

Learning-Based Speed Estimation from Accelerometer-Only Inertial Sensing

Barak Or, *Member, IEEE*

Abstract—The proposed model, CarSpeedNet, estimates scalar vehicle speed from a window of three-axis smartphone acceleration, without gyroscope, wheel-odometry, vehicle-bus, or positioning input at inference. The reported experiment comprises 13.2 hours of on-road driving. Beyond the network comparison, a finite-context analysis treats window length as part of the sensing problem. For nested histories, the minimum Bayes mean-square error is non-increasing with context; a complementary cue-coverage relation links the same window to the amount of speed-dependent vibration presented to the network and to the observation horizon. At a 1-s input, CarSpeedNet is compared with five temporal-network alternatives; the effect of temporal context is then measured over six window lengths. On the 0.5-hour holdout, a 4-s window yielded a root-mean-square error (RMSE) of 1.8 m/s and a mean absolute error (MAE) of 0.72 m/s, compared with 2.9 and 1.3 m/s at 1 s. The 178,169-parameter network requires about 0.68 MiB for 32-bit weights.

Index Terms—Accelerometer, intelligent sensors, inertial sensing, smartphone, vehicle speed estimation.

I. INTRODUCTION

Wheel encoders, vehicle networks, and satellite navigation are the usual sources of vehicle speed. They are not always accessible to an auxiliary or low-cost sensing system, and direct integration of smartphone acceleration is quickly corrupted by bias, mounting orientation, and vibration [1], [2]. Learning-based inertial methods offer a different route: they infer the state from temporal structure rather than integrating each sample [3]–[6].

Prior smartphone studies have combined acceleration with stop-based recalibration [7], used accelerometer-gyroscope pairs with recurrent networks [8], [9], or addressed phone placement and orientation through alignment, augmentation, and attention [10]–[12]. Our earlier speed estimator [13] used a six-axis inertial measurement unit (IMU), about 3 hours of data, RTK-GNSS supervision, and a dead-reckoning evaluation. The present experiment is distinct: inference uses only the three accelerometer axes, the reported corpus is 13.2 hours, five alternative temporal models are analyzed at 1 s, and CarSpeedNet is tested at six input lengths. Vehicle-vibration studies also indicate that inertial signatures can carry route- and motion-dependent information [14].

The main contribution is a finite-context characterization of the acceleration window supplied to CarSpeedNet in (2) and Fig. 1. For nested histories, the minimum Bayes mean-square error is non-increasing with context. We then connect the physical scale of a vibration cue to the finite input seen by the network by deriving how many cycles of that cue fall

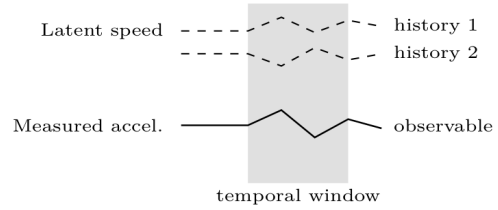


Fig. 1. Partial observability in accelerometer-only sensing. Distinct latent speed histories can be consistent with similar local acceleration. The shaded region marks the temporal context supplied to the estimator.

inside a W -sample window. This yields a minimum-context condition linking vehicle speed, spatial scale, sampling rate, and the observation horizon carried by each prediction. Across the six tested contexts, holdout RMSE decreases from 4.9 to 1.8 m/s and MAE from 2.50 to 0.72 m/s; the separate 1-s study compares CarSpeedNet with five temporal alternatives.

The remainder of the letter is organized as follows. Section II gives the sensing model and finite-context analysis, Section III describes the data and CarSpeedNet, Section IV reports the holdout results, and Section V concludes.

II. ACCELEROMETER-ONLY SENSING

A smartphone accelerometer measures specific force, not velocity. A simplified measurement model is

$$\mathbf{a}_m(t) = \mathbf{R}^T(t)(\dot{\mathbf{v}}(t) - \mathbf{g}) + \mathbf{b}(t) + \mathbf{n}(t), \quad (1)$$

where t denotes time, $\mathbf{a}_m(t)$ is the measured three-axis specific force, $\mathbf{R}(t)$ is the phone orientation rotation matrix, $\mathbf{v}(t)$ is vehicle velocity in the navigation frame, $\dot{\mathbf{v}}(t)$ is its time derivative, $\mathbf{g}$ is the gravity vector, $\mathbf{b}(t)$ is additive accelerometer bias, and $\mathbf{n}(t)$ is measurement noise. Short acceleration sequences are not sufficient to distinguish all velocity histories, particularly during constant-speed or low-dynamic motion. Fig. 1 illustrates this partial observability. A longer window cannot make the problem fully observable, but it can expose stops, accelerations, and changes in vibration that are absent from an instantaneous sample.

CarSpeedNet implements the finite-horizon mapping

$$\hat{s}_t = f_{\theta}(\mathbf{a}_{t-W+1:t}), \quad (2)$$

where $\hat{s}_t$ is the estimated scalar speed at time t , f_{θ} is the learned regression function with parameters θ , $\mathbf{a}_{t-W+1:t}$ denotes the sequence of three-axis acceleration samples in the input window, and W is the number of samples. The estimator learns correlations present in the training corpus; it does not separately identify orientation, bias, road excitation, or vehicle dynamics.

TABLE I
CLOSEST VEHICLE-SPEED STUDIES AND EXPERIMENTAL SCOPE

Study	Inference sensors	Estimation method	Reported scope
Ustun and Cetin [7]	Smartphone accelerometer	Integration with stop-based recalibration	Average error reported within 10 mph on its collected sample data
Freydin and Or [13]	Six-axis IMU	Recurrent DNN speed pseudomeasurement	Approximately 3 h; RTK-GNSS supervision; aided dead-reckoning evaluation
Shin et al. [12]	Smartphone accelerometer and gyroscope	Attention-LSTM regression	Four phones; 41.8 km; several orientations; OBD-II reference
Xiao et al. [11]	Smartphone IMU	Pose alignment and sequence learning	Crowdsourced data; phone-placement augmentation; GNSS supervision
This letter	Smartphone accelerometer only	Recurrent-convolutional regression	13.2 h; five alternatives at 1 s; six CarSpeedNet input lengths

A. Finite-Context Accuracy-Responsiveness

Fig. 1 makes the role of finite context explicit: the learned mapping in (2) receives only a W -sample segment of acceleration, while similar local measurements may remain compatible with different latent speed histories. Thus, W controls the information presented to the network, not only the dimensions of its input. Let $\mathcal{A}_W = \mathbf{a}_{t-W+1:t}$ denote that history. Under squared-error loss, the Bayes-optimal estimate based on this window is $\hat{s}_W^* = \mathbb{E}[s_t | \mathcal{A}_W]$. Its minimum mean-square error is

$$\mathcal{R}^*(W) = \mathbb{E}[(s_t - \hat{s}_W^*)^2] = \mathbb{E}[\text{Var}(s_t | \mathcal{A}_W)]. \quad (3)$$

The final term is the conditional variance averaged over the observed acceleration windows. If $W_2 > W_1$, then $\mathcal{A}_{W_1}$ is contained in $\mathcal{A}_{W_2}$. Any estimator defined on the shorter history remains admissible on the longer history by ignoring the extra samples. Hence

$$\mathcal{R}^*(W_2) \leq \mathcal{R}^*(W_1), \quad W_2 > W_1. \quad (4)$$

The monotonicity follows from the information set itself rather than from a particular architecture. Here, however, $\mathcal{A}_W$ is exactly the sequence presented to CarSpeedNet, so (4) separates the information benefit of a longer input history from the choice of recurrent or convolutional layers.

Equation (4) characterizes the information limit, but it does not show what additional temporal evidence becomes available to the network as W grows. Every recurrent and convolutional layer in CarSpeedNet ultimately operates on the same finite W -sample record. To make the effect of that record explicit, let x denote traveled distance and let $r(x)$ be one spatial vibration component with amplitude A and wavelength λ :

$$r(x) = A \cos\left(\frac{2\pi x}{\lambda}\right). \quad (5)$$

Over a short interval with approximately constant speed s , $x(t) = x_0 + st$. The component therefore reaches the accelerometer with temporal period

$$\tau(s, \lambda) = \frac{\lambda}{s}, \quad (6)$$

equivalently at frequency s/λ . At sampling rate f_s , a W -sample input spans $T_W = W/f_s$ and contains

$$C_W(s, \lambda) = \frac{T_W}{\tau(s, \lambda)} = \frac{sW}{\lambda f_s} \quad (7)$$

cycles of that component. Thus, increasing W changes a quantity directly available to the temporal network: the number of repetitions of a speed-dependent cue contained in one input. If at least k cycles are to be present, the required context satisfies

$$W \geq W_k(s, \lambda) = \left\lceil \frac{k\lambda f_s}{s} \right\rceil. \quad (8)$$

This relation couples the physical scale of the excitation to the network input length. It also exposes the responsiveness cost. The first full-context estimate requires $T_W = W/f_s$ seconds of measurements, and every later prediction still depends on that same span of history. Longer context therefore increases cue coverage and cannot worsen the Bayes limit, but it lengthens the observation horizon represented by each prediction. Section IV-B examines this finite-context trade-off using the six available CarSpeedNet windows.

III. DATA AND MODEL

A. Acquisition and Preprocessing

The study reports 13.2 hours of highway and urban-road driving collected in Israel with a Samsung Galaxy smartphone. The stated allocation is 10.3 hours for training, 2.6 hours for validation, and 0.5 hours for holdout evaluation. The 0.5-hour subset is used as the holdout set.

Data collection began after stable GPS positioning had been confirmed from the geometric dilution of precision indicator [15]. Acceleration was acquired at 500 Hz, low-pass filtered, and downsampled to 20 Hz. GPS speed was sampled at 1 Hz and supplied the training reference, but was not used at inference.

B. CarSpeedNet

Fig. 3 reproduces the original architecture. A 100-unit bidirectional LSTM is followed by four LSTM layers with 50, 20, 20, and 20 units and batch normalization. Three one-dimensional convolutional layers use 64, 64, and 32 filters with kernel size 3. A 32-unit fully connected layer and scalar output complete the model, with rectified-linear activations between applicable layers [16]. Training used Adam [17], an initial learning rate of 0.001, batch size 32, and at most 200 epochs. The configuration lists a decay factor of 0.2 over 30,000 steps. CarSpeedNet has 178,169 trainable parameters, equivalent to 0.68 MiB of 32-bit weights, excluding activations and framework overhead.

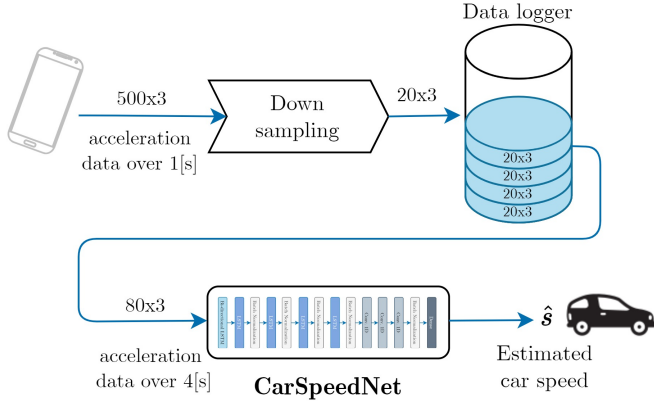


Fig. 2. Recorded sensing pipeline. One-second, 500-Hz acceleration blocks are downsampled to 20 samples; four blocks form the 80-sample CarSpeedNet input.

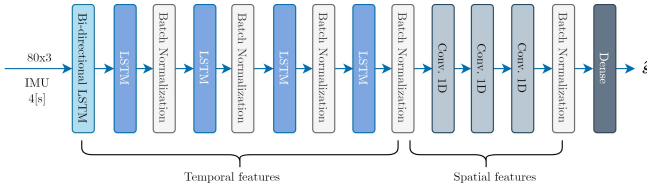


Fig. 3. Original CarSpeedNet architecture. The input contains 80 three-axis accelerometer samples (4 s at 20 Hz).

IV. HOLDOUT RESULTS

Performance is reported using root-mean-square error

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{j=1}^N (s_j^{\text{GPS}} - \hat{s}_j)^2}, \quad (9)$$

and mean absolute error

$$\text{MAE} = \frac{1}{N} \sum_{j=1}^N |s_j^{\text{GPS}} - \hat{s}_j|, \quad (10)$$

where N is the number of evaluated speed samples, j is the sample index, s_j^{GPS} is the GPS-derived reference speed for sample j , and $\hat{s}_j$ is the corresponding CarSpeedNet estimate.

A. Model Comparison

Five alternatives were trained on the same accelerometer-only input: DNN* adapted from [13], a conventional LSTM [18], WaveNet [19], a bidirectional LSTM, and a ResNet-inspired temporal model [20]. Fig. 4 contains the single-run comparison for a 1-s window. CarSpeedNet yielded 2.9 m/s RMSE and 1.3 m/s MAE. The ResNet-inspired model had the same RMSE and 1.4 m/s MAE. Repeated-seed results and a common tuning budget are unavailable, so the small difference does not support a statistical ranking. CarSpeedNet is also not the smallest network: its 178,169 parameters are fewer than WaveNet's 239,937 but more than the other alternatives.

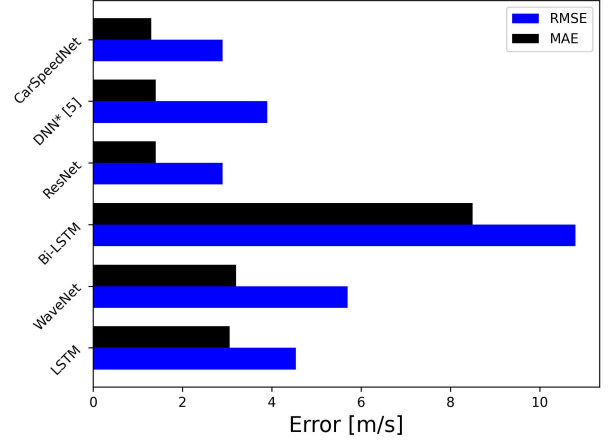


Fig. 4. Single-run holdout RMSE and MAE for six accelerometer-only models using a 1-s window.

TABLE II
HOLDOUT ACCURACY ACROSS INPUT WINDOWS

Samples / duration	RMSE (m/s)	MAE (m/s)
5 / 0.25 s	4.9	2.50
10 / 0.50 s	4.2	2.10
20 / 1.00 s	2.9	1.30
40 / 2.00 s	2.3	1.00
60 / 3.00 s	2.2	0.88
80 / 4.00 s	1.8	0.72

B. Temporal Context

Table II provides the empirical counterpart to the risk relation in (4). RMSE decreases from 4.9 to 1.8 m/s as the window grows from 0.25 to 4 s; MAE decreases from 2.50 to 0.72 m/s. Both errors decrease monotonically across all six tested contexts, corresponding to reductions of 63.3% in RMSE and 71.2% in MAE between the shortest and longest windows. At 20 Hz, the tested inputs span observation horizons from 0.25 to 4 s. For any fixed speed s and spatial scale λ , (7) therefore gives a 16-fold increase in cue coverage between the shortest and longest inputs, while the represented observation horizon also grows by the same factor.

Fig. 5 reproduces the saved traces for $W \in \{10, 40, 80\}$. In the displayed intervals, the estimate approaches zero with the GPS reference and becomes smoother as the window grows. Quantitative comparisons rely on Table II.

C. Experimental Limitations

The evaluation is based on the available recorded corpus and does not include a dedicated cross-device or cross-vehicle generalization study. Robustness to different smartphone placements, sensor characteristics, and driving environments remains to be evaluated in future work.

V. CONCLUSION

CarSpeedNet estimates vehicle speed from smartphone acceleration alone. The finite-context analysis treats window

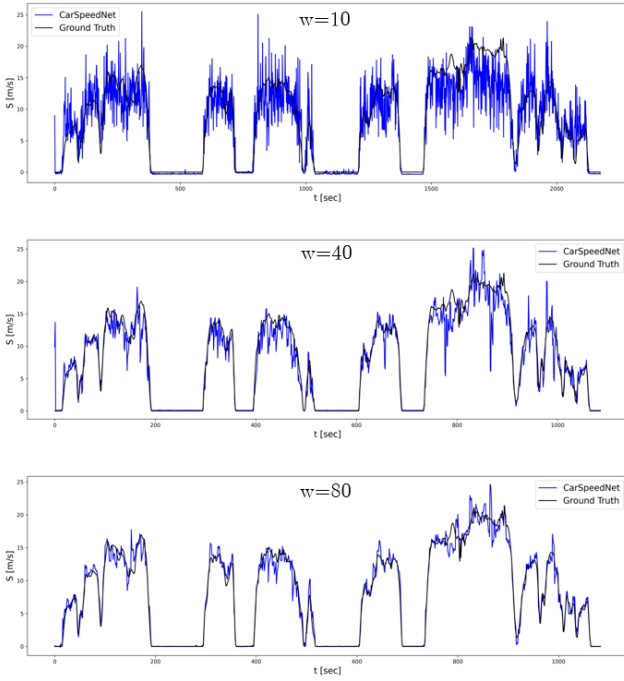


Fig. 5. Representative CarSpeedNet traces and 1-Hz GPS-derived reference for $W \in \{10, 40, 80\}$. The embedded legend retains the original label “Ground Truth.”

length as part of the sensing problem rather than only a network setting. A longer nested history cannot increase the minimum Bayes mean-square error. For speed-dependent vibration, the same window also determines how many repetitions of a physical spatial scale are exposed to the temporal network, while $T_W = W/f_s$ sets the history carried by each prediction. The main empirical result is the reduction from 4.9/2.50 m/s RMSE/MAE at 0.25 s to 1.8/0.72 m/s at 4 s on the recorded holdout. At 1 s, CarSpeedNet and the ResNet-inspired alternative have the same RMSE. The monotonic window-length trend is consistent with this finite-context view: longer windows provide a larger information set and more repeated temporal evidence, at the cost of a longer observation horizon.

CONFLICT OF INTEREST

The author declares no conflict of interest.

REFERENCES

- [1] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. New York, NY, USA: McGraw-Hill, 2008.
- [2] E. Mantouka, E. Barmounakis, E. Vlahogianni, and J. Golias, “Smartphone sensing for understanding driving behavior: Current practice and challenges,” *Int. J. Transp. Sci. Technol.*, vol. 10, no. 3, pp. 266–282, 2021.
- [3] M. Brossard, A. Barrau, and S. Bonnabel, “AI-IMU dead-reckoning,” *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 585–595, 2020.
- [4] Y. Li et al., “Inertial sensing meets machine learning: Opportunity or challenge?” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 9995–10011, 2022.
- [5] B. Or and I. Klein, “Learning vehicle trajectory uncertainty,” *Eng. Appl. Artif. Intell.*, vol. 122, Art. no. 106101, 2023.
- [6] Y. Tong et al., “Vehicle inertial tracking via mobile crowdsensing: Experience and enhancement,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.
- [7] I. Ustun and M. Cetin, “Speed estimation using smartphone accelerometer data,” *Transp. Res. Rec.*, vol. 2673, no. 3, pp. 65–73, 2019.
- [8] Q. Wang, Y. Gu, J. Liu, and S. Kamijo, “DeepSpeedometer: Vehicle speed estimation from accelerometer and gyroscope using LSTM model,” in *Proc. IEEE ITSC*, 2017, pp. 1–6.
- [9] N. E. A. Abdelgawad, A. El Mahdy, W. Gomaa, and A. Shoukry, “Estimating vehicle speed on highway roads from smartphone sensors using deep learning models,” in *Proc. IEEE ICTAI*, 2019, pp. 979–986.
- [10] M. Freydin, N. Sfaradi, N. Segol, A. Eweida, and B. Or, “Deep learning for inertial sensor alignment,” *IEEE Sensors J.*, vol. 24, no. 10, pp. 17282–17290, 2024.
- [11] X. Xiao, X. Ren, and H. Li, “An inertial sequence learning framework for vehicle speed estimation via smartphone IMU,” arXiv:2505.18490, 2025.
- [12] B. Shin, S. Li, and B. Kim, “Deep learning-based vehicle speed estimation using smartphone sensors in GNSS-denied environment,” *Appl. Sci.*, vol. 15, no. 16, Art. no. 8824, 2025.
- [13] M. Freydin and B. Or, “Learning car speed using inertial sensors for dead reckoning navigation,” *IEEE Sensors Lett.*, vol. 6, no. 9, Art. no. 6003104, 2022.
- [14] B. Or, N. Segol, A. Eweida, and M. Freydin, “Learning position from vehicle vibration using an inertial measurement unit,” *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 10766–10776, 2024.
- [15] I. Sharp, K. Yu, and Y. J. Guo, “GDOP analysis for positioning system design,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3371–3382, 2009.
- [16] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *Proc. ICML*, 2010, pp. 807–814.
- [17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv:1412.6980, 2014.
- [18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [19] A. van den Oord et al., “WaveNet: A generative model for raw audio,” arXiv:1609.03499, 2016.
- [20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE CVPR*, 2016, pp. 770–778.