

Algorithms For Pre-Processing And Noise Filtering In Streaming Telemetry Data Of Iot Devices

Abdugafur Khotamov

Associate Professor at SamDU named after Sharof Rashidov

Kim Andrey Odilovich

Student of SamDU named after Sharof Rashidov

ondriflore@gmail.com

Annotation

This article examines algorithms for preprocessing and filtering noise in streaming telemetry data from IoT devices. We classify noise sources (measurement noise, ADC quantization, and impulse noise during packet loss and retransmission) and formulate requirements for online processing: limited memory, on-the-fly processing without re-passing the data, and predictable computational complexity per sample. We consider moving averages, exponential smoothing, median filters, Hampel filters, recursive Kalman filters, and wavelet noise suppression, as well as hybrid pipelines that combine outlier removal with subsequent smoothing. Using a synthetic telemetry stream simulating temperature sensor readings with Gaussian noise and impulse spikes, we compare the methods in terms of root mean square error (RMSE), processing time per sample, and response latency. The hybrid Hampel filter + exponential smoothing pipeline is shown to reduce RMSE by 84.1% while maintaining $O(1)$ computational complexity per sample, making it suitable for deployment on low-power edge devices.

Keywords: Internet of Things, streaming data processing, noise filtering, Kalman filter, median filter, exponential smoothing, edge computing, telemetry.

1. Introduction

The growth of connected Internet of Things (IoT) devices and the widespread adoption of telemetry monitoring systems—from industrial sensors and smart greenhouses to wearable electronics—have led to a dramatic increase in the volume of real-time streaming data from peripheral devices. Sensor readings (temperature, humidity, vibration, current, position) are almost never free of noise: measuring transducers introduce their own thermal and quantum noise, wireless communication channels (LoRaWAN, Zigbee, Wi-Fi, NB-IoT) are susceptible to interference and packet loss, and operating conditions themselves (vibration, electromagnetic interference, power surges) create pulse spikes that dramatically distort individual readings.

The presence of incorrect or noisy values in the telemetry stream directly reduces the quality of subsequent analysis: false alarms in alarm systems, erroneous decisions by machine learning algorithms trained on "dirty" data, and excessive load on communication channels and storage systems when transmitting redundant information. Unlike traditional offline processing, where the entire data set is immediately available, streaming telemetry requires algorithms that operate "on the fly": without re-tracing the history, with limited RAM and predictable latency for processing a single reading.

The aim of this work is to systematize and comparatively analyze pre-processing and noise filtering algorithms applicable to streaming telemetry data of IoT devices, as well as to experimentally evaluate their accuracy, computational complexity and response delay on a synthetic dataset reproducing typical distortions of a real sensory stream.

2. Noise sources and features of streaming telemetry processing

Classification of noise sources

Telemetry streams from IoT devices typically contain several classes of interference. Additive measurement noise arises from thermal noise in analog circuits and the limited bit depth of the ADC; it is typically well described by a zero-mean Gaussian model. Glitches are single, sharp spikes in values caused by short-term sensor failures, electromagnetic interference, or packet decoding errors. Unlike additive noise, they cannot be effectively suppressed by linear filters and require robust methods. Zero drift and slow trends are associated with component aging and the temperature dependence of sensor characteristics. Finally, packet

loss and reordering during wireless transmission lead to an uneven sampling grid, which must be taken into account when selecting the sliding window size.

Requirements for stream filtering algorithms

Working in stream processing mode imposes a number of limitations on filtering algorithms that are not typical for classical digital signal processing with access to the entire data array:

- Online processing. The algorithm should produce an estimate of the filtered value immediately after the next sample is received, without waiting for future data (except for a limited delay within the window).
- Limited memory. The buffer size must not grow with the length of the stream; it is acceptable to store only the last N samples or a compact state (e.g., an exponential average or a Kalman filter state vector).
- Constant computational complexity per sample. For deployment on low-power microcontrollers (ESP32, STM32), algorithms with complexity $O(1)$ – $O(\log W)$ are preferred, where W is the window size.
- Minimal response delay (lag). Excessive smoothing improves accuracy in static conditions, but introduces a delay during sudden but legitimate signal changes, which is critical for real-time control systems.

3. Pre-processing and filtering algorithms

Sliding anti-aliasing filters

The simplest method for suppressing additive noise is a simple moving average (SMA), which replaces the current sample with the average of the last W observations. This method is easily implemented in a circular buffer, but requires storing the last W values and introduces a latency of about $W/2$ samples. Exponential moving average (EMA) eliminates the need for buffer storage: the new estimate is a weighted combination of the current measurement and the previous estimate, resulting in $O(1)$ memory and time complexity, making EMA particularly attractive for embedded systems.

Robust methods for eliminating outliers

Linear smoothing filters perform poorly on spikes: a single sharp jump is "smeared" across the entire averaging window. A median filter, which replaces the current value with the median of the last W samples, is robust to spikes because the median is insensitive to individual extreme values in the sample. The Hampel filter is more selective: it calculates the median and median absolute deviation (MAD) in a sliding window and replaces a sample with the median only if it deviates from the median by more than a specified number of MAD sigmas; otherwise, the original value is preserved without distorting the useful signal.

Recursive Bayesian Filtering: Kalman Filter

The Kalman filter is a recursive Bayesian estimator for the state of a linear dynamic system observed with additive Gaussian noise. Each step involves a prediction step (extrapolation of the state and its variance from the process model) and a correction step (updating the estimate based on a new measurement weighted by the Kalman gain). The key advantage of the method is its optimality in terms of minimum mean square error for linear-Gaussian systems and natural adaptation of the gain to the ratio of process noise to measurement noise. A disadvantage is its sensitivity to spikes, which the filter interprets as unlikely but still valid measurements.

Wavelet noise reduction

Wavelet noise suppression is effective for signals with temporally localized features (sharp edges, short-term events): the signal is decomposed into approximating and detailing coefficients at several levels, after which thresholding (a soft or hard threshold, such as the universal Donoho threshold) is applied to the detailing coefficients, and the signal is reconstructed using the inverse transform. Unlike the recursive methods discussed above, the classical wavelet transform requires block-based data processing (windows of fixed length, a multiple of a power of two), which complicates its application in a purely streaming mode and is typically used in conjunction with a sliding buffer processed with overlapping processing.

Hybrid preprocessing pipelines

In practice, two-stage pipelines yield the best results: in the first stage, a selective robust filter (Hampel filter) removes spikes without distorting the rest of the signal; in the second stage, recursive smoothing (EMA or Kalman filter) is applied to the cleaned stream to suppress residual additive noise. This architecture combines the robustness of robust methods to anomalies with the effectiveness of linear filters in suppressing random noise, while maintaining $O(1)$ – $O(W)$ complexity per sample.

4. Mathematical foundations

Exponential smoothing is described by the recurrence relation

$$\hat{y}_k = \alpha x_k + (1 - \alpha) \hat{y}_{k-1}, 0 < \alpha \leq 1,$$

where x_k is the current measurement, $\hat{y}_k$ is the smoothed estimate, α is the smoothing coefficient: the closer α is to unity, the lower the filter's inertia and the higher its sensitivity to noise.

The median absolute deviation (MAD) used in the Hampel filter is calculated as

$$MAD = k \cdot \text{median}(|x_i - \text{median}(x)|), k \approx 1.4826,$$

where the coefficient k brings the estimate to the scale of the standard deviation for a normal distribution; a sample x_i is considered an outlier and is replaced by the median if $|x_i - \text{median}(x)| > n \cdot MAD$ (usually $n = 3$).

The scalar Kalman filter at each step k performs a prediction

$$\hat{x}_k^- = \hat{x}_{k-1}, P_k^- = P_{k-1} + Q,$$

and correction for the newly received measurement z_k :

$$K_k = P_k^- / (P_k^- + R), \hat{x}_k = \hat{x}_k^- + K_k \cdot (z_k - \hat{x}_k^-), P_k = (1 - K_k) \cdot P_k^-,$$

where Q is the process noise variance, R is the measurement noise variance, and K_k is the Kalman gain. The Q/R ratio determines the balance between confidence in the model and the measurement: a small Q yields a smoother but more inertial estimate.

5. Experimental evaluation

Experimental methodology

To compare the methods, a synthetic stream of $N = 2000$ samples was generated, simulating temperature sensor readings. The true signal is a superposition of a daily sine component and a faster harmonic, supplemented with Gaussian measurement noise ($\sigma = 0.6$ °C) and pulse spikes with an amplitude of 5–12 °C in 2% of randomly selected samples, which corresponds to the typical frequency of ADC failures and packet losses in real-world LoRaWAN/Zigbee network deployments. Each algorithm processed the stream sequentially, sample by sample, without access to future values. The metrics used were the root mean square error (RMSE) relative to the true signal, the average processing time for one sample, and the response delay (lag), estimated from the position of the maximum cross-correlation between the filtered and true signals.

Results

Figure 1 shows a fragment of the flow: noisy measurements, the true signal, and the results of the hybrid Hampel + EMA pipeline and the Kalman filter. It is clear that the hybrid pipeline effectively suppresses the single spike (around the 240th sample), while the Kalman filter only partially smooths it out, interpreting the spike as an unlikely but valid measurement.

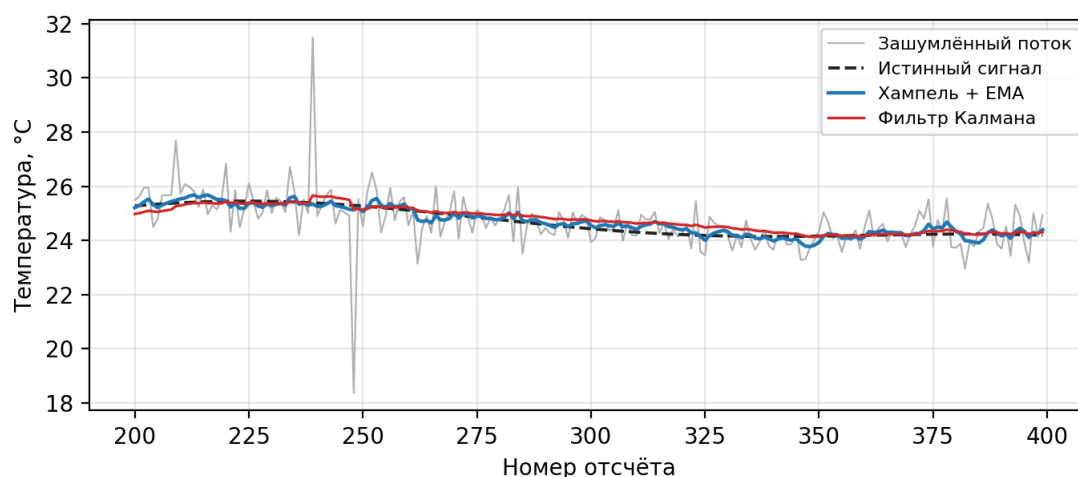


Fig. 1. A fragment of the telemetry stream: raw data, true signal and filtering results

The summary results for all the considered methods are shown in Table 1 and Figure 2. The raw stream has $RMSE = 1.335$ °C. Simple moving average and exponential smoothing reduce the error by 67–71%, but are sensitive to impulse outliers that are “smeared” across the averaging window. The median filter provides a higher RMSE reduction (81.8%) due to its robustness to outliers, but requires window sorting at each step (complexity $O(W \log W)$ with a naive implementation). The Hampel filter alone shows a more modest result (58.7%), since it is initially configured to conservatively correct only obvious anomalies, leaving additive noise virtually unchanged. The best result is demonstrated by the hybrid Hampel + EMA pipeline: an RMSE reduction of 84.1% while maintaining low computational complexity.

Table 1. Comparison of filtering algorithms on a synthetic telemetry stream

Method	RMSE, °C	RMSE reduction, %	Time / count, μ s	Lag, countdowns
No filter	1.335	—	—	—
SMA (window = 9)	0.439	67.1	4.49	+4
EMA ($\alpha = 0.15$)	0.390	70.8	0.32	+6
Median (window = 9)	0.244	81.8	12.03	+3
Hampel filter (window = 7, 3σ)	0.552	58.7	24.87	+1
Kalman filter	0.377	71.8	0.29	+15
Hybrid: Hampel + EMA	0.212	84.1	24.75	+5

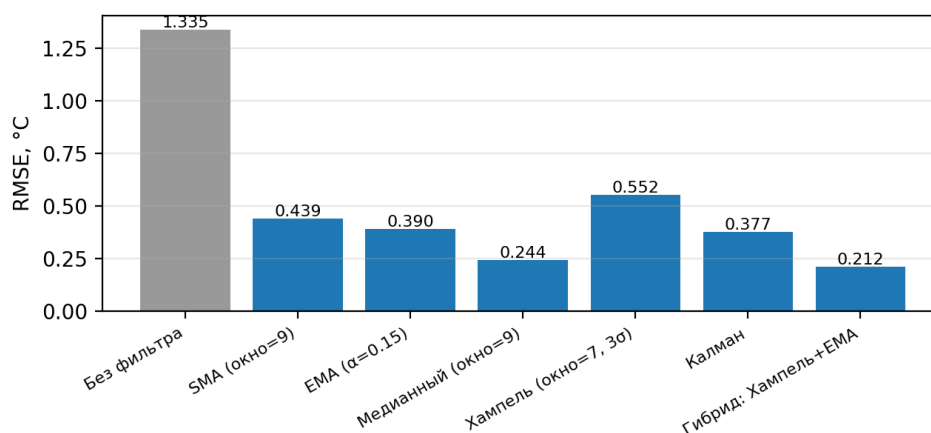


Fig. 2. Root Mean Square Error (RMSE) for different filtering methods

The response delay deserves special attention: the Kalman filter with a low process noise variance Q produces the largest lag (+15 samples), as it places a high degree of confidence in the model and reacts slowly to real trend changes. The Hampel filter, on the other hand, introduces minimal lag (+1 sample), as it does not modify non-outlier values. The hybrid pipeline occupies an intermediate position (+5 samples), providing an acceptable compromise between accuracy and speed.

6. Edge Deployment Architecture

The algorithms discussed differ fundamentally in their resource requirements, which determines their place in the telemetry collection architecture. Algorithms with $O(1)$ sample complexity (EMA, Kalman filter, Hampel filter with a small window) are best executed directly on the edge device (edge filtering) before transmitting data over the network. This reduces traffic volume, conserves radio module power, and reduces the load on the message broker (e.g., an MQTT broker) receiving data from multiple devices. More resource-intensive methods, such as wavelet noise suppression or ensemble of multiple filters, are best transferred to the edge gateway or to the cloud portion of the system, where the stream, pre-cleaned by edge filtering, is

combined with data from other sensors for deeper analysis. This separation of responsibilities between the device, gateway, and cloud is consistent with the general concept of a multi-tiered edge-cloud architecture for IoT systems and allows for a balance between latency, power consumption, and the quality of the final filtering.

7. Conclusion

This paper systematizes noise sources in streaming telemetry data from IoT devices and examines key preprocessing algorithms: sliding smoothing filters, robust outlier removal methods, a recursive Kalman filter, wavelet noise suppression, and hybrid pipelines. An experimental comparison on a synthetic stream reproducing Gaussian measurement noise and impulse interference showed that none of the "pure" methods is universally optimal: linear filters (SMA, EMA, Kalman) effectively suppress additive noise but are sensitive to outliers, while robust methods (median filters, Hampel filters) are resistant to outliers but less effective against random noise. The hybrid Hampel filter + exponential smoothing pipeline demonstrated the best combination of accuracy (84.1% RMSE reduction), $O(1)$ – $O(W)$ computational complexity, and moderate response latency, making it a practical choice for telemetry preprocessing directly on low-power IoT devices. Further research focuses on adaptively selecting filter parameters (windows, thresholds, smoothing coefficients) based on the current noise level of the communication channel, as well as using lightweight machine learning models for online anomaly detection at the network edge.

Bibliography

1. **Voronkov A. V., Grif M. G., Popov A. A.** Algorithms for primary processing and filtering of data in distributed monitoring systems // Reports of Tomsk State University of Control Systems and Radioelectronics. - 2020. - Vol. 23, No. 2. - P. 45–52.
2. **Kuznetsov A. S., Sergeev M. B.** Method of adaptive filtering of impulse noise in telemetry streams of IoT device parameters // Information and control systems. - 2021. - No. 4. - P. 18–26.
3. **Fedorov D. A., Popov E. V.** Preliminary processing of Internet of Things sensor data based on sliding statistical filters // Control, Communications and Security Systems. - 2022. - No. 1. - P. 112–129.
4. **Grigoriev V. A., Vasiliev A. N.** Algorithms for compression and noise cleaning in streaming data of IoT sensor networks // News of higher educational institutions. Instrument engineering. - 2021. - Vol. 64, No. 8. - P. 635–644.
5. **Ivanov M. S., Petrov I. V.** Cleaning of IoT streaming telemetry data using median filtering and exponential smoothing // Bulletin of Computer and Information Technologies. - 2022. - No. 6. - P. 34–41.
6. **Smirnov A. A., Lebedev D. V.** Application of recurrent Kalman filtering algorithms for processing noisy readings of IoT sensors // Automation and Telemechanics. - 2023. - No. 3. - P. 89–104.
7. **Chernov V. G., Morozov K. E.** Algorithm for detecting anomalies and cleaning noise in telemetry of Industrial Internet of Things (IIoT) devices // Programming. - 2022. - Vol. 48, No. 5. - P. 55–63.
8. **Dmitriev S. P., Makarov A. V.** Pre-processing of IoT temporal telemetry using real-time wavelet transform // Journal of Radio Electronics. - 2021. - No. 9. - Article 7.
9. **Sokolov Yu. A., Kovalev V. N.** Algorithmic support for streaming filtering of outliers in IoT data based on a sliding interquartile range // Proceedings of SPIIRAS. - 2020. - Vol. 19, No. 5. - P. 1021–1045
10. **Belov E. N., Kostin A. G.** Method of filtering and eliminating gaps in telemetry flows of intelligent sensors // Bulletin of Bauman Moscow State Technical University. Series: Instrument-making. - 2023. - No. 2. - P. 78–92.