Pervasive and Mobile Computing 98 (2024) 101887

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Optimization of mobility sampling in dynamic networks using predictive wavelet analysis

Peppino Fazio [a,b,*], Miralem Mehic [c,b], Floriano De Rango [d], Mauro Tropea [d], Miroslav Voznak [b]

[a] DSMN, Ca' Foscari University of Venice, Via Torino 155, Mestre, 30172, VE, Italy
[b] VSB – Technical University of Ostrava, 17. listopadu 2172/15, Ostrava, 70800, Czechia
[c] Department of Telecommunications, Faculty of Electrical Engineering, University of Sarajevo, Zmaja od Bosne bb, Sarajevo, 71000, Bosnia and Herzegovina
[d] Department of Computer Science Engineering, Modellistic, Electronic and Systems, University of Calabria, via P.Bucci 39c, Arcavacata di Rende, 87036, CS, Italy

ARTICLE INFO

Keywords:
Mobile networking
Mobility
Frequency domain
Mobility spectrum
Dynamic sampling
Adaptive mobility
Predictive wavelet

ABSTRACT

In the last decade, the investigation of mobility features has gained enormous significance in many scenarios as a result of the significant diffusion and deployment of mobile devices covered by high-speed technologies (e.g., 5G). Many contributions in the literature have attempted to discover mobility properties, but most studies are based on the time features of the mobility process. No study has yet considered the effects of setting a proper sampling frequency (generally set to 1 s), in order to avoid information loss. Following our previous works, we propose a novel predictive spectral approach for mobility sampling based on the concept of a predictive wavelet. With this method, the choice of sampling frequency is governed by the current spectral components of the mobility process and derived from an analysis of future, predicted components. To assess whether our proposal may yield a helpful method, we conducted several simulation campaigns to test sampling accuracy and obtained results that confirmed our expectations.

1. Introduction

With the advent of 5G technology and recent studies that have explored its performance possibilities [1], the overall level of traffic generated within networks has rapidly increased: the natural effect of this phenomenon on 5G networks is the ability to store, analyze and learn from the incoming data and thereby autonomously modify its behavior according to user experience. It is evident that *knowledge discovery* has an essential function in these networks, given that the most repeated operation is data collection, from which *hidden knowledge* that describes interesting features of the observed process can be extracted. The present study investigates the data acquisition process: very few works in the literature have examined the role of suitable sampling frequencies, and most existing studies have focused on this parameter only from the aspect of time [2–5]. To our knowledge, no works in the literature discuss how a process should be "stored". We refer especially to understanding a suitable sampling frequency derived from the analysis of the features of the observed process. We investigate some features of user mobility to form an idea of which sampling frequencies might aid in reducing both energy/space consumption (oversampling) and avoiding the loss of precious hidden information (downsampling). In brief, we are not concerned with proposing a new mobility prediction approach but with

* Corresponding author at: DSMN, Ca' Foscari University of Venice, Via Torino 155, Mestre, 30172, VE, Italy.
E-mail address: peppino.fazio@unive.it (P. Fazio).

https://doi.org/10.1016/j.pmcj.2024.101887
Received 23 September 2023; Received in revised form 11 December 2023; Accepted 20 January 2024
Available online 3 February 2024

presenting a novel method for determining which sampling frequency is more useful in a given mobile environment. We want to emphasize that if the chosen frequency is lower than the right one (unknown if the mobility process is not analyzed adequately), the collected information may leak some precious features applicable to the network to ''understand'' mobile hosts behavior during the time. Conversely, if the chosen frequency is higher than the right one, unneeded data is stored in the network systems, causing unwanted overhead and space wastage. The ability to predict user movements benefits an entire range of mobile wireless systems, including location-based applications, mobile access control, Quality of Service/Experience (QoS/E) provisioning, resource allocation/management, urban planning, epidemic control, location-based services, and intelligent transportation management [6]. Our proposal provides a valuable approach to change the mobility sampling frequency time-by-time, following the current behavior of mobile hosts, not only the mobile opportunities that the mobile environments give (in terms of maximum speeds, road topology, etc.). Specifically, we propose a novel predictive method of dynamically changing the sampling period according to the considered mobility pattern's current and future spectral content. We are proposing a new predictive wavelet approach by integrating our previous contribution of [7] with the signal processing theory related to AD-PCM analog to digital conversion (mostly used in audio applications). The approach in [7] is based only on the current knowledge of the mobility process, while here, we are adding the possibility of knowing which spectral content will be present in the near future into the mobility process so that the sampling frequency can be chosen more stably. So, starting from the wavelet-based approach of [7], we follow the same theory used in Adaptive Differential Pulse Code Modulation (ADPCM) [8]. The spectral content of a mobility pattern in a given observation window is analyzed using wavelets. Our proposed algorithm, Smoothed Predictive Spectral WAVElet (SPS-WAVE), can dynamically select a suitable sampling frequency/ period according to the wavelet coefficients from previous samples and future predicted spectral values. Several predictors based on different optimization algorithms (such as Recursive Least Squares — RLS, Least Mean Squares — LMS, Least Squares — LS) were considered to allow their accuracy to be compared in terms of prediction and overall sampling performance. The SPS-WAVE is based on wavelet theory [9–11] and allows inspection of the different spectral components that compose the considered mobility trace.

From our preliminary results in [7,12,13] in which we introduced spectral mobility analysis (in terms of autocorrelation and prediction models), we continue with our main aims, as follows:

- The literature contains many contributions that have highlighted the importance of selecting a proper sampling frequency [14, 15], especially in environments where high GPS accuracy is required (loss of granularity in positioning sampling is undesired) but too frequent sampling operations could result in a rapid battery depletion;
- Node location accuracy in cellular networks, for example, is a central and desirable goal for service providers [16,17]; probing is performed during location tracking, but the probe period must be suitably selected;
- Storage of the sampled data during the observation of mobility traces is necessary. Gaining storage space by selecting the *minimum* sampling frequency without degrading overall accuracy is preferable; *W*e want to emphasize the importance of introducing a predictive approach in estimating the sampling frequency to optimize the use of available resources (energy, space, time).

The remainder of the paper is structured as follows: Section 2 introduces some recent works on mobility sampling and mobility prediction; Section 3 describes the main proposed algorithm; Section 4 provides a detailed description of the main results attained using SPS-WAVE; Section 5 concludes the paper and comments on the main advantages of our proposal.

2. Related work

To the best of our knowledge, no works in literature apart from our previous contribution [7,12,13], which introduced some preliminary results and the wavelet analysis, have studied the mobility process from a spectral point of view or proposed a predictive approach in estimating a suitable sampling frequency. This section provides an overview of contributions that have applied a spectral analysis of time-dependent processes. We again point out that most of the existing works have analyzed time-dependent mobility features to obtain suitable data on the progression of these processes to produce a set of training samples that can be used by the considered predictor (generally, it is a machine-learning based approach, a Kalman filter, etc.).

In [18], the authors proposed a new approach on the topic of pattern reconstruction, discovering that the average error in reconstruction of the sampled patterns scaled linearly and directly proportionally to a constant sampling interval (e.g., the error increased from 1 to 4 meters when the sampling period was extended by 60 s). We note that a sampling frequency of 1/60 = 1.67 mHz may lead to a significant error in accuracy if we consider that a vehicle moving at 50 km/h travels a distance of about 800 meters in that time, which is too great an inaccuracy to be ignored.

The contribution given in [19] discusses the issue of analyzing non-stationary processes to find a more suitable time–frequency relation. The authors suggested employing a complex Auto-Regressive (AR) model derived from the Short-Term AR, STAR model [20,21] and validating it through training with synthetic and real data.

The authors of [22] considered time series processes, introducing a general approach for analysis in the frequency domain. Their proposed method can be generally applied and reveal the main hidden spectral information underlying the considered process. The authors used wavelets to extend the capabilities of classic Fourier theory and demonstrated some interesting results of the spectral relations between time samples.

In the papers [23,24], the authors attempted to study the effects of changing the sampling frequency while determining the most visited locations during a trip. However, they did not consider the time needed to re-arrange the visiting sequence while capturing the transitions between the most frequently visited places.

Table 1
The main parameters defined for the SPS-WAVE algorithm.

Symbol	Description
S	Set of the overall samples of the considered signal (mobility trace)
f^*	Sampling frequency
S_n^k	Subset of S, with length n, extracted at the discrete time k
W_n	Size of the observation time window over S
A_n	Approximation Coefficient (AC) for the wavelet transform of S_n^k
$a_n^k(l)$	Low-frequency function, as the mean approximation of the input signal at step k
D_j^i	jth Determination Coefficient (DC) for the wavelet transform of S_n^k
$d_j^k(l)$	jth high-frequency component of the input signal at step k
F_j	jth spectral range for the wavelet transform of S_n^k equal to [f^*/2^j, f^*/2^{j-1}]
$W_\psi(\cdot)$	The wavelet transform operator
ψ	The mother wavelet function used to approximate the time domain signal
m, r, l	Parameters of the wavelet transform: scaling factor, shifting factor, and time step
L_p	Linear predictor of order p
$L_1^p, \ldots, L_p^p$	Linear predictor coefficients
α	Smoothing factor for the predictive wavelet algorithm
$C[d_1^k(l)]$	High-frequency contribution of the input signal at step k (evaluated as in Eq. (4))
$C^p[d_1^{(k+1)}(l)]$	Predicted high-frequency contribution of the input signal at step k
$C^{pred}[d_1^{(k+1)}(l)]$	Smoothed predicted high-frequency contribution of the input signal at step k

In [25], the authors proposed using the Hilbert-Huang Transform (HHT) for spectral decomposition and analysis of the time series according to temperature on different scales (days, months, years). The authors investigated and discovered the trend of correlation between the samples and the function of the considered scales.

The cited studies are the only ones dedicated to spectral analysis of several processes, and none of them focus on node mobility in dynamic networks [7,26,27]. Many studies in the literature explore mobility prediction. Therefore it is essential to find a method of evaluating the period of time between two collected samples. In real applications of mobility prediction, the selected period will affect the interval between the current and the next predicted position. After an examination of the studies mentioned above and consideration of the need for further investigation of the sampling frequency, we arrived at the following proposed contributions:

- An analysis of real mobility traces extracted from real maps and real node movements obtained in an online environment;
- Based on deep wavelet analysis (already proposed in [7]), integration of SPS-WAVE with the main structure of ADPCM encoders from which we inherit the predictive component. This will aid in determining the sampling frequency in future observation windows according to both the current and predicted spectral constant;
- A comparison of static detection of the sampling frequency and dynamic and predictive SPS-WAVE to highlight the significance of the proposed innovation and accuracy in reconstructing the original mobility samples;
- Presentation of SPS-WAVE as a dynamic algorithm that can select and predict a suitable sampling frequency according to the current and future spectral components (always guaranteeing the minimum sampling frequency derived from Shannon's theorem).

The following sections provide a comprehensive background to the research topic and present the main results attained.

3. Smoothed predictive spectral WAVElet (SPS-WAVE) approach

We first give a brief overview of the main wavelet theory and its features, followed by a thorough description and analysis of the ADPCM technique integrated with the SPS-WAVE algorithm. Table 1 summarizes the main parameters and variables used in our proposal.

3.1. A brief overview of spectral mobility analysis using the wavelet approach

Let us consider a generic discrete set of samples $S = \{s_1, s_2, \ldots, s_m\}$, extracted from a continuous signal with a sampling frequency f^*. In our case, the continuous signal is a coordinate (x, y or z) process of a mobile node in a given network. As shown in [7], wavelets [9] can provide a proper representation in time of the spectral components in S, therefore, if we consider a sub-set of samples $S_n^k = \{s_1, \ldots, s_n\}$ in an observation window W_n at a certain discrete time kt, it is possible to determine the spectral components of the n samples by applying the n-level wavelet transform to S_n.

Wavelet theory is well known [9–11] and it remains beyond the scope of this study to describe in detail, but we briefly review it to recall the main concepts which are important to our proposal. Fig. 1 shows the conceptual scheme of a wavelet transform: starting from an n-samples signal, it can obtain $n+1$ components in the frequency domain (light-blue part of the image).

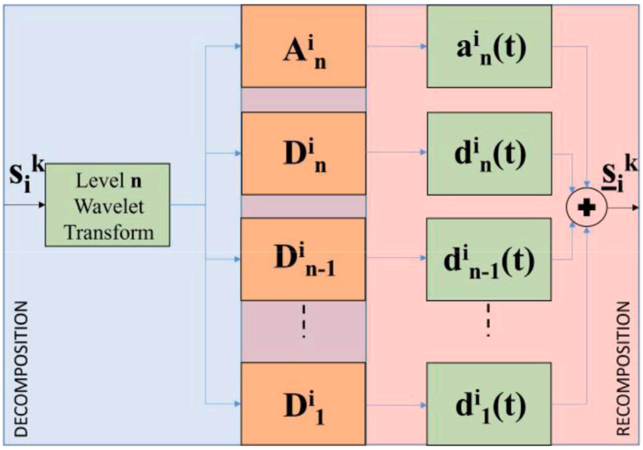

Fig. 1. Schematic representation of the main concept behind the application of a wavelet transform to S_i^k and its inverse transform to express the analyzed signal as a sum of the spectral coefficients.

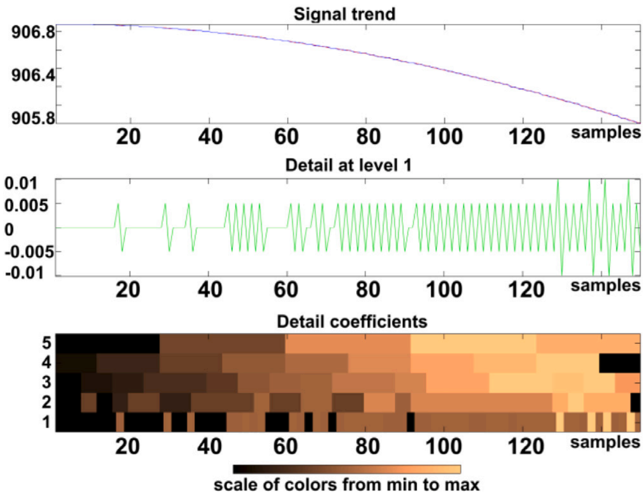

Fig. 2. Example of $n = 5$ wavelet decomposition for a set of $m = 150$ samples process.

So, a decomposition algorithm extracts from the subset S_i^k the $n+1$ coefficients: n Detail Coefficients (DCs) $\{D_1^i, D_2^i, \dots, D_n^i\}$ and one Approximation Coefficient (AC) A_n^i; the last one represents the low-frequency component (or the mean) of the input signal. The main feature of this approach is that each component relates to a specific frequency range in function of the considered sampling frequency. Therefore, at each time kt, it is possible to know which parts of the signal are contributions according to the frequency components. In particular, D_j^i is related to the spectral content of the signal in the range $F_j = \left[\frac{f^*}{2^j}, \frac{f^*}{2^{j-1}}\right]$ and A_n^i, D_n^i are related to the spectral range $F_n = \left[0, \frac{f^*}{2^{n-1}}\right]$, where f^* is the sampling frequency. In this way, the knowledge of AC and DCs allows knowing which spectral components are present in the considered signal.

We would like to describe only the general expression of the wavelet transform at the generic discrete time k while recalling that all the related details and theory can be found in [9–11]:

$$W_\psi[S_n](m,r) = 2^{-\frac{m}{2}} \cdot \sum_l S_n(l)\psi\left(\frac{l - r \cdot 2^m}{2^m}\right), \tag{1}$$

where several indices can be noticed, such as m (the scaling factor), r (the shifting factor), ψ (the mother wavelet function); where m, r, and l are integer numbers (l is the time step). Mother wavelet functions are used instead of sinusoids (as in the Fourier transform), and they define the shape of the approximating function, scaled in both amplitude (by m) and time (by r). This is exactly the result of the light-blue block of Fig. 1 on input S_i^k. The second part (light-red background) of Fig. 1 represents the inverse wavelet transform, by which it is possible to reconstruct a *copy* of the original time-domain signal $\underline{S}_i^k$, starting from AC and DCs and obtaining the result by the sum of a low-frequency function $a_n(k)$ and a set of higher frequency functions $d_j(l)$.

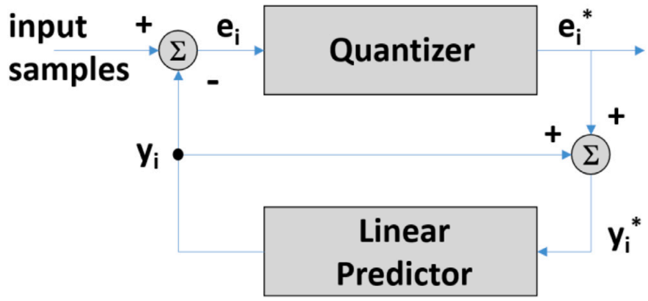

Fig. 3. Basic block diagram of an ADPCM encoder.

Fig. 2 illustrates an example of a Daubechies wavelet applied to S, where $m = 150$ and $n = 5$: it can be observed that the original signal S can be approximated as the sum of the approximation function and all the detail functions, each one representing the content of a particular frequency range. A deeper analysis of the preliminary wavelet transform can be found in [7], while a complete description of the theory about wavelet transforms and their inverse can be found in [9–11].

3.2. Adaptive differential pulse code modulation approach

This section briefly reviews the well-known ADPCM approach [8], as our proposal (next section) is based on its main features.

ADPCM is an encoding algorithm in which scaling during the quantization stages is adapted according to the signal content. It is mainly used in ITU-T audio standards, for example, G.726 [28]. Our attention to ADPCM focuses on its capability of using signal history to predict its content in the near future. As soon as the signal has been processed, the availability of a prediction function allows the future content of a signal to be forecast. We are not interested, instead, in differential coding (i.e., the encoding of only the difference between the actual and predicted values). The basic principle of ADPCM is illustrated in Fig. 3.

As illustrated in the figure, our interest is in the output of the linear predictor y_i at step i, which is subtracted from the input sample to obtain the differential value (error residual) e_i, which will be quantized as e_i^*. The sum of e_i^* and y_i, denoted y_i^*, will be the new input for the predictor in step $i+1$. From the block structure of 3, it is clear that the goodness of the linear predictor has a fundamental role in quantizing accuracy.

ADPCM is, therefore, able to change the step size according to the signal variation. In addition, ADPCM needs only a low-power single wire for data transmission instead of several wires, as in other communications systems. It does not require any synchronization procedure when bits are transmitted. In our proposal, we consider different linear predictors to implement a proper selection of a sampling frequency in the function of a signal's spectral content. We highlight that we do not consider predictive approaches with high computational complexity (such as Artificial Intelligence or other complex mechanisms) because we are interested in low-complexity real-time predictions. The following section formalizes the adaptation of the ADPCM linear prediction for our purposes.

3.3. Smoothed predictive spectral WAVElet algorithm based on the ADPCM linear prediction approach

This section aims to present the derivation of a new dynamic sampling algorithm based on the principles of the ADPCM approach. Let us, therefore, expand the notations used in Section 3.1. Let k be the discrete time step of our analysis, and without loss of generality, let us assume that we start with $k = 0$. Let n be the number of frequency sub-ranges of the interval $[0, f^k]$, where f^k is the sampling frequency of the kth step. Let $a_n^k(l)$ be the approximation function at step k, while $d_1^k(l), \dots, d_n^k(l)$ are the detail functions at step k. This means that at each step k, a set of n samples is collected from S, and the wavelet transform is applied to the sub-signal S_n^k. At this point, we are interested in evaluating the content of the $d_1^k(l)$ function, which indicates whether the spectral range F_1 (the one containing the highest potential spectral components of S_n^k) can be ignored when the $d_1^k(l)$ spectral contribution falls below a given threshold at step k. If so, the sampling frequency is halved to $f^{*(k+1)} = f^{*k}/2$, because the highest frequencies components are not affecting the signal content (i.e. they are negligible), so the sampling frequency can be divided by 2 (for Shannon's theorem and the wavelet theory). Otherwise, if the contribution of $d_1^k(l)$ is above a given threshold, the sampling frequency is doubled to $f^{*(k+1)} = 2 \cdot f^{*k}$.

The main problem with this approach is that the selection of the sampling frequency for the $(k+1)$th step is done during the function of the signal content at step k. Therefore, there may be better choices if the spectral content varies quickly. Therefore, a predictive filter is suitable for selecting an appropriate sampling frequency during analysis. So, let p be the order of the considered linear predictor and $C[d_1^k(l)]$ the high-frequency contribution of the signal; we can then write the Smoothed Predictive Spectral (SPS) contribution as:

$$C^{pred}[d_1^{k+1}(l)] = \alpha \cdot C[d_1^k(l)] + (1-\alpha) \cdot C^p[d_1^{k+1}(l)], \tag{2}$$

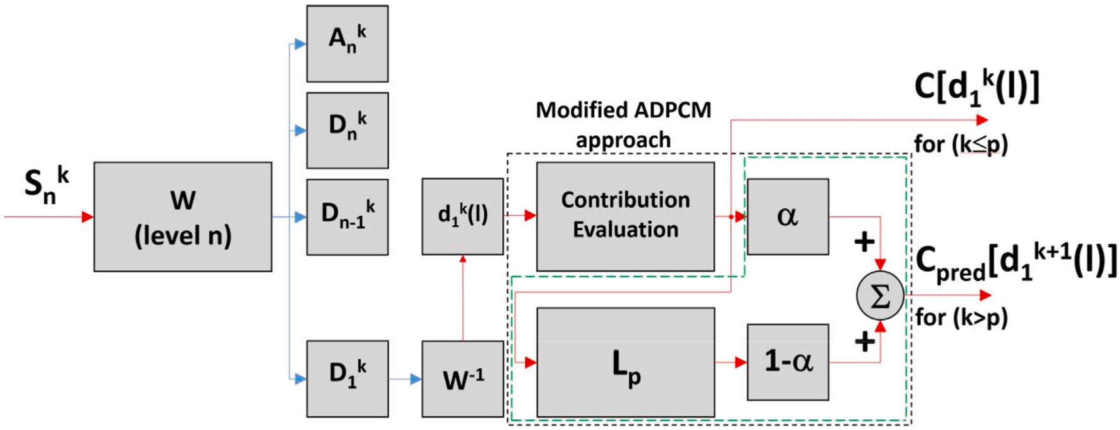

Fig. 4. SPS approach for the wavelet transform with modified ADPCM prediction (red lines for time-domain signals, blue lines for frequency-domain signals).

where α is the smoothing factor and $C^p[d_1^{k+1}(l)]$ is the high-frequency contribution predicted for the next step by the $order-p$ linear predictor L_p applied to the previous p values of $d_1(l)$. The constraints for Eq. (2) are $0 < \alpha < 1$ and $k \gg p$ (the second condition indicates that it is possible to predict the high-frequency contributions only if at least p past contribution values are available to train the predictor). The second term in the sum of Eq. (2) can be expressed as:

$$C^p[d_1^{k+1}(l)] = L_1^p \cdot C[d_1^k(l)] + L_2^p \cdot C[d_1^{k-1}(l)] + \cdots \\ + L_p^p \cdot C[d_1^{k-p}(l)] \tag{3}$$

where the coefficients $\{L_1^p, \dots, L_p^p\}$ can be obtained by several error optimization algorithms, for example, Least Mean Squares (LMS), Recursive Least Squares (RLS), or Mean Square Error (MSE), as discussed in the next section. We refer to Fig. 3 to redefine the predictive algorithm concerning the SPS approach, thus obtaining the structure in Fig. 4.

Fig. 4 illustrates how the input signal S_n^k, which contains n samples at step k, is first transformed by the wavelet W into A_n^k, $D_n^k, \dots, D_1^k$ terms in the frequency domain. The inverse wavelet transform W^{-1} is then applied to the high-frequency component D_1^k, and the time function $d_1^k(l)$ is obtained. This function is analyzed to evaluate its contribution (the rationale is to evaluate how many samples in d_1^k are near 0 or, in general, below a given threshold). Therefore, given a positive near-to-zero threshold Tr_1 and $d = |d_1^k(l)|$ (it always assumes the same value independently of step k because it depends only on n), the contribution function $C[d_1^k(l)]$ can be defined as follows:

$$C[d_1^k(l)] = \frac{\sum_{i=1}^{d} count[d_1^k(l), Tr_1, i]}{d}, \tag{4}$$

where the function $count$ accepts three arguments as input (a set of samples, and two constant values); it is defined as follows:

$$count[d_1^k(l), Tr_1, i] = \begin{cases} 1 & if \;\; |d_1^k(l)(i)| \geq Tr_1 \\ 0 & else \end{cases} \tag{5}$$

In this manner, $0 \leq C[d_1^k(l)] \leq 1$. This is the first term of the smoothed function defined in (2), while the second term $C^p[d_1^{k+1}(l)]$ is the output of L_p and weighted by $1-\alpha$. The final output illustrated in Fig. 4 is exactly the definition introduced in Eq. (2). At this point, we should define how the SPS-WAVE decides to increase (double) or decrease (half) the sampling frequency. We note that the items within the green area in Fig. 4 are active only after the time needed to *train* the linear predictor. Therefore, at least p steps are necessary to evaluate the p coefficients of L_p adequately. Once Eq. (2) becomes applicable, the SPS-WAVE knows the smoothed high-frequency contribution for step $k+1$. Therefore, it must decide whether to change the sampling frequency. Two additional thresholds Tr_{low} and Tr_{high} are used to define whether the sampling frequency should be changed, based on the $C^{pred}[d_1^{k+1}(l)]$ value:

$$f^{k+1} = \begin{cases} 2 \cdot f^k & if \;\; C^{pred}[d_1^{k+1}(l)] \geq Tr_{high} \\ f^k/2 & if \;\; C^{pred}[d_1^{k+1}(l)] \leq Tr_{low} \\ f^k & if \;\; Tr_{low} < C^{pred}[d_1^{k+1}(l)] < Tr_{high} \end{cases} \tag{6}$$

The following Algorithm 1 summarizes the SPS-WAVE approach; obviously, it needs at least p samples before the first prediction can be made to train L_p.

We assume it is necessary to observe at least p steps before predictions can start; the training period may be longer, and we will show the main results related to this issue. From our point of view, it is necessary to discuss the following key aspects of the SPS-WAVE:

Algorithm 1: Smoothed Predictive Spectral WAVElet

Result: Adapt the sampling frequency dynamically to the spectral content of the current and predicted sample contributions
input: S; n; α; f_{start}; Tr_1; Tr_{low}; Tr_{high}; p;
init: set k=0; set $f^k = f_{start}$;
while *n available samples in S* **do**
  $k++$;
  Wait for the storage of n samples ($\frac{n}{f^k}$ sec.);
  $[D_1^k ... D_n^k A_n^k] = \mathrm{W}(S_n^k)$;
  $d_1^k(l) = W^{-1}(D_1^k)$;
  set $d = |d_1^k(l)|$;
  $count = 0$;
  for $q = 1;\ q \leq d;\ q++$ **do**
    if $|d_1^k(l)(q)| > Tr_1$ **then**
      $count++$;
    end
  end
  $curr\_contrib = \frac{count}{d}$;
  if $k \leq p$ **then**
    $train(L_p, curr\_contrib)$; $contrib = curr\_contrib$;
  end
  else
    $pred\_contrib = L_p(curr\_contrib)$;
    $contrib = \alpha \cdot curr\_contrib + (1-\alpha) \cdot pred\_contrib$;
  end
  if $contrib \leq Tr_{low}$ **then**
    $f^{k+1} = f^k/2$;
  end
  if $contrib \geq Tr_{high}$ **then**
    $f^{k+1} = 2 \cdot f^k$;
  end
end

- **Inputs**: S is the original vector of data (in our context, it contains the samples related to one of the mobility coordinates and is generated in real-time while the mobile node is moving); n is the size of the subset of samples considered for each observation window (it is the same value of the wavelet decomposition level); α is the smoothing parameter, able to provide a balance between the current and predicted values; f_{start} is the initial sampling frequency (in most existing systems, it is a static value fixed to 1 Hz); $Tr_1, Tr_{low}, Tr_{high}$ are three parameters which need to be tuned (Tr_1 is a near-to-zero value and $Tr_{low} < Tr_{high}$); p is the order of the linear predictor;
- **Initialization**: once the input parameters have been acquired, the algorithm sets the time-step k to zero and the current sampling frequency to f_{start};
- The main **While**: this loop is related to the time-step k, which is increased by 1 for each cycle. The algorithm waits for the collection of n samples (n/f^k seconds) to have the option of applying the wavelet transform W to S_n^k. Then, the time vector $d_1^k(l)$ is returned by the inverse wavelet transform W^{-1} applied to the detail coefficient D_1^k. At this point, the high-frequency contribution of $d_1^k(l)$ is evaluated according to Eq. (4). If the linear predictor has not yet been trained (at least p samples), then the function $train(L_p, curr_{contrib})$ is invoked; otherwise, a smoothed predictive evaluation of the contribution is performed according to Eq. (2) and the approach in Fig. 4. Finally and eventually, the frequency is changed according to Eq. (6).

Regarding the complexity of the SPS-WAVE, the **init** step can be considered constant, and the main **while** cycle is repeated until n samples are available for the current step (let us say q times). After the collection of the current n samples for the kth step, the W and W^{-1} operations are executed, with a complexity of $O(n)$ [11]. Then the contribution is evaluated with $o(d)$, which is less than $O(n)$. After these preliminary operations, the training and prediction operations may be considered $O(n^2)$ [29], while the update operations related to (6) can be ignored, given their constant complexity. Therefore, the complexity of the SPS-WAVE in the worst case can be described as $q \cdot [O(n) + O(n) + O(n^2)] \approx O(n^2)$.

4. Numerical results

To demonstrate the performance of SPS-WAVE, we worked with several mobility traces. Using SUMO [30], we generated them by importing a road map in .osm format from Open Street Map (OSM) [31] and converting it with the "netconvert.py" command and producing mobile nodes with the "randomtrips.py" script. This script allows the user to select from different profiles (we used

Fig. 5. Map of New York used to generate mobility traces.

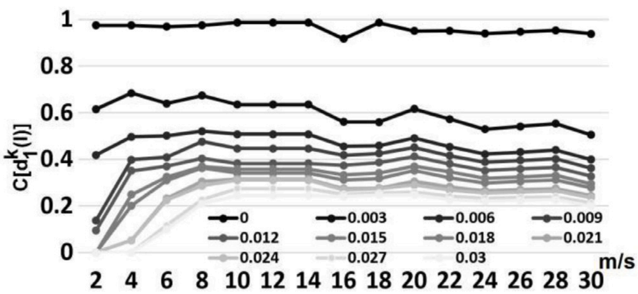

Fig. 6. Trend of $C[d_1^k(l)]$ for different values of Tr_1 as a function of the maximum user speed from 2 m/s to 30 m/s.

the "coach" profile) and set the maximum attainable speed, "speed-factor" (deviation from the speed limit), and "step length", which is an essential component of our proposed method since it defines the mobility sampling period. In our simulations, we used the map of a residential area of New York (NY) centered on the coordinates (lat. 40.7145887, lon. −74.2757245), with an average area of 4 km^2 (Fig. 5). The main input parameters for the "randomtrips.py" script were: "-maxSpeed" in the range [2..30] m/s, "-p" = 4 (period in seconds before each new car enters the map area), "-e" = 100 (simulated time, in seconds), "-device.fcd.period" (variable, representing the sampling period, in seconds). A residential area was selected to set the option of users also moving at high speed compared to downtown mobility. The following sub-sections describe and evaluate the required SPS-WAVE parameters to better illustrate the proposed algorithm.

4.1. The first threshold analysis

We want to underline that the SPS-WAVE needs to be tuned and trained because it is based on some thresholds and a linear predictor. To this aim, mobile traces need to be observed by some *monitoring simulations*, during which mobile information is just stored for further processing, as explained in the following sub-sections. In this way, it is possible to set the needed parameters correctly for the best performance of the SPS-WAVE. First, we verify the effect of the threshold Tr_1 on our observations of the spectral contributions. To this aim, we simulated user mobility according to the previously introduced parameters, a sampling period of $T_s = 10$ ms, and a variable maximum permitted speed in the range of 2 m/s to 30 m/s.

Fig. 6 charts the preliminary results of the simulation for the term defined in Eq. (4), showing the maximum speed and the Tr_1 threshold from 0 to $3e^{-2}$, with a step of $3e^{-3}$. The curves become shallower at higher values of Tr_1. For $Tr_1 = 0$, we can see that the high-frequency contribution is always equal to 0.99, indicating that most of the $d_1^k(l)$ values are different from zero (less than 1% is equal to zero), independently of the moving speed. The contribution decreases at higher values of Tr_1 because fewer terms are over the threshold. We can also see that at very low speeds (between 2 m/s and 4 m/s), the high-frequency contribution reaches zero (no values in $d_1^k(l)$ are more significant than Tr_1). This effect is less visible at greater maximum speeds, where the high-frequency component increases and attains a more stable trend (although it slightly decreases) because the roads become saturated (already at a maximum speed of 8–10 m/s) and mobile nodes cannot increase their speeds indefinitely.

4.2. Evaluating the order of the linear filter L_p

Before the effectiveness of the proposed SPS-WAVE algorithm can be demonstrated, the correct value of p, which is the order of the linear predictor L_p must be found. As input, it receives the p contribution values obtained from Eq. (4) and returns the

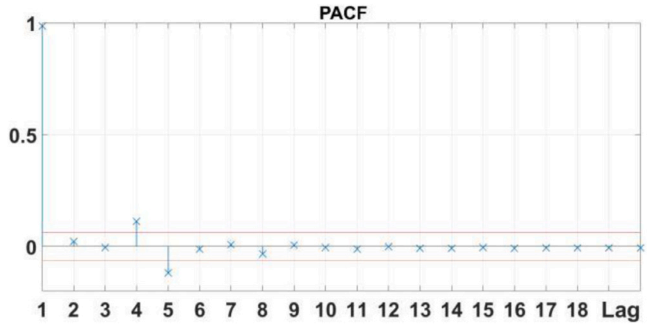

Fig. 7. Typical trend in the PACF correlogram for different lag values (1,20).

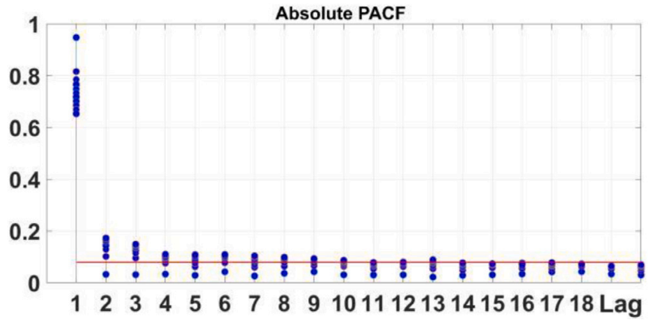

Fig. 8. Average absolute PACF values obtained for different Tr_1 and lag values up to 20.

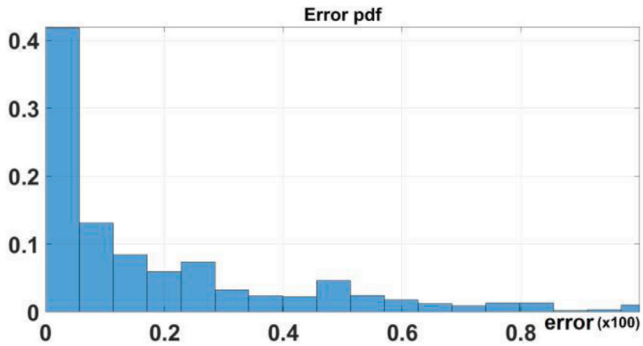

Fig. 9. Contribution prediction error probability density function, where Tr_1 = 2.5e−2.

next predicted contribution value for the $(k+1)$th step (as in Eq. (3)). To investigate the correlation between contribution samples based on the generated mobility traces, we evaluated the Partial Auto-Correlation Function (PACF) [12] for different lag values and obtained several correlograms, as the one depicted in Fig. 7.

Recalling that the PACF indicates the relationship between a sample and the previous jth sample (where j is the lag), Fig. 7, for example, indicates that the PACF at a lag of 1 assumes its maximum value, whereas, at other lag values, the PACF falls between the red lines (representing the 95% confidence interval), except for lags 4 and 5. We can, therefore, conclude that the order of the process is 5 (the PACF in Fig. 7 concerns a wavelet decomposition contribution applied to an x-coordinate process, where the Tr_1 value is equal to 0.003 and a maximum speed of 13.9 m/s). We analyzed all traces, with values of Tr_1 from 0 to 3e−2, and obtained the PACF points in Fig. 8 (absolute values were considered to depict only positive points).

The red line in Fig. 8 represents the upper bound of the 95% confidence interval. Beyond a lag of 10, the PACF remains below the limit, and we can preliminarily conclude that the contribution process is a 10th-order process, i.e., p can be set to 10. We provided to verify the goodness of the assumption regarding p = 10, by evaluating the contribution prediction error (for now, simply the absolute value of the difference between the predicted and actual values), obtaining the pdf illustrated in Fig. 9: it can be seen that the probability of having low prediction errors is very high, with a huge decreasing trend for the probability if higher errors are considered.

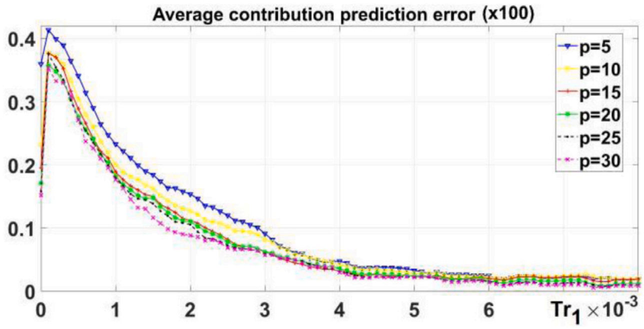

Fig. 10. Trend in average contribution prediction error in the Tr_1 and p functions; graphs are plotted for a Tr_1 step of 1e−4.

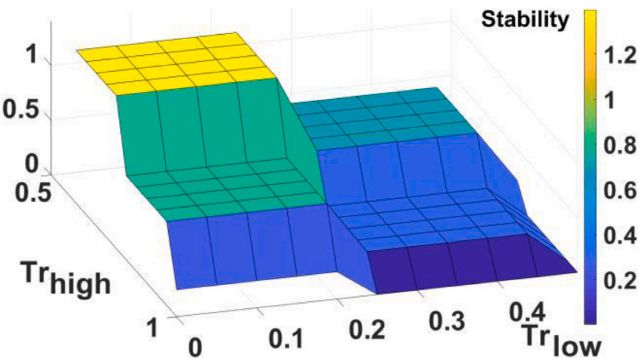

Fig. 11. Trend of SPS-WAVE stability in the Tr_{low} and Tr_{high} functions.

Fig. 10 plots the results for the average prediction error of L_p at different Tr_1 and p values.

Except for the initial points (for a $Tr1$ of 0 to 1e−4), Fig. 10 reveals that the filtering error decreased with higher values of Tr_1 and that beyond Tr_1 = 3.5e−3, the difference at various values of p was negligible. We can especially see that for Tr_1 = 4e−3, a value of p equal to or greater than 8 guaranteed a contribution prediction error of less than 0.05. We will therefore evaluate the SPS-WAVE performance for Tr_1 = 4e−3 and $p = 8$ (this is very close to the result we derived from the PACF trend in Fig. 8).

4.3. Second and third threshold analysis

The final analysis, which should be performed before testing our proposal, concerns tuning Tr_{low} and Tr_{high}. To this aim, we considered several performance parameters, such as the number of upscaling and downscaling (in terms of sampling frequency) that occur according to the selected thresholds, the stability of the sampling frequency adaptation, and the number of stored samples. We considered the ranges [1e−6, 0.5] for Tr_{low} and $[0.5, \dots, 1]$ for Tr_{high} to avoid any overlap.

Fig. 11 shows the trend of the average number of sampling steps before upscaling/downscaling of the sampling frequency, which we consider an index of SPS-WAVE stability (higher values are better, as they indicate a greater number of sampling operations before a change in sampling frequency). Therefore, regarding the stability parameter, setting the values near the lower bounds of both thresholds is the best choice: $Tr_{low} \in (0, 0.2]$ and $Tr_{high} \in [0.5, 0.7]$.

Figs. 12 and 13 plot the average number of upscaling (increased sampling frequency) and downscaling (decreased sampling frequency) in the threshold value functions. It is evident that downscalings, on average, occur more frequently than upscalings: this is an advantage of the SPS-WAVE because reducing the sampling frequency is a desired feature in terms of storage. The effect of Tr_{low} on performance is low (null for upscaling and low for downscaling), whereas the selected Tr_{high} values have a significant effect. In particular, referring to the optimal ranges for stability, the values of Tr_{high} can be left in the interval $[0.5, 0.7]$. At the same time, Tr_{low} can be set to 0.25, thereby ensuring a negligible number of upscaling and an average number of downscaling while only reducing stability within an acceptable range (around 8 steps instead of 11).

Fig. 14 illustrates the trend of stored samples in the threshold value functions. We can observe that the number of processed samples remains between 150 and 180 (the maximum attained value is around 270) for the values of Tr_{low} = 0.25 and $Tr_{high} \in [0.5, 0.7]$.

In this manner, sufficient results were collected to tune the SPS-WAVE algorithm: $p = 8$, Tr_1 = 4e−3, Tr_{low} = 0.25, and Tr_{high} = 0.6. In the following section, the performance of the SPS-WAVE is analyzed concerning the smoothing parameter function α and sampling accuracy.

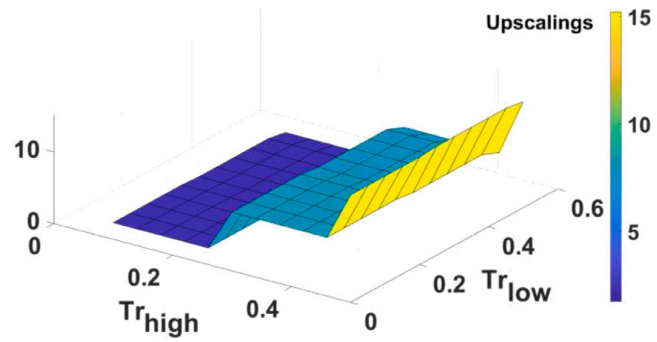

Fig. 12. SPS-WAVE average number of upscalings in the Tr_{low} and Tr_{high} functions.

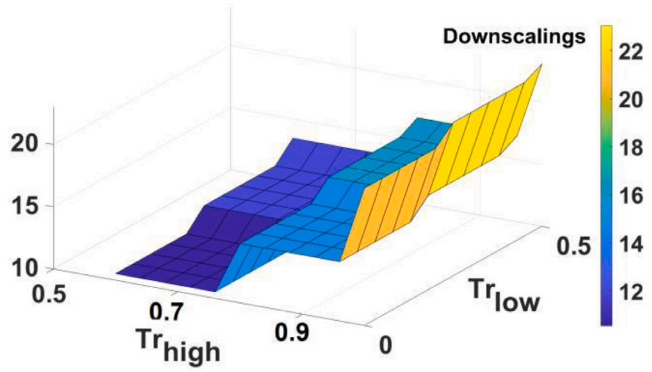

Fig. 13. SPS-WAVE average number of downscalings in the Tr_{low} and Tr_{high} functions.

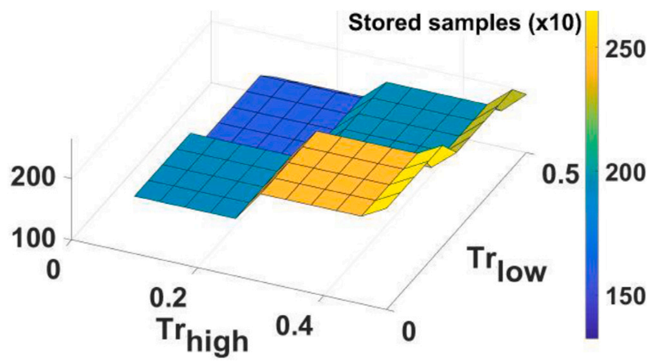

Fig. 14. Average number of stored SPS-WAVE mobility samples in the Tr_{low} and Tr_{high} functions.

4.4. SPS-WAVE performance evaluation

At this point, the main parameters of the SPS-WAVE have been determined, and we, therefore now proceed to evaluate the overall performance of our proposed approach in comparison to other standard approaches (constant sampling and the concept applied in [7], in which no prediction was considered). The only parameter that needs to be considered is α, which establishes the correct weights of the current frequency contribution value and the future value.

Fig. 15 shows the average number of downsamplings according to the (x, y) coordinate pair. The results are presented in a single chart of averages of the x and y values. The red line represents the results of traditional sampling (fixed sampling frequency, i.e., f_k = 1 for the entire period, indicated by $constant$). In contrast, the blue line reflects the results of the study presented in [7], where no prediction feature was considered (indicated by $no - pred$). The $constant$ approach does not vary the sampling frequency. Herefore, no downsamplings were obtained. In this case, instead of $no - pred$, we observe a constant trend of downsamplings because it

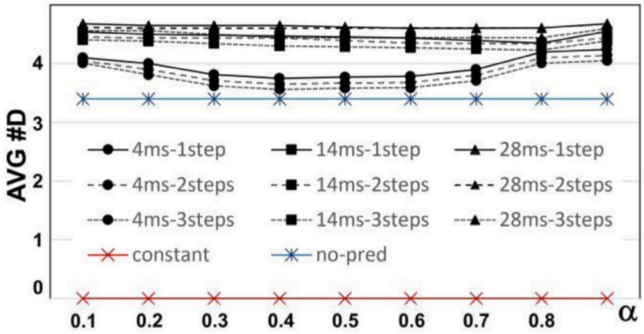

Fig. 15. Average number of SPS-WAVE down-sampling operations ($f^{k+1} = f^k/2$), compared with a constant approach and the not predictive one [7].

does not rely on the α parameter. Regarding SPS-WAVE, we investigated different degrees of mobility (maximum speed): 4 m/s (equal to approx. 15 km/h, low-mobility), 14 m/s (equal to approx. 50 km/h, mid-mobility), and 28 m/s (equal to approx. 100 km/h, high-mobility). In the last two cases, the vehicles were forced to respect the speed limits in downtown but were free to travel at greater speeds outside the critical area. In the low-mobility case, we see an initial slight decrease, while after $\alpha = 0.5$ (the same weight for both the current and predicted coordinates), we see an increasing trend. The difference is negligible in the mid and high-mobility cases because vehicles were forced to respect road speed limits. However, the number of downsamplings was larger because more time was spent at a higher constant speed; the sampling frequency could, therefore, be decreased without any information loss. Regarding the "steps" parameter, we wanted to extend the SPS-WAVE predictive feature by considering one (as already defined), two, or three future samples (steps-ahead); modifying Eqs. (2) and (3), we obtain:

$$\begin{aligned} C^p[d_1^{k+2}(l)] &= L_1^p \cdot C^p[d_1^{k+1}(l)] + L_2^p \cdot C[d_1^k(l)] + \cdots \\ &\quad + L_p^p \cdot C[d_1^2(l)] \ldots \\ C^p[d_1^{k+3}(l)] &= L_1^p \cdot C^p[d_1^{k+2}(l)] + L_2^p \cdot C^p[d_1^{k+1}(l)] + \cdots \\ &\quad + L_p^p \cdot C[d_1^3(l)] \end{aligned} \tag{7}$$

Then, Eq. (2) becomes:

$$C_{pred}[d_1^{k+1}(l)] = \alpha \cdot C[d_1^k(l)] + (1-\alpha) \cdot C^p[d_1^{step}], \tag{8}$$

where

$$\begin{aligned} C^p[d_1^1] = C^p[d_1^{k+1}(l)] &= L_1^p \cdot C[d_1^{k+1}(l)] + L_2^p \cdot C[d_1^k(l)] + \cdots \\ &\quad + L_p^p \cdot C[d_1^2(l)], \end{aligned} \tag{9}$$

$$C^p[d_1^2] = \{C^p[d_1^1] + C^p[d_1^{k+2}(l)]\}/2, \tag{10}$$

$$C^p[d_1^3] = \{C^p[d_1^1] + C^p[d_1^{k+2}(l)] + C^p[d_1^{k+3}(l)]\}/3. \tag{11}$$

In this manner, we can consider 1, 2, or 3 steps ahead to decide whether to reduce or increase the sampling frequency. Fig. 15 shows a negligible difference when considering more steps ahead. Similarly, Fig. 16 shows the average number of upsampling operations.

Also, in this case, constant sampling yields no upsampling, while the constant value in the "no-prediction" algorithm is greater than all the SPS-WAVE results, which all concentrate between 2.89 and 3.6. The results in Figs. 15 and 16 let us conclude that SPS-WAVE allows the option to increase the number of downsampling operations concerning both the "no-pred" or classic constant sampling methods. This is significant since it reduces the number of stored samples while maintaining an acceptable approximation error level (shown later). The number of upsampling operations is also less than the "no-pred" approach (while constant sampling yields a zero value).

Fig. 17 is an example of how mobility is downsampled: we assume a "sample-and-hold" behavior (the reconstructed mobility between two sampled points is constant and equal to the previous point): the blue and black curves represent the trend of the x-coordinate. In contrast, the red curves represent the related mobility sample.

Fig. 18 shows a very interesting result for SPS-WAVE concerning comparing the original and reconstructed samples and evaluating the average MSE. As in the previous cases, constant sampling has a null error, whereas the error in the no-pred case is around 1.22 meters (it does not depend on α). In SPS-WAVE, we can observe that the average reconstruction error in the low-mobility case is almost negligible (from 12 to 19 centimeters). In contrast, in the higher-mobility case, the error increases (approx. 26 centimeters for mid-mobility and nearly 82 centimeters for high-mobility). The steps ahead which were applied did not strongly affect the obtained results.

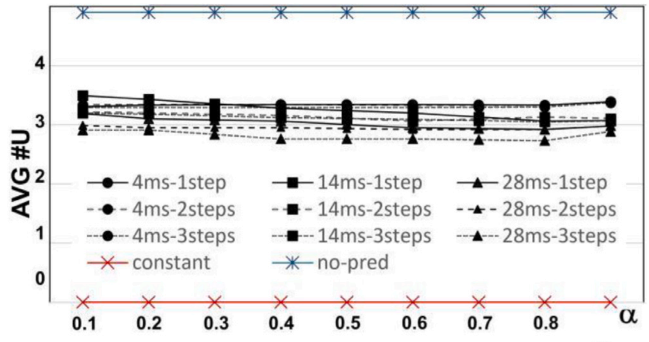

Fig. 16. Average number of SPS-WAVE up-sampling operations ($f^{k+1} = f^k \cdot 2$), compared with a constant approach and the not predictive one [7].

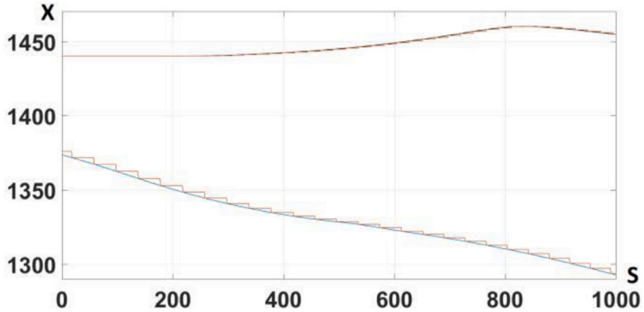

Fig. 17. Example of 1000 sampled mobility (x-coordinate) points, with different downsampling levels.

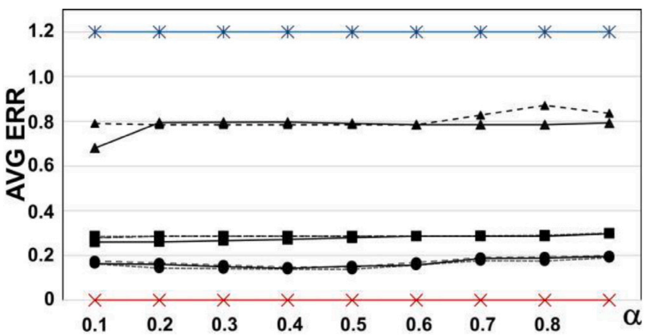

Fig. 18. Trend of the average reconstruction error for different α values and steps ahead, compared with a constant approach and the not predictive one [7].

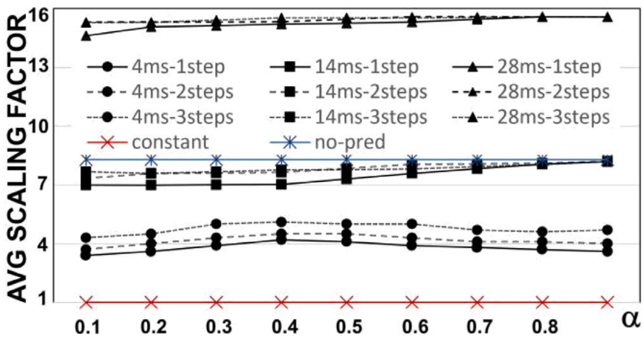

Fig. 19. Trend of the average frequency scaling factor, compared with a constant approach and the not predictive one [7].

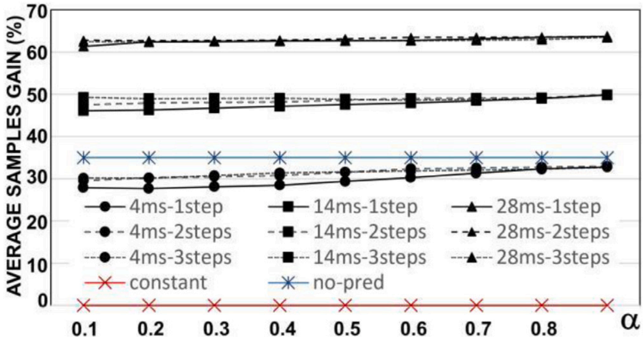

Fig. 20. Trend of the average gain in terms of percentage of sampling operations and stored samples, compared with a constant approach and the not predictive one [7].

Fig. 19 charts the average trend of the scaling factor related to the upsampling and downsampling operations; it can be recursively defined for the sampling step $k+1$ as follows:

$$sf_{(k+1)} = 2^{log_2(sf_{(k)})+c}, \tag{12}$$

where $c = 1$ if a downsampling occurs, $c = -1$ if an upsampling occurs and $c = 0$ otherwise. This definition is based on the starting condition of $sf_{(0)} = 1$ (SPS-WAVE ensures that a frequency greater than the f_{start} cannot be attained; therefore, the minimum sf is always equal to 1). The logical meaning is that if, for example, the average $sf = 5$, then the average sampling frequency was five times lower than f_{start}. Fig. 19 shows that in the high-mobility case, sf increases from 14.7 to 15.9 as α increases. This is one of the best results we obtained because the vehicles' constant speed decreases the mobility process's spectral content without requiring a large amount of constant sampling. As the mobility level decreases, sf also decreases, maintaining an almost constant trend in the function of α; the steps ahead, also in this case, do not significantly affect the overall average curve trend. In each case, we have a minimum gain in sampling operations of approximately 3.2 times (low mobility) and a maximum of 15.9 times (high mobility). For the constant-sampling scheme, sf always equals 1, whereas the no-predictive approach obtains sf, which remains within the average of SPS-WAVE performance.

In Fig. 20, the effects of the scaling factor sf are shown in terms of percentage of gained sampling operations: SPS-WAVE obtained a gain of about 62% (210 k samples on average) for high-mobility, 48.5% (82 k samples on average) for mid-mobility, and 30.2% (54 k samples on average) for low-mobility. Steps ahead did not affect the results, although it demonstrated a slightly increasing trend for higher α values. Constant sampling did not show advantages in gained samples; the no-predictive approach performed better than low-mobility SPS-WAVE but was outperformed in the other cases.

5. Conclusions and future work

The study extends the knowledge introduced in our previous works with a proposed predictive spectral analysis of user mobility. We did not incorporate a specific architecture; therefore, our proposal is suitable for application with any technology (5G, Wireless Sensor Networks, Vehicular Networks, etc.). We demonstrated that introducing a predictive feature in dynamic mobility sampling improves the overall performance of the sampling algorithm: maintaining a constant sampling frequency during the overall process is illogical since the spectral content is dynamic as users move. We demonstrated that achieving considerable improvements in sampling operations and stored data is possible at the cost of applying a dedicated algorithm. We also showed that considering many steps ahead in prediction will noticeably improve the overall system. Therefore, considering only the following sample when the sampling frequency is adjusted is much more appropriate. With these considerations, we plan to extend our proposed method to encompass 2D/3D energy-aware scenarios and factor energy consumption as a critical system parameter; for example, unmanned aerial vehicles could extend flight times if they performed fewer energy-intensive sampling operations.

CRediT authorship contribution statement

Peppino Fazio: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft. **Miralem Mehic:** Data curation, Formal analysis, Investigation, Software, Supervision. **Floriano De Rango:** Formal analysis, Investigation, Resources, Writing – review & editing. **Mauro Tropea:** Formal analysis, Investigation, Methodology, Resources, Software. **Miroslav Voznak:** Conceptualization, Formal analysis, Funding acquisition, Project administration, Supervision, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The research leading to the published results was supported by the European Union within the REFRESH project — Research Excellence For Region Sustainability and High-tech Industries ID No. CZ.10.03.01/00/22 003/0000048 of the European Just Transition Fund.

References

[1] G. Liu, et al., Vision, requirements and network architecture of 6G mobile network beyond 2030, China Commun. Mag. 17 (9) (2020).
[2] B. Zheng, et al., SOUP: Spatial-temporal demand forecasting and competitive supply, IEEE Trans. Knowl. Data Eng. (2021) (preprint).
[3] T. Martin, et al., Effective machine-learning models for predicting routability during FPGA placement, in: ACM/IEEE 3rd Workshop on Machine Learning for CAD, MLCAD, 2021.
[4] A. Singh, et al., Using deep reinforcement learning for routing in IP networks, in: International Conference on Computer Communications and Networks, ICCCN, 2021.
[5] R. Xu, et al., DarkTE: Towards dark traffic engineering in data center networks with ensemble learning, in: IEEE 19th International Symposium on Quality of Service, IWQOS, 2021.
[6] P. Fazio, et al., Prediction and QoS enhancement in new generation cellular networks with mobile hosts: A survey on different protocols and conventional/unconventional approaches, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1822–1841.
[7] P. Fazio, M. Mehic, M. Voznak, On the relationship between speed and mobility sampling frequency in dynamic urban networks, IEEE Syst. J. 17 (1) (2023) 696–707, http://dx.doi.org/10.1109/JSYST.2022.3186640.
[8] M. Honig, et al., Comparison of adaptive linear prediction algorithms in ADPCM, IEEE Trans. Commun. 30 (7) (1982).
[9] M. Vetterli, J. Kovacevic, V.K. Goyal, Fourier and Wavelet Signal Processing, Cambridge University Press, 2014.
[10] A. Graps, An introduction to wavelets, IEEE Comput. Sci. Eng. 2 (2) (1995) 50–61.
[11] W.H. Press, et al., Numerical Recipes in C: The Art of Scientific Computing, second ed., Cambridge University Press, 1992, pp. 591–606.
[12] P. Fazio, et al., A deep stochastical and predictive analysis of users mobility based on auto-regressive processes and pairing functions, J. Netw. Comput. Appl. 168 (2020).
[13] P. Fazio, et al., An innovative dynamic mobility sampling scheme based on multi-resolution wavelet analysis in IoT networks, IEEE Internet Things J. (2022) http://dx.doi.org/10.1109/JIOT.2021.3126550.
[14] Y. Qi, C. Yu, Y.J. Suh, S.Y. Jang, GPS tethering for energy conservation, in: IEEE Wireless Communications and Networking Conference, WCNC, 2015, pp. 1320–1325.
[15] S. Vanini, F. Faraci, A. Ferrari, S. Giordano, Using barometric pressure data to recognize vertical displacement activities on smartphones, Comput. Commun. 87 (2016) 37–48.
[16] A. Finamore, M. Mellia, Z. Gilani, K. Papagiannaki, V. Erramilli, Y. Grunenberger, Is there a case for mobile phone content pre-staging? in: ACM Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT, 2013, pp. 321–326.
[17] M. Ficek, et al., Performance study of active tracking in a cellular network using a modular signaling platform, in: Proc. of the 8th Int. Conference on Mobile Systems, Applications, and Services, MobiSys, 2010, pp. 239–254.
[18] P. Katsikouli, et al., On the sampling frequency of human mobility, in: IEEE GLOBECOM, 1–6, Singapore, 2017.
[19] M.C.B. Andrade, M.J. Porsani, B. Ursin, Complex autoregressive time-frequency analysis: Estimation of time-varying periodic signal components, IEEE Signal Process. Mag. 35 (2) (2018) 142–153.
[20] I. Zoukaneri, M.J. Porsani, A combined Wigner-Ville and maximum entropy method for high-resolution time-frequency analysis of seismic data, Geophysics 80 (6) (2015) 1–11.
[21] T.J. Ulrych, T.N. Bishop, Maximum entropy spectral analysis and autoregressive decomposition, Rev. Geophys. 13 (1) (1975) 183–200.
[22] P. Masset, Analysis of financial time series using wavelet methods, in: Handbook of Financial Econometrics and Statistics, Springer, 2014, pp. 539–573.
[23] B. Thierry, B. Chaix, Y. Kestens, Detecting activity locations from raw gps data: A novel kernel-based algorithm, Int. J. Health Geogr. 12 (1) (2013) 14–26.
[24] B.C. Csaji, A. Browet, V. Traag, J.C. Delvenne, E. Huens, P.V. Dooren, Z. Smoreda, V.D. Blondel, Exploring the mobility of mobile phone users, Physica A 392 (6) (2013) 1459–1473.
[25] D. Kbaier Ben Ismail, P. Lazure, I. Puillat, Advanced spectral analysis and cross correlation based on the empirical mode decomposition: Application to the environmental time series, IEEE Geosci. Remote Sens. Lett. 12 (9) (2015).
[26] H. Zhang, et al., Mobility prediction: A survey on state-of-the-art schemes and future applications, IEEE Access 7 (2018) 802–822.
[27] M. Zareei, et al., Mobility-aware medium access control protocols for wireless sensor networks: A survey, J. Netw. Comput. Appl. 104 (2018) 21–37.
[28] J. Vehvilainen, et al., A processor core for 32 kbit/s G.726 ADPCM codecs, in: EEE International Symposium on Circuits and Systems, ISCAS, vol. 3, 1995, pp. 1932–1935.
[29] S. Ciolino, et al., On the use of wavelet packets in ultra wideband pulse shape modulation systems, IEICE Trans. (2005) 2310–2317.
[30] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, Evamarie Wießner, Microscopic traffic simulation using SUMO, in: IEEE Intelligent Transportation Systems Conference, ITSC, 2018.
[31] Open Street Map (OSM) , https://www.openstreetmap.org/.