Lightweight Multi-sensor Modeling for Virtual Braking-Force Sensing

Mariana Fernandes[1], Mariana Almeida[1], Hugo Baptista[1], Suelma Pina[2], Ricardo Teixeira[2], Tiago Vale[2], Cristiana Neto[1], Diana Ferreira[1], Tiago Aston[3], Carlos Ferreira[3], Nuno M. C. da Costa[2], José Machado[1], Susana Bulas Cruz[2], and Hugo Peixoto[1(✉)]

[1] LASI, Algoritmi Center, University of Minho, 4710-057 Braga, Portugal
hpeixoto@di.uminho.pt
[2] DTx–Digital Transformation CoLab, University of Minho, Azurém Campus, 4800-058 Guimarães, Portugal
[3] Electronics and Connectivity (EC) Team, CEiiA, 4450-017 Matosinhos, Portugal

Abstract. We present a lightweight virtual sensing framework for braking force estimation focused on micro electric mobility, where direct pedal force sensing is unavailable due to cost and hardware constraints. It uses a load cell on the braking pedal as ground truth and applies machine learning to low-cost multi-sensor telemetry, combining fixed temporal window aggregation, sparse feature selection through L1 regularisation, and a computationally efficient LightGBM regressor, suitable for real-time embedded deployment. The method is developed and evaluated on the Q-Brake dataset, collected on a micro electric vehicle under diverse but controlled driving conditions. A systematic study examines the influence of window size, sensor modalities, feature selection strategies, and sampling rates. Our analysis shows that inertial measurements dominate braking force related information, with motor telemetry offering complementary benefits. The final 36 feature LightGBM model achieves robust performance on an independent naturalistic trip (RMSE 8.37 N; MAE 4.38 N), demonstrating good generalisation. Due to its bounded per cycle computational cost and hardware friendly design, the proposed virtual sensor is suitable for real time embedded deployment in resource constrained micro EV platforms, enabling scalable and cost-effective solutions for vehicle dynamics monitoring.

Keywords: Braking Force Estimation · Virtual Sensing · Automotive Sensor Fusion · Embedded Machine Learning · Micro Electric Vehicles · Vehicle Dynamics Monitoring

1 Introduction

Braking force is key for understanding vehicle dynamics, ensuring safe operation, and supporting functions such as wear assessment and predictive maintenance. Defined here as the force applied to the brake pedal and measured via a load

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16761, pp. 644–661, 2027.
https://doi.org/10.1007/978-3-032-30527-5_42

cell sensor, it is typically not available in production vehicles, where dedicated sensing hardware is uncommon due to cost and integration complexity.

This is particularly relevant in micro electric vehicles (EVs) and urban mobility platforms, where compact architectures, limited energy budgets, and cost sensitive designs restrict the integration of additional hardware. Their sensor suites are usually limited to basic powertrain telemetry. Low-cost inertial and GPS data may be integrated or available through external devices (e.g. driver's smartphone).

Virtual sensing provides an attractive alternative by estimating otherwise unmeasured physical quantities from existing onboard telemetry, thus enabling cost efficient monitoring. While it has been applied to tyre forces and brake torque, pedal force estimation in micro EVs remains largely underexplored, despite its relevance.

Traditional braking-related estimation approaches have several limitations. Physics-based friction or slip models are sensitive to road conditions, tyre wear, and calibration drift. Observer-based methods demand detailed vehicle dynamics models. Deep learning (DL) imposes computational and memory overhead unsuitable for resource constrained embedded electronic control units (ECUs).

To address these challenges, this work proposes a lightweight virtual sensor for braking force estimation based on supervised learning over multi-sensor temporal windows. Our design focuses on bounded computation per cycle, achieved through fixed-size window aggregation, sparse feature selection, and an efficient tree-based regression model, a structure well suited for embedded deployment while retaining strong predictive power.

The main contributions of this work are:

- A computational virtual sensor for braking force using only low-cost signals.
- A window-based multi sensor modelling strategy with constant per-window computational cost, supporting real time oriented deployment.
- Sparse and interpretable feature selection preserving predictive performance.
- A systematic design space analysis covering algorithm, window size, feature selection strategies, sampling rates and sensor subsets.
- A reproducible and modular machine learning (ML) pipeline tailored to vehicle dynamics virtual sensing.
- Q-Brake dataset, a multi-sensor braking dataset for micro-EVs, enabling reproducible research in lightweight virtual sensing.

The remainder of this paper is organised as follows: Sect. 2 reviews related work; Sects. 3 and 4 describe data collection and processing, respectively; Sect. 5 presents computational methodology; Sect. 6 reports results; Sect. 7 discusses implications and limitations; and Sect. 8 provides conclusions and future work.

2 Related Work

Over the past decades, the automotive industry has extensively focused on improving vehicle control and safety. Virtual sensing techniques have emerged as

a viable solution, estimating unmeasurable variables through software observers implemented in ECUs, reducing system cost and complexity [14]. This paradigm has been applied to a wide range of vehicle dynamic quantities, from longitudinal and lateral tyre force estimation using both model-based and model-less techniques [1], to the simultaneous estimation of tyre forces and braking torques via an LPV Unknown Input Observer validated on a real vehicle prototype [10]. Similarly, brake clamping force estimation in electromechanical brake systems has been addressed using a Kalman Filter Unknown Input Observer combined with nonlinear model predictive control [12]. Although effective, such model based approaches depend on accurate system representations, making them sensitive to parameter uncertainty and limiting robustness across operating conditions.

Several data-driven methods have also been explored for vehicle dynamics estimation. Manghai et al. [2] applied tree-based models such as Random Forest (RF) and Logistic Model Trees for brake-fault diagnosis from vibration signals, achieving high classification accuracy in controlled setups, but relying on small datasets and simplified experimental conditions. Pan et al. [9] proposed a hybrid approach combining a thermal model with a Neural Network (NN) to estimate braking torque, though with limited real-time feasibility. For tyre force estimation, Xu et al. [13] compared NN, RF, and Recurrent Neural Networks using intelligent tyre accelerometers, finding that NN achieved the best accuracy, although performance degrades under extreme conditions, requires specialised in-tyre sensors and controlled testing environments. Barbosa et al. [4] demonstrated accurate lateral force prediction using Gaussian Process Regression, at the expense of high computational cost and limited scalability. More recently, Lu et al. [7] showed that Long Short-Term Memory models outperform traditional observers for tyre force estimation in electric vehicles, although with inherent limitations such as high computational cost, limited scalability, reduced generalisation to unseen driving conditions, and lack of real-world validation.

Since vehicle sensor data typically consists of high-frequency time-series signals, appropriate preprocessing is essential. Window segmentation can be used to transform continuous sensor streams into fixed-length segments suitable for data-driven models [6,8]. Within each window, statistical features such as mean, standard deviation, minimum, and maximum values are extracted to summarise the temporal behaviour of the signals [5,8].

In practical automotive applications, estimation algorithms must operate within ECUs, where strict latency and resource constraints require computationally efficient solutions [11]. While recent ML approaches achieve notable accuracy, many rely on DL architectures with memory and execution requirements incompatible with embedded platforms. To bridge the gap between accuracy and efficiency for ECU deployment, this work adopts a lightweight virtual sensing approach, combining window-based feature extraction with sparse feature selection and an efficient tree-based regressor. This design aims to capture relevant braking dynamics while meeting the computational demands of real time embedded execution.

3 Experimental Setup and Data Collection

This study, focused on micro electric mobility, uses data from Q-Brake dataset [3], collected from a single acquisition session on an L6e electric quadricycle prototype in Matosinhos, Portugal, illustrated on the data collection stage of our methodological pipeline (Fig. 1).

It encompasses 14 trips (total of 23.9 km) under diverse conditions, such as varying number of occupants, their weight ranges, road surface, road inclination, and braking duration and intensity.

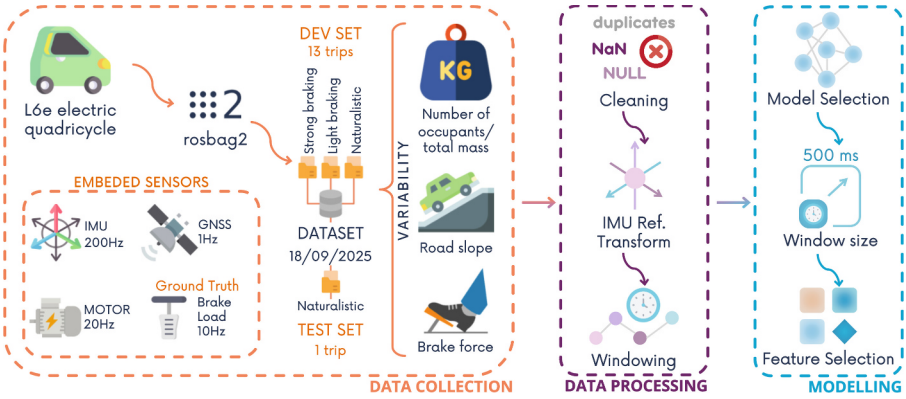

Fig. 1. Overview of the methodological pipeline used in the study. This figure has been designed using resources from Flaticon.com.

3.1 Sensor Instrumentation

We performed data acquisition using ROS 2 (Robot Operating System 2, Humble version) nodes for real time publication and logging of each sensor stream. All sensors were rigidly mounted to minimise vibration artefacts. The collected signals and sampling rates were:

- **IMU**: inertial measurement unit with 3-axes accelerometer and gyroscope data measured in the local X-Y-Z coordinate system of the device [180 Hz];
- **MOTOR**: motor telemetry data from the vehicle CAN (Controller Area Network) bus, including motor performance and EV battery metrics [32 Hz];
- **GNSS**: Global Navigation Satellite System location and velocity data [1 Hz];
- **Brake pedal force (ground truth)**: acquired through a load cell that measures the force that the driver exerts on the brake pedal [10 Hz].

Data streams were stored as rosbag2 files to enable reproducible playback and also converted to CSV for ML workflows.

3.2 Q-Brake Dataset and Data Acquisition Protocol

Because the prototype vehicle cannot operate on public roads, all trips were executed on a 500 m private circuit, shown in Fig. 2 (M). An extended version with an extra loop (E) was also used to introduce additional gradients. Both routes were driven counter clockwise.

This environment allowed consistent yet diverse conditions, including road surfaces (asphalt, cobblestone, and polished concrete), as well as level, uphill, and downhill segments.



Fig. 2. Main Circuit (M) and Extended Circuit with the added detour (E).

The vehicle has three seats, two in the front, and a third in the back, behind the front passenger. To introduce variability in the vehicle's total mass during braking events, passenger configurations and weight ranges were varied, while the driver was fixed, producing masses between 876 kg to 1030 kg. The dataset includes:

- 10 controlled trips (19.1 km) with frequent instructed braking events, ranging from light to strong and also varying in duration (extended circuit);
- 4 naturalistic trips (4.9 km), where braking was performed freely by the driver where appropriate (main circuit), one for each possible number of occupants (1–3), and an additional one with two occupants for final independent testing.

Figure 3 shows how braking force distributions differ across trip types: strong braking trips reach approximately 900 N, while light braking sessions peak near 300 N, and naturalistic trips near 200 N. Light braking trips show more frequent low force inputs (0–50 N), whereas naturalistic trips contain proportionally more moderate forces (50–100 N), reflecting unconstrained driving behaviour.

4 Data Processing

The raw sensor streams were processed through a pipeline to ensure data quality for machine learning experiments. The steps included:

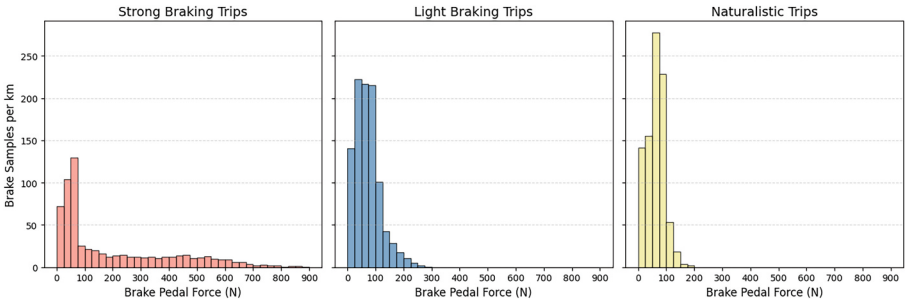

Fig. 3. Brake Pedal Force Distribution Histograms by Trip Type.

1. **Cleaning**: removal of null and duplicate values.
2. **IMU frame transformation**: conversion of IMU measurements from the device's local coordinate system to the vehicle's Front–Left–Up frame, using GNSS speed and heading, and accelerometer data to determine orientation.

After preprocessing, the following variables were available:

- **IMU**: *acc front*, *acc left*, *acc up*, *gyr front*, *gyr left*, *gyr up*;
- **MOTOR**: *current*, *voltage*, *rpm*, *throttle position*, *torque request* and *applied*.
- **GNSS**: *speed* (heading used only for coordinate calibration).

To capture short term temporal behaviour, ensure a fixed computational load during inference without redundancy, and accommodate variable sampling rates, the signals were segmented into fixed size, non overlapping windows. Four statistical features were extracted within a window for each signal: *mean*, *median*, *standard deviation*, and *slope* (defined as the difference between last and first samples), producing 48 aggregated features from the 12 variables (IMU and Motor). GNSS speed was only used in sensor ablation studies due to its sampling rate lower than considered window sizes (producing only mean). The target value for each window was defined as the corresponding average brake pedal force.

Because the analysis focuses on dynamic behaviour, static windows were removed. These were defined as consecutive windows (≥ 3) with GNSS speed below 0.5 m/s, ensuring the retained samples represent active vehicle motion.

5 Computational Method

5.1 Data Partitioning

The dataset was split into a development set of 13 trips and an independent test trip recorded under naturalistic driving with two occupants. The development set included the ten controlled trips with frequent braking events and three naturalistic trips with varying occupancy. These 13 trips were randomly divided into 80% for training and 20% for validation. The independent trip was withheld entirely during model development to provide an unbiased evaluation of the final model.

5.2 Model Training

Data processing and model training were implemented in Python using standard machine-learning libraries. Model selection was performed via 3-fold cross-validation on the training portion of the development set. Validation metrics guided all remaining design choices, and the final selected configuration was retrained on the full development set before evaluation on the independent test trip.

Twelve regression models were evaluated as candidates: CatBoost, XGBoost, LightGBM, Gradient Boosting Regressor (GBR), Random Forest (RF), Extra Trees, Decision Tree, K-Nearest Neighbors (KNN), Stochastic Gradient Regressor (SGR), Support Vector Machine (SVM) with linear kernel, Linear Regression (LR) and Multi-Layer Perceptron (MLP). Tree based ensembles were prioritised due to their favourable balance of prediction accuracy and computational efficiency, making them suitable for embedded deployment.

Hyperparameter tuning explored typical configurations for each model family, including number of estimators, maximum depth, learning rate, and regularisation parameters for tree based regressors; number of layers and units for MLP; and number of neighbours for KNN.

5.3 Experimental Design

A series of controlled experiments was conducted to evaluate the impact of key design choices on braking force estimation. Each experiment varied a single component while keeping all others fixed, enabling an isolated assessment of its contribution to overall performance.

Window Size: The temporal window defines the interval over which sensor data streams are synchronised and aggregated to form each training sample. Window durations from 100 ms to 1000 ms in 100 ms increments were compared. Smaller windows provide higher temporal resolution and lower latency but contain noisier, less stable features, and may capture only partial braking phases. Larger windows offer richer temporal context and improved noise averaging but risk masking brief events and increasing latency. The goal is to identify a window size that balances model performance and system responsiveness.

Feature Selection: To reduce model complexity and improve computational efficiency for embedded deployment, several feature selection techniques were evaluated to identify informative variables and remove redundant ones: L1 regularisation (Lasso), variance-based selection, recursive feature elimination with cross-validation (RFECV), variance inflation factor (VIF), Random Forest Regressor feature importance, and R-regression correlation methods. Each method was assessed using the same model configuration to quantify its effect on dimensionality and predictive performance.

Sampling Rate Variation: Since sensors operate at different native frequencies, an additional experiment assessed the effect of resampling the signals to alternative uniform rates prior to window aggregation. This evaluation examines if lower sampling rates, common in constrained embedded platforms, degrade the quality of the extracted statistical features. Multiple rates were tested while keeping all other components of the pipeline fixed, without feature selection to facilitate interpretation. Comparing performance across these configurations enables understanding the extent to which braking force estimation depends on high frequency dynamics versus aggregated temporal descriptors, and indicates whether the framework remains robust under reduced sensor bandwidth.

Sensor Ablation Study: To evaluate the contribution of each sensor modality, we conducted a systematic ablation study in which sensor subsets were removed or added. Using the final model configuration from previous optimisation stages, without feature selection, different combinations of IMU, motor, and GNSS signals were tested. This analysis identifies the most influential sensor sources, highlights opportunities for reducing hardware requirements, and helps understand expected performance on limited hardware scenarios.

Training Scope Comparison: In embedded deployments, braking instants can be detected from the binary brake light signal available on the CAN bus. Since the Q-Brake dataset does not include this signal, we trained on all dynamic windows, avoiding obtaining a binary indicator directly from the ground truth pedal force to prevent data leakage. To assess the influence of the training-data composition on predictive performance, an additional experiment compared models trained on all windows against models trained exclusively on braking instances. The former reflects scenarios where an external device (e.g. a smartphone) has no access to brake pedal status, while the latter isolates the estimator on braking events.

5.4 Evaluation Metrics

To assess predictive performance of the braking force estimation models, we calculate a set of standard regression metrics. Model selection during cross-validation was based on Mean Squared Error (MSE), since being more sensitive to outliers (due to its quadratic weighting) is critical for identifying potential failures in safety-critical braking estimation. We report the following evaluation metrics to provide a more comprehensive evaluation, as each metric captures different aspects of prediction error:

- **Mean Absolute Error (MAE)**: provides a direct and interpretable measure of prediction error in the original units (N).
- **Root Mean Squared Error (RMSE)**: penalises larger errors while maintaining the same units as the target variable (N).

- **Coefficient of Determination ($\mathbf{R^2}$)**: represents the proportion of variance in the observed target variable (braking force measurements) explained by the model, serving as a standardised measure of goodness-of-fit.
- **Adjusted $\mathbf{R^2}$ ($\mathbf{R^2_{adj}}$)**: modified version of R^2, penalises the inclusion of non-informative predictors, ensuring that performance improvements are due to actual information gain rather than simply increasing the number of features.

To maintain clarity and avoid redundancy, we report a subset of these metrics depending on the experimental context. Particularly, we use R^2 for comparing different ML architectures with a fixed feature set, and R^2_{adj} during feature selection and sensor ablation studies to evaluate the efficiency of the input signals.

6 Results and Evaluation

6.1 Comparative Model Performance

The initial stage of the evaluation consisted of comparing the predictive performance of twelve models. To ensure a consistent comparison across algorithms, all were trained using standard feature normalisation, no feature selection and an intermediate window size of 500 ms. The results are summarised in Table 1.

Hyperparameter tuning explored different configurations for each model. The main search ranges were:

Table 1. Cross-validation performance (mean ± standard deviation).

Model	RMSE	MAE	R^2
LightGBM	**15.32 ± 0.27**	6.48 ± 0.25	**0.957 ± 0.002**
CatBoost	15.57 ± 1.31	6.81 ± 0.73	0.955 ± 0.008
GBR	15.76 ± 0.37	**6.28 ± 0.15**	0.954 ± 0.002
XGBoost	16.59 ± 0.84	6.60 ± 0.13	0.949 ± 0.005
Decision Tree	20.66 ± 0.51	7.87 ± 0.15	0.921 ± 0.004
LR	20.95 ± 0.00	11.77 ± 0.00	0.919 ± 0.000
SVM	22.16 ± 0.34	11.20 ± 0.17	0.910 ± 0.003
RF	22.61 ± 4.19	8.67 ± 1.52	0.903 ± 0.036
KNN	23.97 ± 2.37	10.72 ± 0.68	0.894 ± 0.022
SGR	28.82 ± 9.68	13.84 ± 3.65	0.834 ± 0.113
MLP	30.47 ± 12.07	15.22 ± 5.35	0.808 ± 0.129
ExtraTrees	42.65 ± 6.98	18.65 ± 2.81	0.657 ± 0.111

- Tree-based models: number of estimators/iterations (50–500), maximum depth (3–10), learning rate (0.01–0.2), and regularisation terms ([0, 1]; [1, 10] for CatBoost).

- MLP: number of hidden layers (1–3) and units per layer (10–20).
- KNN: number of neighbours (1–10).
- SVM: regularisation parameter C (0–1).

Among the evaluated models, LightGBM achieved the best overall performance, obtaining the lowest RMSE and highest R^2, and GBR achieved the lowest MAE. CatBoost also demonstrated competitive performance, with results close to LightGBM across all metrics.

Based on these results, LightGBM was selected for the subsequent experiments. Since tree-based models are inherently insensitive to feature scaling, feature normalisation was not applied in the subsequent steps.

6.2 Influence of Window Size

Using the best-performing model from the selection phase, different window sizes were compared, considering their impact on the regression performance.

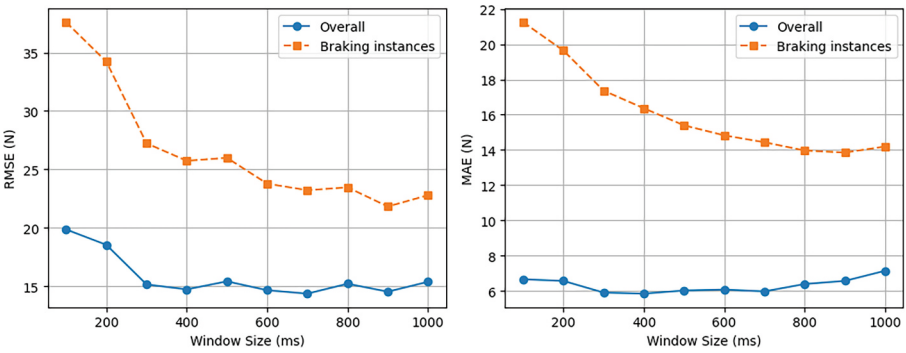

Fig. 4. Effect of temporal window size on model performance (validation set).

The results presented in Fig. 4 show that increasing the window size improves model performance (both RMSE and MAE decrease) when moving from very small windows (100–200 ms) to larger temporal contexts. Beyond approximately 500 ms, the improvements become less pronounced and the performance metrics remain relatively stable. This behaviour suggests that sufficient temporal context has already been captured for reliable feature extraction.

Considering these observations, a window size of 500 ms was selected for the final model and used in the following experiments. This value lies within the region of consistent performance improvement and provides an effective balance between estimation quality and temporal responsiveness.

6.3 Influence of Feature Selection Strategies

The impact of six feature selection strategies on LightGBM regressor performance was evaluated. Table 2 summarises the results, showing only the best performing experiment for each strategy with the respective parameter.

The Variance method achieved the strongest overall performance, yielding the lowest RMSE of 14.48 and the highest R^2_{adj} of 0.966, better than using no feature selection, while selecting 45 features (3 less features). However, since one of the objectives of this work is to reduce model complexity by limiting the number of input features, the Lasso method ($\alpha = 0.2$) was selected for subsequent experiments. Despite presenting slightly lower RMSE and MAE, Lasso achieves a comparable performance while reducing the feature set 25% to 36 variables, offering a better trade-off between accuracy and complexity.

Table 2. Impact of different feature selection strategies on performance (validation set). "Brk" stands for braking instances, "# feat." is the number of selected features.

FS Method	Param.	RMSE	MAE	R^2_{adj}	RMSE (Brk)	MAE (Brk)	R^2_{adj} (Brk)	# feat.
Variance	0.1	**14.48**	5.90	**0.966**	**24.38**	15.37	**0.949**	45
RFECV	2	14.57	5.79	0.965	24.58	15.20	0.948	48
None	–	14.58	**5.76**	0.965	24.58	**15.06**	0.948	48
Lasso	0.2	14.63	6.05	0.965	24.66	15.75	**0.949**	36
VIF	10	15.45	6.38	0.962	25.77	16.34	0.945	23
RF	25	21.21	8.39	0.928	35.31	20.43	0.900	2
R-Regr.	10	30.88	15.58	0.848	48.75	33.17	0.807	10

Regarding the remaining methods, RFECV did not remove any features relative to the no-selection baseline, offering no dimensionality reduction. The VIF, Random Forest Regressor, and R-Regression methods removed the largest number of features, which is reflected in a degradation of predictive performance.

The top 10 feature importance distributions resulting from Lasso ($\alpha = 0.2$) and Variance (threshold = 0.1) are illustrated in Fig. 5. We observe that *acc_front* is the most influential parameter, having the top two features with its mean and median, respectively, and its standard deviation and slope among top 7, in both cases. This is consistent with the underlying physical relationship between longitudinal deceleration and braking force. Motor RPM (mean and slope) and lateral standard deviation of gyroscope are also amongst the most important features for braking estimation.

6.4 Influence of Sampling Rates

Sensor signals may be acquired at different sampling rates, which can affect both the quality of the extracted features and the robustness of the model.

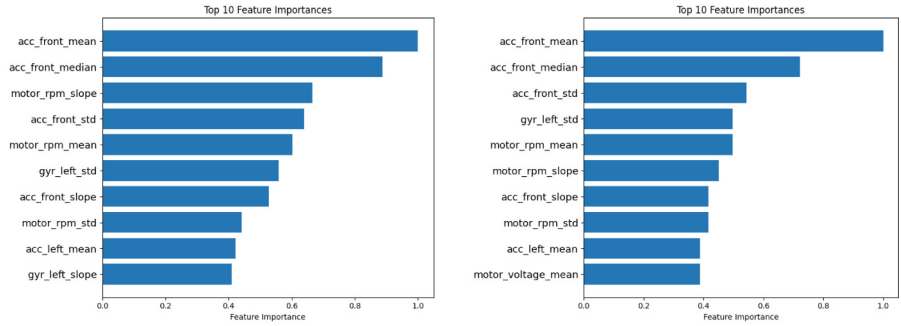

Fig. 5. Feature importance for the LightGBM model using Lasso ($\alpha = 0.2$) (left) and Variance (threshold = 0.1) (right).

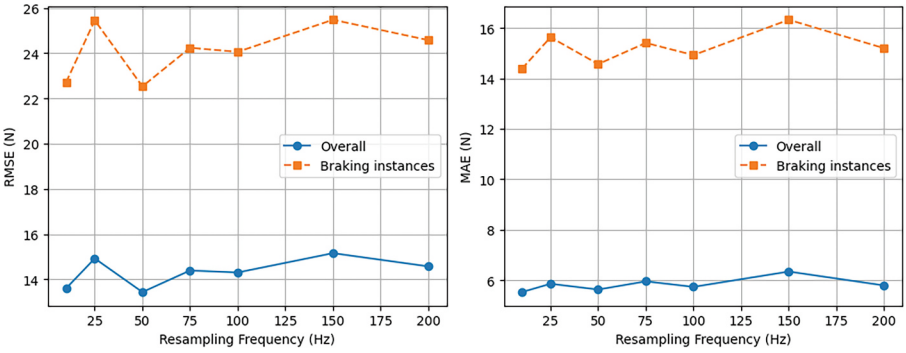

Fig. 6. Effect of sampling rate (after resampling) on model performance.

The reference frequency in this study is 10 Hz, corresponding to the sampling rate of the brake pedal sensor, which provides the target variable used in brake force estimation. As illustrated in Fig. 6, performance remained stable across the tested frequency range, with no meaningful degradation at frequencies as low as 10 Hz. Values of RMSE and MAE on both the full dataset and braking instances show similar insensitivity to resampling rate.

From a practical perspective, this suggests that the framework can operate effectively with moderate sampling rates (≤ 50 Hz), which are commonly supported by low-cost sensors and embedded systems. Consequently, this makes the method compatible with low-cost embedded platforms, where sensor bandwidth is typically limited.

6.5 Sensor Contributions

To analyse the contribution of each sensor modality to the braking force estimation task, several sensor configurations were evaluated with no feature selection

strategy applied. The evaluated configurations include combinations of IMU, motor and GNSS signals, as well as single-sensor variants.

The results in Table 3 show that the combined use of IMU and motor (I+M) achieves the best overall performance, with the lowest RMSE and MAE and the highest adjusted R^2. As we suspected due to its lower sampling frequency than the size of the windows, the inclusion of GNSS (I+M+G) does not improve results when added to any configuration, often leading to slightly worse performance.

Among single-sensor configurations, IMU (I) clearly outperforms motor-based setups (M), indicating that inertial signals capture most of the relevant braking dynamics, while motor signals alone are insufficient. This is consistent with longitudinal acceleration being the most important parameter, and its mean as the most important feature.

Table 3. Comparison of braking-force estimation performance across different sensor configurations, where G=GNSS, I=IMU, M=Motor.

Sensors	RMSE	MAE	R^2_{adj}	RMSE (Brk)	MAE (Brk)	R^2_{adj} (Brk)	# feat.
I+M	**14.58**	**5.76**	**0.965**	24.58	**15.06**	**0.948**	48
I+M+G	15.39	6.47	0.961	**25.04**	15.62	0.948	49
I	17.65	6.96	0.950	29.39	17.30	0.929	24
I+G	20.20	8.48	0.933	32.34	19.00	0.909	25
M+G	26.14	12.64	0.885	40.63	27.47	0.852	25
M	26.51	12.36	0.877	42.11	28.17	0.838	24

The configuration with both sensors was kept for subsequent experiments, providing the best balance between estimation accuracy and sensor complexity.

6.6 Effect of Training Scope

To analyse the impact of training strategy on braking force estimation, a comparative study was conducted considering different model configurations: using all dynamic windows ("Dynamic") and using only braking instances ("Brk").

The results show that the LightGBM model trained on all dynamic windows outperforms the braking-only variant on braking instances by a small margin, suggesting that exposure to the complete range of driving dynamics provides a beneficial regularisation effect.

While training solely on braking windows, we decided to apply a simplified Linear Regression (LR) model as a low-complexity alternative and benchmark to the proposed model. It was applied using the mean longitudinal acceleration (*acc_front_mean*) as the sole predictor, based on our previous feature importance analysis and reflecting the underlying linear physical relationship between vehicle deceleration and braking force. The results are summarised in Table 4.

Table 4. Comparison of training strategies for braking force estimation on all active vehicle motion windows (Dyn: dynamic) and windows containing braking events (Brk).

Model	RMSE	MAE	R^2_{adj}	RMSE (Brk)	MAE (Brk)	R^2_{adj} (Brk)	# feat.
LightGBM Dyn	14.63	6.05	0.965	24.66	15.75	0.949	36
LightGBM Brk	–	–	–	25.05	16.06	0.948	37
LR Brk	–	–	–	35.69	22.07	0.894	1

The linear regression model, despite its simplicity, achieves an adjusted R^2 of 0.894, confirming that longitudinal acceleration is a meaningful predictor. As a single-feature model, it captures the dominant physical trend but exhibits substantially larger errors, with RMSE and MAE higher by about 11 N and 6 N, respectively, compared with the more complete LightGBM model. This indicates that a single-feature linear formulation is insufficient to capture the complexity of braking force dynamics. Figure 7 illustrates the behaviour of both models on active braking instances.

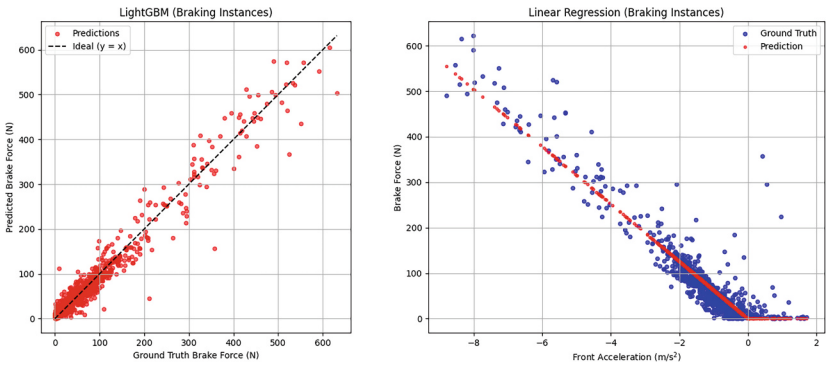

Fig. 7. LightGBM (left) prediction vs ground truth comparison; Linear Regression (right) relationship between front acceleration and braking force for active braking instances.

On the left, the LightGBM shows strong agreement with ground truth values, as evidenced by the alignment along the identity line, despite increased variance at higher forces. Conversely, the Linear Regression (right) exhibits a clear correlation between braking force and front acceleration. However, its univariate nature prevents the model from capturing the non-linear complexities present in the dataset.

6.7 Independent Test Set Results

This section assesses the performance of the final model on previously unseen data in order to evaluate its generalisation capability beyond the training data. The final model corresponds to a LightGBM trained on the full development set with a temporal window of 500 ms, and Lasso feature selection ($\alpha = 0.2$). Model training resulted on the following hyperparameter configuration: Gradient Boosting Decision Tree as the boosting algorithm, 20 leaves, maximum depth of 10, learning rate of 0.168, and no regularization (reg_alpha = 0, reg_lambda = 0).

Table 5. Comparison between cross-validation (best validation fold) and independent test set performance on the final LightGBM model with 500 ms windows and 36 features.

Set	RMSE	MAE	R^2	RMSE (Brk)	MAE (Brk)	R^2 (Brk)
Validation	15.8	6.30	0.958	–	–	–
Test	8.37	4.38	0.912	13.94	10.98	0.803

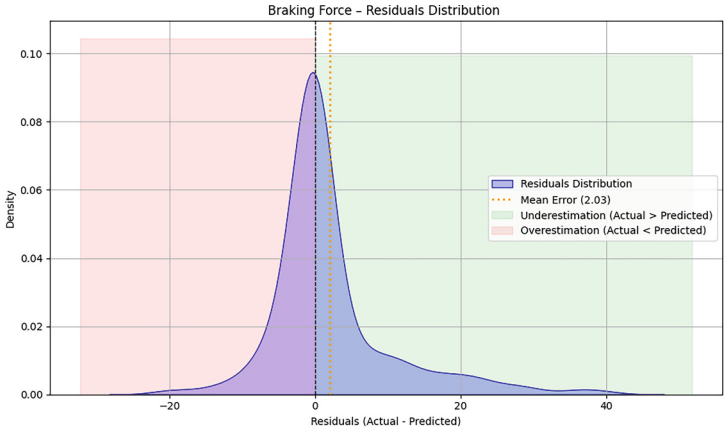

Fig. 8. Distribution of prediction residuals of the LightGBM braking force estimation model on the independent test set.

As shown in Table 5, the model achieves significantly lower RMSE and MAE on the independent test set than on the best fold of CV (with the same hyperparameters), while also presenting lower values on R^2. This happens because the independent test trip contains only light naturalistic braking, leading to substantially lower variance in the braking force target compared with the controlled braking sessions used for training and validation (see Fig. 3), which include very

strong braking events. Although the model achieves lower absolute errors on this easier regime (MAE and RMSE decrease), the reduced variance of the ground truth lowers the achievable R^2, explaining the slightly lower R^2 value despite improved error metrics.

Results indicate good generalisation and adequate errors on unseen data both on all dynamic windows and on braking instances only, although the latter show higher errors as expected. To further analyse the model behaviour, the distribution of prediction residuals is illustrated in Fig. 8.

The residuals are centred close to zero, although a slight positive bias can be observed, suggesting a tendency of the model to underestimate higher braking forces. This behaviour is consistent with the inherent difficulty of modelling sharp braking transitions using aggregated sensor features.

7 Discussion

The results demonstrate that the proposed virtual sensing framework provides a reliable braking force estimation using only existing vehicle telemetry and lightweight architectures. The superior performance of LightGBM over other ensemble methods and NN validates the use of tree-based architectures for automotive virtual sensing. From an embedded systems perspective, the inference complexity of a decision tree is $O(d)$, where d denotes the maximum tree depth, ensuring deterministic latency and a low memory footprint. Furthermore, the model uses 36 statistical features and since LightGBM is invariant to feature scaling, the normalisation step can be omitted in production, simplifying the software pipeline and reducing the computational cost per window.

The simplified linear model captures a large portion of the variance in braking force, indicating that the target signal is strongly influenced by fundamental vehicle kinematics. Longitudinal acceleration exhibits a direct physical relationship with braking force, which explains why a single-variable model can approximate it reasonably well. However, the considerably lower prediction errors reveal that our LightGBM model effectively better captures the nonlinear and multi-sensor dependencies present in real braking dynamics.

Nevertheless, several limitations must be acknowledged to contextualise the scope of these findings. First, our dataset was collected using a single micro-EV on a private circuit, which, despite controlled variation in load, slope, and braking behaviour, does not encompass the diversity of real urban traffic, multiple drivers, or adverse weather conditions. Further validation across additional vehicles, routes, and environments is needed to establish broader generalisability. These controlled acquisition conditions represent the primary limitation of this work. Second, although the framework is explicitly designed for resource-constrained systems, we have not yet conducted runtime benchmarking on embedded hardware. Evaluating latency, memory usage, and energy consumption on representative automotive edge devices will be essential to fully confirm real-time feasibility. Finally, the absence of a CAN-based brake-light signal limited our ability to isolate braking-only intervals without deriving indicators from

the target itself. While we implemented a braking-window variant for comparison, future studies could incorporate such signals to strengthen training-data selection and further reduce noise associated with non-braking intervals.

8 Conclusion and Future Work

This work presents a lightweight virtual sensing framework for braking force estimation using multi-sensor vehicle telemetry and ML. It uses data from Q-Brake dataset, collected on a L6e quadricycle, representative of micro electric mobility, a context where this is particularly relevant given the stringent hardware constraints typical of this vehicle category. The proposed approach combines window statistics, sparse feature selection through L1 regularisation, and a LightGBM model to provide accurate braking force estimation while maintaining computational efficiency suitable for real-time embedded systems.

The analysis of sensor modalities showed that inertial measurements provide most of the relevant information for braking force estimation, while the combination of IMU and motor telemetry yielded the lowest prediction errors. The evaluation of a minimal linear baseline also showed that longitudinal acceleration alone explains a significant portion of braking force variability, although the nonlinear LightGBM model substantially reduces prediction errors by capturing additional dynamics, including effects associated with regenerative braking.

The results further indicate that the proposed framework remains robust across moderate sampling rates, confirming its compatibility with resource-constrained embedded platforms. Evaluation on an independent test trip demonstrates good generalisation capability, achieving an RMSE of 8.37 N, a MAE of 4.38 N and an R^2 of 0.912, although prediction errors increase on braking events due to the higher variability of these dynamics. However, the test trip covers mainly light braking, so these results reflect a less demanding regime than training.

Although the proposed design is computationally efficient by construction, real-time latency has not yet been experimentally benchmarked on target embedded hardware. Future work will focus on expanding the dataset to include multiple micro-EV platforms, a wider range of driving environments, such as urban traffic conditions, different drivers and driving behaviour, and adverse weather scenarios. Furthermore, the proposed virtual sensor will be deployed within a ROS-based pipeline on an edge device to assess real-time performance, including latency and resource usage under embedded constraints.

Acknowledgments. This work has been supported by the European Union under the NextGenerationEU, through a grant of the Portuguese Republic's Recovery and Resilience Plan (PRR) Partnership Agreement, within the scope of the project BE.NEUTRAL – "Agenda da Mobilidade para a neutralidade carbónica das cidades", aiming at the development, industrialization and operation of new carbon-zero mobility cyber-physical products that will accelerate the transition to carbon-neutral cities by 2030 (Project ref. nr. 35 - C644874240-00000016; Total project investment: 221.376.867,63 Euros; Total Grant: 128.619.400,50 Euros).

References

1. Acosta, M., Kanarachos, S., Blundell, M.: Virtual Tyre force sensors: an overview of Tyre model-based and Tyre model-less state estimation techniques. Proc. Inst. Mech. Eng. Part D: J. Automob. Eng. **232**(14), 1883–1930 (2018). https://doi.org/10.1177/0954407017728198
2. Alamelu Manghai, T.M., Jegadeeshwaran, R.: Vibration based real-time brake health monitoring system: a machine learning approach. In: IOP Conference Series: Materials Science and Engineering. vol. 624, no. 1, p. 012027 (2019). https://doi.org/10.1088/1757-899X/624/1/012027
3. Baptista, H., et al.: Q-brake: L6e quadricycle multi-sensor braking-dynamics dataset (2026). https://doi.org/10.17632/4m5h6bv5s4.1
4. Barbosa, B.H.G., Xu, N., Askari, H., Khajepour, A.: Lateral force prediction using gaussian process regression for intelligent tire systems. IEEE Trans. Syst. Man Cybern. Syst. **52**(8), 5332–5343 (2022). https://doi.org/10.1109/TSMC.2021.3123310
5. Hirawat, A., Taterh, S., Sharma, T.K.: A dynamic window-size based segmentation technique to detect driver entry and exit from a car. J. King Saud Univ. – Comput. Inf. Sci. (2022). https://doi.org/10.1016/j.jksuci.2021.11.006
6. Lesti, G., Spiegel, S.: A sliding window filter for time series streams. In: Proceedings of the ECML/PKDD Workshop on IoT Streaming. CEUR Workshop Proceedings, vol. 1958 (2017)
7. Lu, L., Xu, G., Chen, X.: Tire force estimation using model-based and machine learning approaches for distributed drive electric vehicles. In: Proceedings of the 8th CAA International Conference on Vehicular Control and Intelligence (CVCI). IEEE (2024). https://doi.org/10.1109/CVCI63518.2024.10830206
8. Mozaffari, L., Mozaffari, A., Azad, N.L.: Window-based feature extraction framework for multi-sensor data: a posture recognition case study. In: IEEE International Conference on Systems, Man, and Cybernetics. IEEE (2015). https://doi.org/10.1109/SMC.2015.7321471
9. Pan, Y., Sun, Y., Li, Z., Gardoni, P.: Braking torque estimation through machine learning algorithms. Mater. Res. Proc. **26**, 213–218 (2023). https://doi.org/10.21741/9781644902431-35
10. Sentouh, C., Fouka, M., Popieul, J.C.: Virtual sensor: simultaneous state and input estimation for nonlinear interconnected ground vehicle system dynamics. Sensors **23**(9), 4236 (2023). https://doi.org/10.3390/s23094236
11. Wang, J.: Real-Time Embedded Systems. John Wiley & Sons, Hoboken, NJ, USA (2017)
12. Wei, W., He, T., Men, Y., Wang, J., Xiong, L.: Nonlinear model predictive clamping force control for electromechanical brake system. IEEE Int. Conf. Adv. Robo. (2024)
13. Xu, N., Askari, H., Huang, Y., Zhou, J., Khajepour, A.: Tire force estimation in intelligent tires using machine learning. IEEE Trans. Intell. Transp. Syst. **23**(4), 3565–3574 (2022). https://doi.org/10.1109/TITS.2020.3038155
14. Zaharia, C., Clenci, A.: Study on virtual sensors and their automotive applications. Sci. Bull. Automotive Series (2013)