimising the objective loss function in Equation (8). Also, the Bayesian HT method achieves the minimum loss value of 0.048. As a result, the Bayesian HT method is more efficient than the other two methods.

Corresponding to Figure 8, Figure 9 plots the hyperparameter adjustment process of Bayesian optimization HT method from step 5 to step 11 in detail. We can observe from the

### Table 3  Change of the information rate of the UE

| Item | iteration | 1 | 4 | 7 | 10 |
|------|-----------|---|---|---|----|
| The rate of the user (Gbit/s) | 0.8963 | 2.2843 | 5.8432 | 4.7346 |
| The largest rate (Gbit/s) | 9.1374 | 8.9519 | 8.6363 | 8.5073 |

### Table 4  Performance of SBOS with different HT methods

| HT methods | Grid | Random | Bayesian |
|------------|------|--------|----------|
| Iterations | 8 | 10 | 8 |
| Finalmaximarerate(Gbit/s/vehicle) | 4.365 | 4.315 | 4.395 |
| Finalaveragerate(Gbit/s/vehicle) | 4.240 | 4.240 | 4.3175 |
| Runtime(seconds) | 90 | 50 | 80 |
6.2.3 The multi-cell interference analysis

Figure 10 shows the average interference power of the system which is defined by the total interference of the system divided by the number of cells. We can see from the figure that, at the beginning of the algorithm, the average multi-cell interference is around $-55$ dBm. After 14 iterations, the interference has been reduced to around $-70$ dBm. This is due to the reason that the proposed algorithm finally selects the suitable beam set for the traffic pattern based on the corresponding samples and labels. The figure also reveals that the Bayesian HT method achieves the similar performance compared with the Grid Search HT method implemented by libsvm, but the Bayesian HT method is more stable and faster.

Figure 11 plots the average interference power of each cell, which is defined by the total interference in a cell divided by the number of UEs in it. The figure shows that the proposed algorithm can reduce the average interference power gradually. For example, in cell 4, at the beginning of the algorithm, the average interference is around $-55$ dBm and after 14 iterations it has been reduced to around $-74$ dBm. Also, the average interference power of each cell reduces gradually, and then converges to a stable value for a certain traffic pattern.

6.2.4 Performance analysis of SBOS algorithm with dynamic traffic patterns

Figure 12 analyses the average information rate with dynamic traffic patterns by using the proposed SVM-based beam online selection algorithm. At the beginning, UCB algorithm has no history experience; it will also choose the beams randomly. We
can see from the figure that the initial randomness of the UCB algorithm may lead a better performance at the beginning. However, after \( t = 40 \), the performance of SBOS algorithm is above that of UCB. Also, after \( t = 170 \) iterations, the proposed algorithm approaches to the optimal algorithm, but there is still an obvious gap between the UCB method and the optimal algorithm; this is because the UCB may take a long time to explore all the arms. Moreover, by employing Picker algorithm, SBOS algorithm does not have to try every selection schemes, so that it can converge to the optimal algorithm rapidly.

This also indicates that SBOS algorithm learns the beam selection strategy with multi-cell interference more efficiently than the other two methods. The reason is that SBOS algorithm is based on SVM classifier learning algorithm which can obtain a high classification accuracy even with very limited number of samples. Consequently, when selecting each beam selection strategy based on such samples, the negative impact of response errors in the process of adaptive strategy selection would be alleviated rapidly.

## 7 Conclusion and Future Work

This paper has introduced an SVM-based active learning scheme for interference-aware multi-cell mmWave vehicle communications. In particular, we have used a fuzzy support vector machine active learning algorithm to adaptively refine the selection scheme in each cell. Specifically, if two traffic patterns share some commonality in terms of the cosine similarity of the vectorized traffic pattern matrices, they have been treated to use the identical beam selection strategies and the corresponding results have been labelled and stored into the database. We have labelled these results according to their relative performance and store the data into the database. Consequently, the exploration-or-exploitation learning strategy can exploit the beam selection schemes when with a certain similarity or explore new schemes for unrecorded patterns respectively. Our empirical and synthetic studies have shown that the proposed method performs better than the existing MAB learning algorithm where the beams are greedily picked with the highest upper confidence bound. Moreover, the results have also illustrated that the proposed algorithm with Bayesian hyperparameter tuning can achieve near-optimal performance with low computational complexity. In a nutshell, the proposed method has demonstrated the capability of SVM-based online learning on the data-driven mmWave vehicular communications with multi-cell interference coordination for future 5G/B5G vehicle applications.

An interesting extension of this work is to study the fairness issue among different users both from an algorithmic design perspective and also from a mathematical stand point. Also, it is a challenging and open problem to coordinate the multi-cell beam allocation in a distributed fashion, in order to speed up the learning process. Topics about the attack in the future collaboratively intelligence-enabled networks need to be pondered and are very important as 5G/B5G networks are deployed, especially for vehicular communications.

## Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grants No. 2019YFB2103004 and No. 2020YFB1806608; in part by the National Natural Science Foundation of China under Grants No. 61771252, No. 61772287, and No. 61801240; in part by the Key University Science Research Project of Jiangsu Province under Grants No. 18KJA510004 and No. 18KJA510005; in part by the Jiangsu Province Special Fund Project for Transformation of Scientific and Technological Achievements under Grant No. BA2019058; and in part by the Open Research Fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education.

### ORCID

Jiangpei Zhu 🌐 https://orcid.org/0000-0003-0529-3204

## References

1. Sun, S., et al.: Analytical framework of hybrid beamforming in multi-cell millimeter-wave systems. IEEE Trans. Wirel. Commun. 17(11), 7528–7543 (2018)
2. Vahidi, V., Saberinia, E.: MIMO–OFDM communication systems for traffic data transmission in 5G drone small cells. IET Commun. 13(20), 3565–3574 (2019)
3. Zhu, L., et al.: Millimeter-wave full-duplex uav relay: Joint positioning, beamforming, and power control. IEEE J. Sel. Areas Commun. 38(9), 2057–2073 (2020)
4. Cao, J., Tila, F., Nix, A.: Design and implementation of a wideband channel emulation platform for 5G mmWave vehicular communication. IET Commun. 14(14), 2369–2376 (2020)
5. Coll-Perales, B., Gozalvez, J., Grueter, M.: Sub-6GHz assisted MAC for millimeter wave vehicular communications. IEEE Commun. Mag. 57(3), 125–131 (2019)
6. Li, X., et al.: FFR based joint 3D beamforming interference coordination for multi-cell FD-MIMO downlink transmission systems. IEEE Trans. Veh. Technol. 69(3), 3105–3118 (2020)
7. Li, Y., Cheng, X.: Modelling and simulation for UAV-aided vehicular MIMO communication channels. IET Communications 13(18), 3044–3051 (2017)
8. Cheng, X., Huang, Z., Chen, S.: Vehicular communication channel measurement, modelling, and application for beyond 5G and 6G. IET Commun. 14(19), 3303–3311 (2020)
9. Jameel, F., et al.: Propagation channels for mmWave vehicular communications: State-of-the-art and future research Directions. IEEE Wirel. Commun. 26(1), 144–150 (2019)
10. O’Shea, T., Hoydis, J.: An introduction to deep learning for the physical layer. IEEE Trans. Cogn. Commun. Netw. 3(4), 563–575 (2017)
11. Ye, H., Li, G.Y., Juang, B.H.: Power of deep learning for channel estimation and signal detection in OFDM systems. IEEE Wirel. Commun. Lett. 7(1), 114–117 (2018)
12. Wen, G.K., Shih, W.T., Jin, S.: Deep learning for massive MIMO CSI feedback. IEEE Wirel. Commun. Lett. 7(5), 748–751 (2018)
13. Wei, Y., et al.: User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach. IEEE Trans. Wirel. Commun. 17(1), 680–692 (2018)
14. Jiang, N., et al.: Reinforcement learning for real-time optimization in NB-IoT networks. IEEE J. Sel. Areas Commun. 37(6), 1424–1440 (2019)
15. Wang, Y., et al.: MmWave vehicular beam selection with situational awareness using machine learning. IEEE Access 7, 87470–87493 (2019)
16. Long, Y., et al.: Data-driven-based analog beam selection for hybrid beamforming under mm-wave channels. IEEE J. Sel. Top. Signal Process. 12(2), 340–352 (2018)
17. Antón-Haro, C., Mestre, X.: Learning and data-driven beam selection for mmWave communications: An angle of arrival-based approach. IEEE Access 7, 20404–20415 (2019)
18. Xia, W., et al.: A deep learning framework for optimization of MISO downlink beamforming. IEEE Trans. Commun. 68(3), 1866–1880 (2020)
19. Bogale, T.E., Wang, X., Le, L.B.: Adaptive channel prediction, beamforming and scheduling design for 5G v2I network: Analytical and machine learning approaches. IEEE Trans. Veh. Technol. 69(5), 5055–5067 (2020)
20. Li, L., et al.: Millimeter-wave networking in the sky: A machine learning and mean field game approach for joint beamforming and beam-steering. IEEE Trans. Wirel. Commun. 19(10), 6393–6408 (2020)
21. Wang, Q., et al.: Precodernet: Hybrid beamforming for millimeter wave systems with deep reinforcement learning. IEEE Wirel. Commun. Lett. 9(10), 1677–1681 (2020)
22. Hashemi, M., et al.: Efficient beam alignment in millimeter wave systems using contextual bandits. In: Proceedings of IEEE INFOCOM, pp. 2393–2401.IEEE, Piscataway, NJ (2018)
23. Arkan, I., et al.: MAMBA: A multi-armed bandit framework for beam tracking in millimeter-wave systems. In: Proceedings of IEEE INFOCOM, pp. 1469–1478.IEEE, Piscataway, NJ (2020)
24. Va, V., et al.: Beam design for beam switching based millimeter wave vehicle-to-infrastructure communications. In: Proceedings of IEEE ICC, pp. 1–6.IEEE, Piscataway, NJ (2016)
25. Sim, G.H., et al.: An online context-aware machine learning algorithm for 5G mmWave vehicular communications. IEEE/ACM Trans. Netw. 26(6), 2487–2500 (2018)
26. Zhang, R., et al.: A Closed-Form PS-DFT Codebook Design for mmWave Beam Alignment. In: Proceedings of IEEE ICC, pp. 1–6.IEEE, Piscataway, NJ (2019)
27. IEEE/Standard: Enhancements for very high throughput in the 60 ghz band, std, IEEE 8802-11:2012/Amd3:2014(E), pp. 1–634 (2014)
28. Campos, R.S.: Genetic algorithm-based cellular network optimisation considering positioning applications. IET Commun. 13, 879–891(12) (2019)
29. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
30. Lin, H., Shin, W.Y., Joung, J.: Support vector machine-based transmit antenna allocation for multiuser communication systems. Entropy 21(5), 471 (2019)
31. Chang, C., Lin, C.: LIBSVM: A library for support vector machines. in ACM Trans. Intell. Syst. Technol. (TIST) 2(3), 27 (2011)
32. Abdiansah, A., Wardoyo, R.: Time complexity analysis of support vector machines (SVM) in LibSVM. Int. J. Comput. Appl. 128(3), 28–34 (2015)
33. Czarnecki, W.M., Podlewska, S., Bojarski, A.J.: Robust optimization of SVM hyperparameters in the classification of bioactive compounds. J. Cheminform. 7(1), 38 (2015)
34. Mockus, J: Application of Bayesian approach to numerical methods of global and stochastic optimization. J. Global Optim. 4(4), 347–365 (1994)
35. Hsu, C., Chang, C., Lin, C.: A Practical Guide to Support Vector Classification (2016). Available from: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
36. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. 13(Feb), 281–305 (2012)
37. Notsu, A., et al.: Proposal of Grid Area Search with UCB for Discrete Optimization Problem. In: Huynh, V.N., Inuiguchi, M., Demoeux, T. (eds.) Integrated Uncertainty in Knowledge Modelling and Decision Making, pp. 102–111.Springer International Publishing, Cham (2015)
38. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time Analysis of the Multi-armed Bandit Problem. Mach. Learn. 47(2), 235–256 (2002)

How to cite this article: Li D, Zhu J, Zhao H, Wang X, Jiang R. SVM-based online learning for interference-aware multi-cell mmWave vehicular communications. IET Commun. 2021;15:1015–1027. https://doi.org/10.1049/cmu2.12138