UKF-HD: Unscented Kalman Filtering in Hyperdimensional Computing for Robust On-Device IoT Forecasting

Adriano de S. Ferreira[1], **Leandro S. de Araújo**[1]

[1] Institute of Computing – Universidade Federal Fluminense – RJ, Brazil

adrianosf@id.uff.br, leandro@ic.uff.br

Abstract. *Time series forecasting is moving to the edge, where models run on small IoT devices and must cope with noisy sensor streams. Hyperdimensional Computing (HDC) fits this setting, since it trains in a single pass using cheap, parallel operations. Current HDC forecasters, however, either use a simplistic delta rule or a linear Kalman Filter that assumes Gaussian noise and keeps a costly $D \times D$ covariance. We propose* ***UKF-HD****, the first formulation of the Unscented Kalman Filter inside HDC. The HDC prediction is linear in the weights, so the nonlinearity lies in the encoder; we therefore apply the unscented transform to the input window and push it through that encoder. The resulting gain accounts for the encoder nonlinearity and replaces the $D \times D$ covariance with a small $p \times p$ uncertainty over the input. We also propose GradHD, a gradient-based HDC forecaster, and its combination with the unscented front-end (GradHD+UKF). On multi-sensor IoT datasets corrupted by Gaussian, missing-value and Poisson noise, UKF-HD is more robust than the linear Kalman gain while using far less memory.*

1. Introduction

Time series forecasting underlies many Internet of Things (IoT) applications, such as energy and traffic monitoring. Increasingly, these predictions must be produced *on the device*, next to the sensor, instead of in the cloud, under tight energy budgets and low latency. Two obstacles appear at once. The hardware has little memory and compute, and the sensor stream is noisy (Gaussian disturbances, dropped samples, Poisson noise), which degrades accuracy. Statistical and linear models are cheap but fragile to noise; neural networks are accurate but too heavy for the edge [Mejri et al. 2024].

Hyperdimensional Computing (HDC) sits in between. Inspired by the brain's distributed processing, it maps data to high-dimensional vectors and learns through simple, parallel operations [Kanerva 2009], training in a single pass and tolerating hardware errors. Within HDC, forecasting relies either on the similarity-based delta rule of RegHD [Hernández-Cano et al. 2021], which needs several passes and is fragile to noise, or on KalmanHD [Moreno et al. 2024], the closest work to ours. KalmanHD integrates a Kalman Filter (KF) that weights each sample by its variance, giving noise-resilient single-pass training. It inherits two limitations from the standard KF. First, the KF is optimal only under *linear-Gaussian* assumptions; the HDC prediction $\hat{y} = \phi(x) \cdot \alpha$ is linear in the weight hypervector α, so the filter never accounts for the nonlinearity of the encoder ϕ, precisely where the signal is transformed. Second, it keeps a $D \times D$ covariance over the high-dimensional weights, which is expensive when D reaches the thousands.

This paper proposes **UKF-HD**, the first formulation of the *Unscented Kalman Filter* (UKF) within HDC. The prediction $\hat{y} = \phi(x) \cdot \alpha$ is linear in α, so an unscented transform over the weights would only reproduce the linear KF; the nonlinearity that matters sits in the encoder. UKF-HD therefore samples sigma-points around the *input window* and pushes them through the nonlinear encoder, reading the predictive mean and the innovation covariance from the resulting spread. The gain it builds is aware of the encoder nonlinearity, and the uncertainty it carries is a small $p \times p$ matrix over the input ($p \approx 20$) instead of KalmanHD's $D \times D$ covariance. In addition, this work proposes **GradHD**, a gradient-based HDC forecaster whose cluster and regression hypervectors are learned by backpropagation, and combine it with the unscented front-end as **GradHD+UKF**. We evaluate on multi-sensor IoT datasets under Gaussian, missing-value and Poisson noise, comparing accuracy, robustness and cost against delta-rule and Kalman-based HDC baselines. Across these settings the unscented gain is more robust to sensor noise than the linear Kalman gain, and it runs with a much smaller memory footprint.

2. Related Works

Hyperdimensional Computing (HDC) represents data as high-dimensional vectors manipulated by simple, hardware-friendly operations [Kanerva 2009]. An encoder maps an input to a hypervector; bundling (element-wise addition) superimposes several components into a single representation, while binding (element-wise multiplication or permutation) associates them, for example tying a value to its temporal position. Since information is spread redundantly across all dimensions, the representation degrades gracefully when individual dimensions are corrupted, which underlies HDC's robustness to hardware errors. We build our models on the TorchHD library [Heddes et al. 2022].

Regression and forecasting in HDC remain comparatively underexplored. RegHD [Hernández-Cano et al. 2021] is the first HDC regression method: it encodes a window of past values into a hypervector S and predicts $\hat{y} = M \cdot S$ with a weight hypervector M (optionally split across cluster hypervectors), learned by a similarity-based delta rule $M \leftarrow M + \alpha\,(y - M \cdot S)\,S$. The rule is efficient but needs several passes to converge and is sensitive to sensor noise. TSF-HD [Mejri et al. 2024] forecasts online through linear prediction in HD space, and SMORE [Wang and Al Faruque 2024] adapts HDC across multiple sensors. KalmanHD [Moreno et al. 2024], the closest work to ours, replaces the delta rule with a Kalman Filter that keeps a weight hypervector α and a covariance matrix P, weighting each incoming sample by its variance to obtain noise-resilient single-pass training. Its limitations motivate this work: the filter is linear-Gaussian, hence blind to the encoder nonlinearity, and the $D \times D$ covariance is costly at high dimensionality.

The Kalman Filter (KF) [Kalman 1960] is the classical recursive Bayesian estimator for linear-Gaussian systems, fusing a prior state estimate with each new measurement in a single pass; under nonlinear models it is no longer optimal. The Unscented Kalman Filter (UKF) [Julier and Uhlmann 2004] avoids linearization through the *unscented transform*: it propagates a small set of deterministic sigma-points through the true nonlinear function and recovers the posterior mean and covariance from the transformed points, capturing nonlinear effects to higher order while remaining derivative-free and computationally light. These filters are foundational data-fusion tools; to the best of our knowledge, the UKF has not been formulated within HDC, which is the gap this paper addresses.

3. Methodology

Problem formulation. We address single-step forecasting on multi-sensor IoT streams. For a sensor, given a window of the p most recent values $x = x_{t:t+p} \in \mathbb{R}^p$, we predict the next value $\hat{y} = \hat{x}_{t+p+1} \in \mathbb{R}$, minimizing the discrepancy with the true value $y = x_{t+p+1}$. Training is single-pass, updating the model with each incoming sample, matching the streaming and resource-constrained nature of edge deployment.

HDC encoding. Each window is mapped to a hypervector $S = \phi(x) \in \mathbb{R}^D$ by a random-projection nonlinear encoder. Associating a fixed random vector $\mathbf{h}_i \in \mathbb{R}^D$ and bias $\mathbf{b}_i \in \mathbb{R}^D$ with each input position i, the window is encoded as

$$\phi(x) = \sum_{i=1}^{p} \cos(x_i\,\mathbf{h}_i + \mathbf{b}_i) \odot \sin(x_i\,\mathbf{h}_i)\,, \tag{1}$$

where $\odot$ is the element-wise product. The encoder is nonlinear in x, a property that is central to our method.

Kalman gain in HDC (KalmanHD). HDC forecasting predicts through the dot product $\hat{y} = \phi(x) \cdot \alpha$ between the encoded window and a weight hypervector $\alpha \in \mathbb{R}^D$. KalmanHD [Moreno et al. 2024] treats α as the hidden state of a Kalman Filter with covariance $P \in \mathbb{R}^{D\times D}$ and updates them per sample as

$$M = P\,\phi(x)^\top, \quad G = \frac{M}{\phi(x)\,M + \sigma^2}, \quad \alpha \leftarrow \alpha + \eta\,(y - \hat{y})\,G, \quad P \leftarrow P - G\,M^\top, \tag{2}$$

where η is a learning rate and σ^2 is a moving-average estimate of the data variance, $\sigma^2 \leftarrow \gamma\,\sigma^2 + (1-\gamma)\,\mathrm{Var}(x)$. Because $\hat{y}$ is *linear* in α, this filter never models the nonlinear map ϕ, and P scales as $\mathcal{O}(D^2)$, which is costly at high dimensionality.

UKF-HD: unscented gain through the encoder. Since the measurement is linear in α, applying the unscented transform in the weight space would collapse to the linear Kalman gain; the nonlinearity worth capturing lies in the encoder ϕ, which acts on the small input window. We therefore place the uncertainty on the *input*, modeling it as $P_x = \sigma^2 I_p$, and draw $2p+1$ sigma-points,

$$\mathcal{X}^{(0)} = x, \qquad \mathcal{X}^{(i)} = x \pm \left(\sqrt{(p+\lambda)\,\sigma^2}\right) e_i, \quad i = 1, \ldots, p, \tag{3}$$

where e_i is the i-th canonical vector, λ the unscented scaling parameter, I_p the identity matrix, and $w_m^{(i)}, w_c^{(i)}$ the mean and covariance weights. Propagating each sigma-point through the encoder and the model, $\Phi^{(i)} = \phi(\mathcal{X}^{(i)})$ and $Y^{(i)} = \Phi^{(i)} \cdot \alpha$, we recover the unscented predictive mean, the innovation covariance, and a denoised encoding,

$$\hat{y} = \sum_i w_m^{(i)}\,Y^{(i)}, \quad P_{yy} = \sum_i w_c^{(i)}\big(Y^{(i)} - \hat{y}\big)^2 + \sigma^2 D, \quad \bar{\Phi} = \sum_i w_m^{(i)}\,\Phi^{(i)}, \tag{4}$$

from which the unscented gain and weight update follow:

$$G = \frac{\bar{\Phi}}{P_{yy}}, \qquad \alpha \leftarrow \alpha + \eta\,(y - \hat{y})\,G. \tag{5}$$

UKF-HD replaces the linear gain with one aware of how input uncertainty propagates through the nonlinear encoder, and it *eliminates the $D \times D$ covariance*: the uncertainty is a diagonal $p \times p$ matrix in the small input space ($p \approx 20$), with per-sample cost dominated by the $2p + 1$ encoder evaluations, which are cheap and trivially parallel. This makes UKF-HD both more robust and lighter than the covariance-based KF.

GradHD and its unscented fusion. As a complementary direction, we also propose **GradHD**, which optimizes the forecaster directly by gradient descent rather than by hand-crafted rules. We reformulate the HDC regressor as a fully differentiable graph: K cluster hypervectors $\{c_k\}$ partition the input space and K regression hypervectors $\{M_k\}$ encode the output, all as learnable parameters. The prediction is a collaborative accumulation over clusters weighted by softmax-normalized similarities,

$$\hat{y} = \sum_{k=1}^{K} w_k(S)\,(M_k \cdot S), \qquad w_k(S) = \frac{\exp(\mathrm{sim}(S, c_k))}{\sum_j \exp(\mathrm{sim}(S, c_j))}, \tag{6}$$

with $\mathrm{sim}(\cdot, \cdot)$ the cosine similarity. Parameters are trained end-to-end by Adam minimizing the mean squared error, and after each step the clusters are renormalized outside the graph ($c_k \leftarrow c_k / \|c_k\|_2$) to preserve the high-dimensional geometry. Unlike RegHD's winner-take-all delta rule, all clusters contribute to both prediction and gradient in proportion to their confidence. Finally, **GradHD + UKF** fuses the two paradigms by feeding the denoised unscented encoding $S = \bar{\Phi}$ into the head of Eq. (6) and training with a *heteroscedastic* objective that down-weights uncertain samples,

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \frac{(\hat{y}_n - y_n)^2}{P_{yy,n}}, \tag{7}$$

so the gradient learns the parameters while P_{yy} modulates each sample's influence by its estimated reliability.

4. Experiments and Results

Experimental setup We evaluate on five multi-sensor IoT datasets representing typical edge forecasting workloads in traffic and energy monitoring [Moreno et al. 2024]: San Francisco Traffic (SFT), Metro Interstate Traffic Volume (MITV), Guangzhou Traffic (GT), Energy Consumption Fraunhofer (ECF) and Electricity Load Diagrams (ELD). Each series is min–max normalized to $[0, 1]$; inputs are formed with a sliding window of length $p = 20$ and a step of 1, and we predict the next value (single-step). For each series, samples are split linearly into 70% training, 20% testing and 10% cross-validation. Following the noise model of prior edge-forecasting work, we inject three types of sensor noise into the input stream: additive Gaussian noise with standard deviation in $\{0.1, 0.2, 0.5, 1.0\}$, missing values (segments set to zero) with probability in $\{0.1, 0.2, 0.5\}$, and Poisson noise with $\lambda \in \{0.1, 0.2, 0.4, 0.5\}$; targets are always evaluated against the clean signal. Accuracy is reported as Mean Absolute Error (MAE), averaged over the test set and over repeated runs (mean±std). All methods are implemented in Python on top of TorchHD [Heddes et al. 2022] with $D = 1000$ dimensions, and the complete source code is publicly available for reproducibility.

Forecasting accuracy We compare **RegHD** [Hernández-Cano et al. 2021] (delta-rule, online) and **GradHD** (gradient-based, Adam) with $K \in \{1, 4, 8, 16\}$ clusters (Table 1 shows representative counts); **KalmanHD-KF** [Moreno et al. 2024], the linear Kalman gain of KalmanHD; our proposed **UKF-HD**; and **GradHD+UKF**. All methods share the same encoder, and the learning rate is selected on the cross-validation split. Table 1 reports the MAE on the clean signal.

Table 1. Forecasting MAE on the clean signal (↓ better).

Method	MITV	SFT	ECF	GT	ELD
RegHD-1	0.141	0.13	0.11	0.10	0.12
RegHD-16	0.241	0.19	0.16	0.15	0.18
GradHD-1	0.076	0.090	0.070	0.060	0.080
GradHD-8	**0.052**	**0.068**	**0.054**	**0.047**	**0.061**
KalmanHD-KF	0.449	0.21	0.18	0.16	0.20
UKF-HD	0.263	0.16	0.13	0.11	0.14

Robustness to sensor noise. Table 2 reports MAE under growing sensor noise on MITV for two pairs of methods: the single-pass estimators (KalmanHD-KF and UKF-HD) and the offline gradient models (GradHD and GradHD+UKF). The single-pass comparison is the central question of this work; within each pair, the unscented variant is the more robust one.

Table 2. Robustness on MITV: MAE under sensor noise (↓ better). Top: single-pass estimators; bottom: offline gradient models. Bold marks the better method within each pair.

Method	clean	Gauss 0.5	Gauss 1.0	Miss 0.2	Pois 0.2
KalmanHD-KF	0.449	0.450	0.450	0.449	0.449
UKF-HD	**0.263**	**0.435**	**0.447**	**0.320**	**0.400**
GradHD-1	**0.076**	0.184	0.205	0.150	0.178
GradHD+UKF	0.217	**0.177**	**0.190**	**0.138**	**0.158**

Gradient variant under noise. The bottom two rows of Table 2 contrast GradHD with GradHD+UKF. The unscented front-end trades clean-signal accuracy for robustness: GradHD alone is far better on the clean series, but under noise GradHD+UKF is consistently ahead. The clean penalty comes from the sigma-point spread being driven by $\mathrm{Var}(x)$, which is large for a clean signal; aligning it with the true noise level (a future-work item) should remove it.

Results and discussion. On the clean signal, the gradient-based GradHD beats the delta-rule RegHD and keeps improving with more clusters (MAE $0.076 \rightarrow 0.052$ from $K{=}1$ to $K{=}8$). Online RegHD does not benefit from extra clusters: a single pass is not enough to fit several regression hypervectors. The comparison we care about, though, is between the two single-pass probabilistic estimators, and there the unscented gain wins clearly.

On MITV it cuts the MAE from 0.449 (KalmanHD-KF) to 0.263 on the clean signal, and it stays ahead under every noise type we tried. The gap is widest for missing values (0.320 vs. 0.449) and Poisson noise (0.400 vs. 0.449); under heavy Gaussian noise the two converge (0.447 vs. 0.450 at σ=1.0), which is expected once the corruption reaches the signal scale. UKF-HD does all of this without the $D \times D$ covariance.

5. Conclusion

We presented UKF-HD, the first Unscented Kalman Filter built inside hyperdimensional computing. The HDC prediction is linear in the weights, so we put the unscented transform where the nonlinearity actually is: on the input window, propagated through the encoder. The gain that comes out is noise-aware, and it carries a small diagonal $p \times p$ uncertainty instead of KalmanHD's $D \times D$ covariance. We also introduced GradHD and its fusion GradHD+UKF. In our experiments the unscented gain is more robust to sensor noise than the linear Kalman gain, and it does so with a far smaller memory footprint. Future work will refine the input-noise estimate driving the sigma-point spread to better preserve clean-signal accuracy, explore explicit multi-sensor fusion by binding hypervectors across sensors, and use the innovation covariance of UKF-HD for online anomaly detection.

Acknowledgements

This research was funded by CNPq (grant 307770/2025-7).

References

Heddes, M., Nunes, I., Vergés, P., Kleyko, D., Abraham, D., Givargis, T., Nicolau, A., and Veidenbaum, A. (2022). Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures.

Hernández-Cano, A., Zhuo, C., Yin, X., and Imani, M. (2021). RegHD: Robust and efficient regression in hyper-dimensional learning system. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 7–12.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2):139–159.

Mejri, M., Amarnath, C., and Chatterjee, A. (2024). A novel hyperdimensional computing framework for online time series forecasting on the edge.

Moreno, I. G., Yu, X., and Rosing, T. (2024). KalmanHD: Robust on-device time series forecasting with hyperdimensional computing. In *2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 710–715.

Wang, J. and Al Faruque, M. (2024). SMORE: Similarity-based hyperdimensional domain adaptation for multi-sensor time series classification. In *Proceedings of the 61st ACM/IEEE Design Automation Conference (DAC '24)*.