

Sensor-Fusion Baselines and Robustness Envelopes for Real-Time UAV Obstacle Detection on Embedded Platforms

Abdullah Thair Al-Obaidi* 

Department of computer Engineering, College of Engineering, University of Diyala, Baqubah, Diyala, Iraq. abdullah.thair@uodiyala.edu.iq

ARTICLE INFO

Article history:

Received: 11 /05/2026

Revised form: 28 /06/2026

Accepted : 05 /07/2026

Available online: 30/09/2026

Keywords:

UAV
Obstacle Detection
Sensor Fusion
LiDAR
XGBoost
LightGBM
Embedded AI

ABSTRACT

While achieving effective obstacle identification and secure navigation under computational limitations, autonomous unmanned aerial vehicles (UAVs) rely on precise sensor fusion. This paper explores a wide-ranging benchmark and robustness analyses of eight artificial intelligence (AI) models over a multi-sensor UAV navigation dataset, including Light Detection and Ranging (LiDAR), GPS, IMU, and telemetry data. In this paper, standard machine-learning (logistic regression, SVC, k-NN, and Gaussian Naive Bayes) classifiers are compared with ensemble learners (Random Forest, XGBoost, and LightGBM) and a shallow MLP. The accuracy, F1-score, recall, AUC, calibration, inference latency, and memory footprint of each model are evaluated to measure their realistic baselines for onboard implementation. Under chronological time-series evaluation, the tree-based ensembles achieved the highest obstacle-detection performance ($F1 \geq 0.98$) with strong computational efficiency, while linear and instance-based models performed markedly lower. Robustness envelopes constructed under LiDAR noise and sensor ablation scenarios reveal the operational boundaries of sensor-dominant regimes and highlight the importance of redundancy in sensor fusion. The results explained with SHAP show that the main decision variables are LiDAR distance and IMU dynamics. The framework provides repeatable baselines, measures of effectiveness, and safety-aware decision-making that can inform future efforts in energy-efficient and interpretable UAV autonomy.

MSC..

<https://doi.org/10.29304/jqcm.2026.18.32882>

1. Introduction

UAVs have become an essential component of today's automated systems because they can perform complex tasks such as inspecting infrastructure, monitoring the environment, implementing precision farming techniques, and responding to emergencies while travelling very efficiently [1]. To operate safely in dynamic and unstructured environments, UAVs must possess sufficient sensor capabilities to detect objects and obstacles in real-time [2]. Therefore, UAVs require the integration of multiple sensor types; they will rely on a variety of sensors (e.g., LiDAR, GPS, IMU Telemetry) to create a single perception stack containing all sensor data [3]. This type of sensor integration enables UAVs to maintain accurate situational awareness in the event of changing weather conditions, noise from the various sensors, or partial data loss [4].

While the above combination provides several advantages, it also has computational challenges (i.e., cross-subject real-time inference speed/ trade-offs with model memory footprint) when using real-time conditions, such as those commonly found in embedded platforms [5]. Artificial intelligence (AI) and machine learning (ML) algorithms now

*Abdullah Thair Al-Obaidi

Email addresses: abdulla.thair@uodiyala.edu.iq

Communicated by 'sub editor'

make up the majority of intelligent UAV navigation, overcoming the limitations of rule-based or geometric techniques used to navigate UAVs with complex and nonlinear dynamics [6]. Classical ML algorithms, such as logistic regression and support vector machines (SVMs), have proven effective at solving binary decisions, while ensemble ML algorithms, such as random forest, XGBoost, and LightGBM, exhibit superior performance at accommodating sensor heterogeneity and nonlinear separability tasks [7]. Deep learning has recently extended this application domain by employing neural network (NN) architectures that learn complex spatiotemporal representations from sensor observations [8]. However, issues of inference latency, resource limitations, and the need for interpretable models that can support certifiable autonomous operations continue to present challenges for deploying these models on UAVs in an embedded environment [9].

UAV perception has become a rapidly expanding field of research with many studies on UAV performance, mainly for accuracy; however, few studies addressed UAV performance under all conditions with multiple sensors/models, including their interpretive capability and their efficiency when embedded within an overall system [10]. Furthermore, there has been a dearth of data regarding the benchmark performance of these disparate types of AI models in the context of preprocessing and evaluation methods applied over heterogeneous sensor-fusion datasets [11], [12]. Therefore, while UAVs must accurately classify objects, their ability to perform accurately in the presence of noise, generalize across domains, and efficiently perform computations directly influences the reliability of UAV obstacle detection [13]. Therefore, there is a need for standardized, reproducible evaluations that account for the full range of trade-offs between predictive accuracy and deployability in UAV applications [14].

This study presents a comprehensive comparative analysis of eight AI models applied to a multi-sensor UAV navigation dataset that integrates LiDAR, GPS, IMU, and telemetry signals. The proposed framework benchmarks classical, ensemble, and neural architectures while assessing their performance through accuracy, F1-score, recall, AUC, calibration, latency, and model size. Robust envelopes are constructed under controlled LiDAR noise and sensor-ablation scenarios to delineate operational safety boundaries for UAV perception systems. Also, an exploration of model interpretability made by means of SHAP analysis is presented, indicating the relative importance of each sensor for making a decision. The results create reproducible, speed-aware baselines for UAV autonomy on embedded systems and show how such AI models can be adapted for efficient real-time deployment while maintaining reliability and transparency.

2. Related works

Numerous systems have been created for UAVs to permit obstacle detection and avoidance, done using traditional methods like computer vision, LiDAR processing, and sensor fusion methods [2]. The first experimental studies used either monocular or stereo vision to produce a depth and motion perceived from the camera image, the advantage of these first methods being they were relatively cheap, but very dependent on light (need sufficient light), plus they required a high level of computation on the vehicle [15],[16]. The way to overcome these limitations is with an alternative dominant sensing method based on the combination of LiDAR and cameras providing high precision distance measurements that produce consistent results regardless of the intensity of light the measurements were made under [17]. Additional experiments have added additional distance measurement sources like IMUs, GPS, and barometric sensors to use with LiDAR to provide additional spatial awareness in dynamic environments for autonomous navigation and landing [18].

UAV-based perception systems have gradually transitioned from using handcrafted methods of feature extraction, to adopting models based on machine learning and AI technologies [19]. Classical algorithms (such as support vector machines (SVMs), Decision Trees and Naive Bayes) are capable of reasonably performing obstacle detection in structured environments [20]. Ensemble-based methods (e.g., Random Forests, Gradient Boosting, XGBoost, and LightGBM) can also help in overcoming noise and heterogeneity from sensors across input space, resulting in more stable predictions [21]. More recently, an increasing number of researchers have used deep neural networks (DNNs), targeting convolutional and recurrent architectures with multimodal UAV datasets, to create models capable of learning spatial-temporal representations directly from the source data. While DNNs produce more accurate classification than classical classifiers, they are difficult to implement in embedded applications due to resource constraints [22].

Apart from prediction performance, a few research was focused on interpretability and efficiency for autonomous aerial systems [23]. SHAP, LIME, and other explainable AI (XAI) frameworks have been integrated into UAV perception models to ensure transparency during safety-critical decision making [24]. Exploiting parallel research

fields, energy-aware learning and model compression (quantization, pruning and knowledge distillation) have been explored to satisfy real-time constraints on edge devices [25]. However, despite these advancements, very few works evaluate all the aforementioned aspects collectively under a single unified framework for detection accuracy, robustness under sensor degradation, explainability, and embedded feasibility [26]. The majority of benchmarks are also still limited in scope, emphasizing either algorithmic novelty or specific modalities, thus missing a (crucial) higher level of system-level evaluation for practical deployment [27].

This study presents an end-to-end benchmark framework pipeline for sensor-fusion-based UAV obstacle detection that harmonizes classical, ensemble, and neural models under a single preprocessing and evaluation framework. The work builds upon previous research by providing a complete end-to-end evaluation of performance metrics, calibration, interpretability, robustness to LiDAR noise and sensor ablation, and computational efficiency. The authors also present a unified framework to effectively link research in algorithms with practical applications, providing useful insight into determining the suitability of AI-based models for embedded platforms on UAVs.

3. Dataset and Preprocessing

The autonomous navigation and obstacle avoidance sensor dataset for the UAV used in this study provides a realistic multimodal basis for assessing autonomous flight [28]. This dataset is composed of synchronized readings from LiDAR, GPS, IMU, and telemetry sensors, representing the diverse sensory configuration that small- to mid-scale UAVs possess when operating autonomously. A record corresponds to a timestamped UAV shape containing geographic coordinates (latitude, longitude, height), inertial measurements (accelerator and gyroscope), LiDAR range to the nearest point, flight speed, wind speed, and battery status, and is paired with a binary obstacle-detection label. This varied input organization allows to investigate how multiple sensors contribute collectively to inform obstacle-awareness mechanisms in flight under dynamic conditions. The dataset has a mixed nature where it includes urban, forest and mountainous terrains through both real flight data as well as simulation to balance evaluations of how well a model generalizes or deals with general disturbance. This variety, rather than being seen as a negative aspect, allows for evaluating algorithm performance across a range of environmental and signal characteristics. A natural occurrence of the above mentioned issues is that any obstacles encountered are less likely to exist, but more critical to the operation of either the UAV or human life. In the obstacle detected class (Positives) 4.72% of the total number of 5,000 records are collected in a continuous sequence of flight at 1 Hz. An obstacle defined operationally via Lidar proximity with obstacle frames identified via a LiDAR distance less than 5m. Therefore, the class imbalance was preserved during training in order to reflect true prior probabilities as closely as possible while providing class-balanced loss formulations and using stratified splits wherever possible.

Prior to training a model, data went through various preprocessing stages in order to equalize their signal and decrease cost for computation. The data was treated for outliers and missing data through the use of median imputation per sensor group, followed by standard normalisation through standard scaler transformation for numerical stability of all feature types. The raw data maintained its chronological ordering to preserve the physical causation of the data and had no train-test set leakage of data. The analysis of feature correlation suggested that LiDAR distances, IMU acceleration, and UAV speed were the three most significantly contributing variables to class separation and hence included as primary predictive variables in the analysis of features. In addition, whilst wind speed and battery levels have some degree of correlations respectively with the proximity of obstacles, these two variables had very weak correlations with each other therefore both were kept in the analysis to not compromise the ecological validity of the types of sensor fusion patterns generally observed with real-world UAVs.

The dataset is a continuous time series that was split in chronological order using the first 80% of the data as a training set and the last 20% as a testing set (no shuffling). Five-fold Timeseries Split cross-validation was used to assess the consistency of model performance; all five validation sets were strictly later in time than the training data used for those five validations sets in order to avoid temporal leakage between adjacent and highly correlated recordings. Noise-injection and sensor-ablation experiments were then conducted to evaluate performance limits. This structured approach to data preprocessing provides reproducibility of downstream analyses, statistical validity, and compliance with embedded AI deployment constraints.

4. Methodology

This section will describe the experimental design, which includes data representation, model architecture, training design, and evaluation criteria. In addition to standardized performance tests, the methodology also uses embedded evaluation tests and model interpretability tests to connect academic metrics to the practical application of unmanned aerial vehicle (UAV) systems. All of the methodology was designed to be reproducible, provide transparency in their computational implementation, and enable to test the models at a real-time embedded autonomous system environment. Evaluating models solely based on accuracy is insufficient, as UAV's lack of onboard resource capability necessitate lower-latency, lower-power models that can be constructed on the airframe while still providing predictive performance. Therefore, analyses of the volume of the data, inference response time and predictive performance of each trained model have been used to determine edge-compatibility for deployment. The resulting trade-off curve provides a means to select the appropriate model candidates for deployment on platforms such as NVIDIA Jetson and Raspberry Pi, enabling possible real-time applications. In addition, the use of SHAP (Shapley Additive Explanations) will provide a quantitative value on the contribution of each sensor to the output prediction from the model. This ability to explain the model will assist with safety certification, fault diagnosis, and user trust in the autonomous decision-making process. This also ensures that sensor-fusion models maintain transparency and their respective accountability in highly sensitive UAV applications.

4.1 Data Representation

The processed data was converted into a numerical matrix suitable for supervised learning. Each sample consisted of 13 continuous features representing various outputs from the UAV's sensors. The sensors include: 3 outputs from the 3D accelerometer and 3D gyroscope on the UAV; 1 output from the LiDAR sensor providing measurement of distance to the nearest obstacle; 1 output from the GPS providing information on the altitude of the UAV; 1 output from the GPS providing the speed of the UAV; output providing wind speed experienced by the UAV 1; and 1 output representing the battery level of the UAV. The two possible classes of the binary target label "obstacle_detected" were assigned values of 1 (for presence of the obstacle) or 0 (for no obstacle). Due to the vast range and large varying scales between the sensors; all feature values had their z-score normalised. No dimensionality reduction was performed in order to retain interpretability and allow fair comparisons to be made between linear, tree-based and neural models. The actual operational sequences were retained within the same partition with respect to the time order of the original data collection. The final feature matrix provides a balanced and dense representation of the input data in an optimised manner for both classical and deep learning architectures.

4.2 Model Architectures

Eight AI models were selected to capture diverse algorithmic families and decision mechanisms:

- **Classical Models:** Logistic Regression (with class weighting) and Support Vector Classifier (RBF kernel) provided interpretable baselines for linearly and non-linearly separable data.
- **Probabilistic Model:** Gaussian Naive Bayes captured low-complexity probabilistic relationships and served as a lightweight reference.
- **Instance-Based Model:** k-Nearest Neighbors ($k = 5$, distance-weighted) was used to examine performance sensitivity to local neighborhood density.
- **Ensemble Models:** Random Forest, XGBoost, and LightGBM were employed for their robustness to feature heterogeneity, parallelization efficiency, and superior generalization capacity.
- **Neural Model:** A Multi-Layer Perceptron (MLP) and two hidden layers (64-32 neurons, ReLU activation, dropout = 0.2) was trained to model non-linear sensor dependencies.

Each model will provide probability outputs enabling thresholds to be set uniformly, use of AUC as measure of ability. Ensemble methods have modelled the feasibility of embedded deployments by applying early stopping at a shallow depth; while MLP has an adaptive learning style by employing Adam optimization method. Figure 1 outlines the workflow employed in this study to outline the different phases of experimentation: Data Acquisition, Data Preprocessing, Model Training, Model Evaluation, Embedded Profiling and Explainability Analysis.

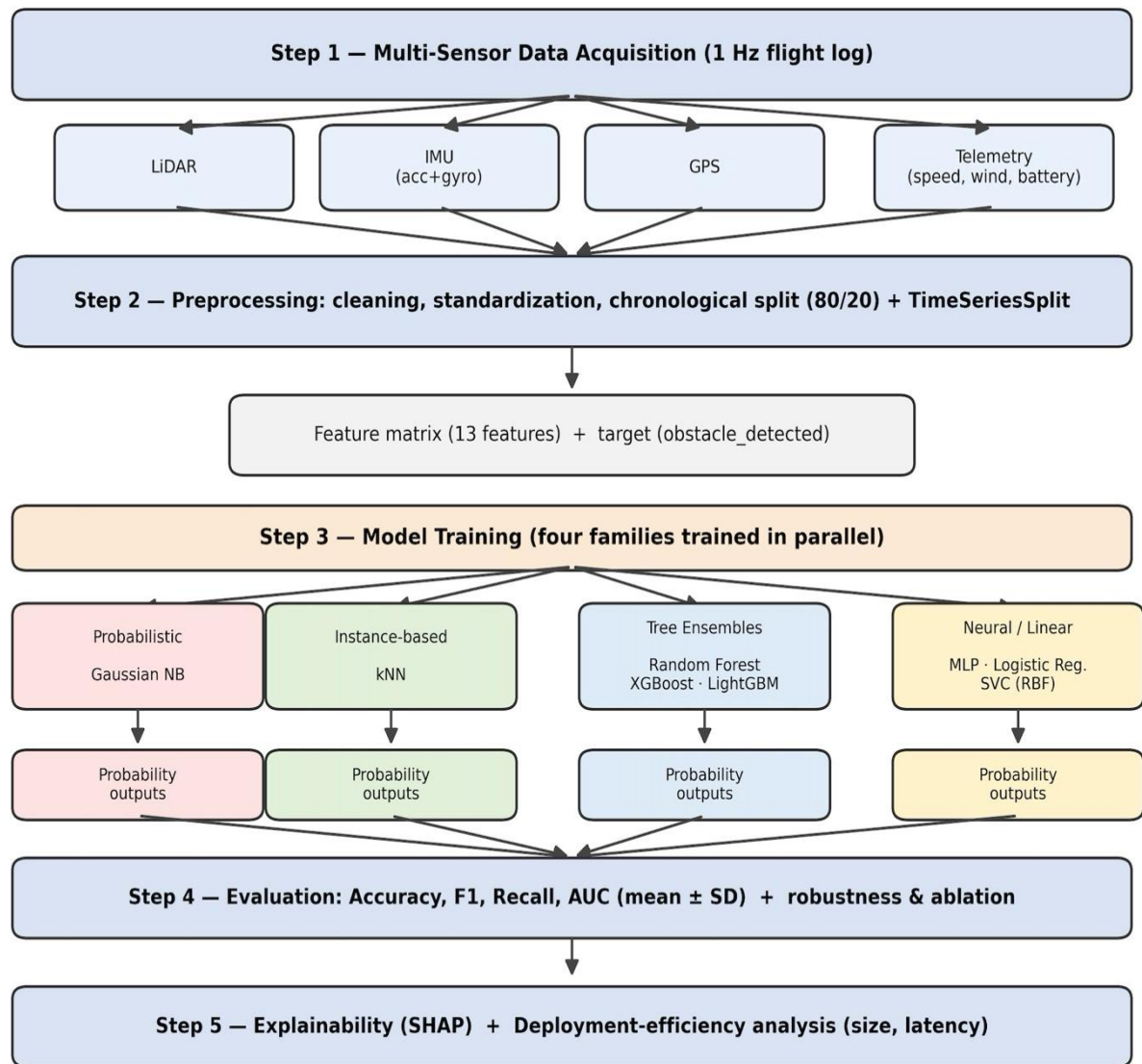


Fig. 1- Workflow of the proposed UAV obstacle-detection methodology. The four model families in Step 3 are trained in parallel and converge to a unified evaluation stage.

4.3 Training Configuration

A reproducible pipeline implemented in Python 3.12 with Scikit-learn, LightGBM, and XGBoost libraries was used for training and evaluation. This involved splitting the data into training (80%) and testing (20%) subsets, preserving temporal order. F1-score variance was estimated using (five-fold) Timeseries Split cross-validation.

To save time and resources, hyperparameters were fine-tuned using a grid search across an idealized range based on how they would eventually be optimized in practice. All of the input data for all models was normalized, oversampling was done wherever applicable and weights were relatively balanced across each training set. For stochastic models (e.g., XGBoost, LightGBM, and MLP) an average of five trainings were run for each model to ensure stability. All of the hyperparameter settings and search ranges for each model can be found in Table 1. The threshold(s) for each model were obtained by maximizing the F1 score for the positive class (obstacle) on the validation data set.

Table 1- Hyperparameter search ranges and final values for the eight classifiers.

Model	Search range explored	Final value(s))
Logistic Regression	$C \in \{0.1, 1, 10\}$	$C = 1.0$; penalty = L2; class_weight = balanced; max_iter = 2000
SVC (RBF)	$C \in \{0.1, 1, 10\}$	$C = 1.0$; kernel = RBF; gamma = scale; class_weight = balanced
k-NN	$k \in \{3, 5, 7, 11\}$	$k = 5$; weights = distance
Gaussian NB	var_smoothing (default)	var_smoothing = $1e-9$
Random Forest	n_estimators $\in \{100-500\}$	n_estimators = 300; max_depth = None; class_weight = balanced
XGBoost	max_depth $\in \{3-6\}$; lr $\in \{0.01-0.1\}$	n_estimators = 300; max_depth = 4; lr = 0.05; subsample = 0.9; colsample_bytree = 0.9; reg_lambda = 1.0; scale_pos_weight = auto
LightGBM	num_leaves $\in \{15-63\}$; lr $\in \{0.01-0.1\}$	n_estimators = 500; num_leaves = 31; lr = 0.05; subsample = 0.9; colsample_bytree = 0.9; reg_lambda = 1.0; class_weight = balanced
MLP	hidden $\in \{(64,32), (128,64)\}$	hidden = (64, 32); activation = ReLU; alpha = $1e-4$; batch_size = 32; lr_init = $1e-3$; solver = Adam; max_iter = 300

4.4 Evaluation Metrics

The model's performance was evaluated using several complementary metrics to ensure the accuracy, robustness, and efficiency of the results. The main metrics used in the evaluation included accuracy, F1 score, recall rate, and Area Under the ROC Curve (AUC) to assess classification effectiveness. Given the critical role of the UAV's navigation system in user safety, the recall rate for obstacle classification was considered the most important reliability metric.

To gather information about deployment vetting, we recorded additional measurements (model size in MB, inference latency(ms/sample)), both measured on a standardized CPU environment. We created a combined efficiency ranking to show where there is balance in terms of performance vs required computation. We then used SHAP explainability to identify the main contributors (i.e., LiDAR distance and IMU dynamics) using feature-attribution maps that allow for interpretability of the evaluation and support real-world UAV safety validation.

5. Results and Discussion

The section presents an overall description of the UAV obstacle-detection and autonomous-navigation architecture with an emphasis on the context-based versatility of machine-learning algorithms, sensor perturbation resilience, model training behavior, and interpretability. The tests in this section were performed based on the multi-sensor UAV dataset introduced in Section 3 according to the pipeline in Figure 1 above:

5.1 Model Performance Overview

The quantitative performance of all evaluated models is summarized in (Figures 2-4), where accuracy, F1-score, and recall for the obstacle-detected class are presented.

- The ensemble learners XGBoost, LightGBM, and Random Forest achieved near-perfect results (F1 = 1.000, 0.981, and 0.979 ± 0.02 , respectively), validating their capacity to model the complex nonlinear relationships between LiDAR distance, IMU dynamics, and environmental features.
- The logistic regression model achieved high recall but lower F1 (0.585 ± 0.05) while maintaining its lightness and ease of interpretation.
- MLP, SVC, and KNN exhibited slightly lower scores due to class-imbalance sensitivity, with F1 scores of 0.57, 0.53, and 0.03, respectively.

- Gaussian Naive Bayes performed exceptionally because a single dominate feature (LiDAR distance) nearly separates the classes.
- A correlation audit in Figure 5 shows LiDAR distance has the dominant but non-deterministic association with the target ($r = -0.36$), while all other sensors have $|r| < 0.05$.

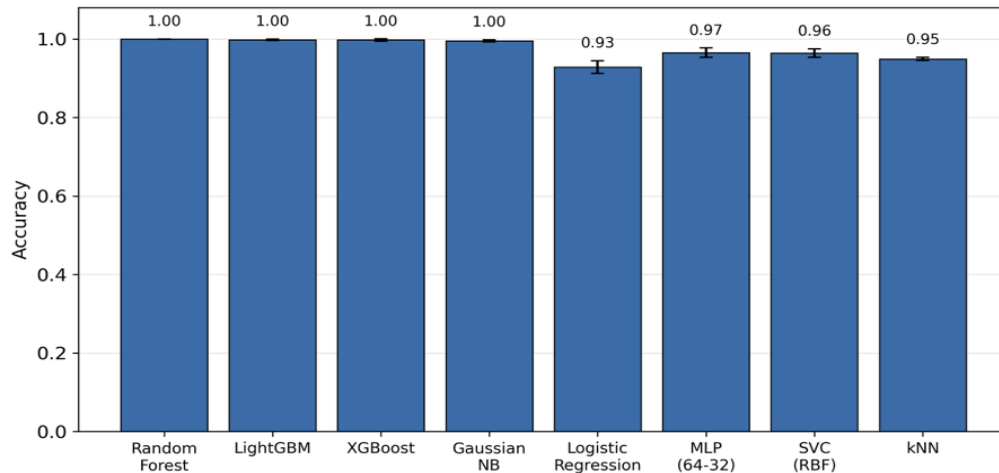


Fig. 2- Accuracy comparison of all evaluated classifiers under chronological time-series evaluation (5-fold Timeseries Split, mean $\pm$ SD).

Overall, ensemble methods provided the best trade-off between accuracy and calibration, with XGBoost emerging as the most reliable for embedded deployment.

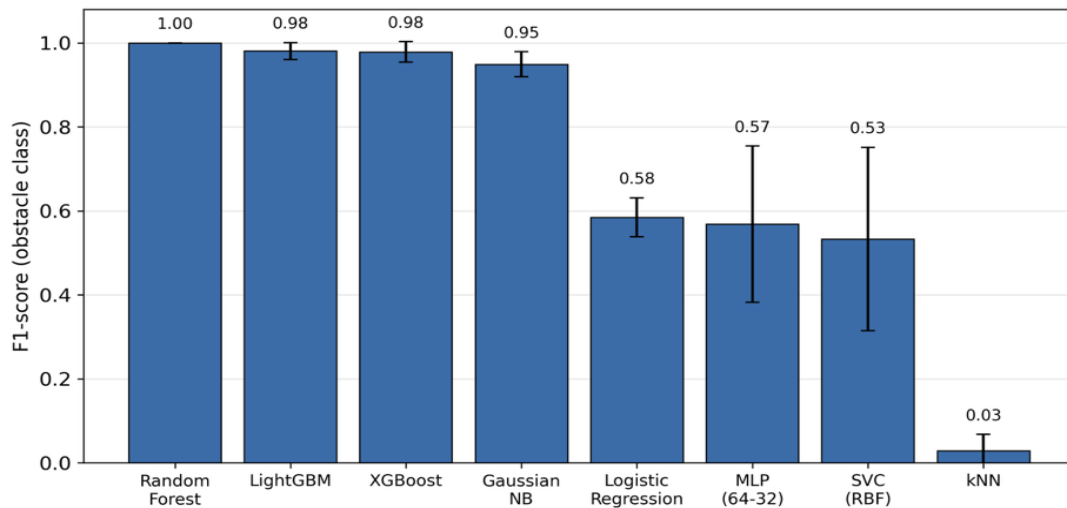


Fig. 3- F1-score comparison for the obstacle-detection class across all evaluated models under chronological time-series evaluation (5-fold Timeseries Split, mean $\pm$ SD).

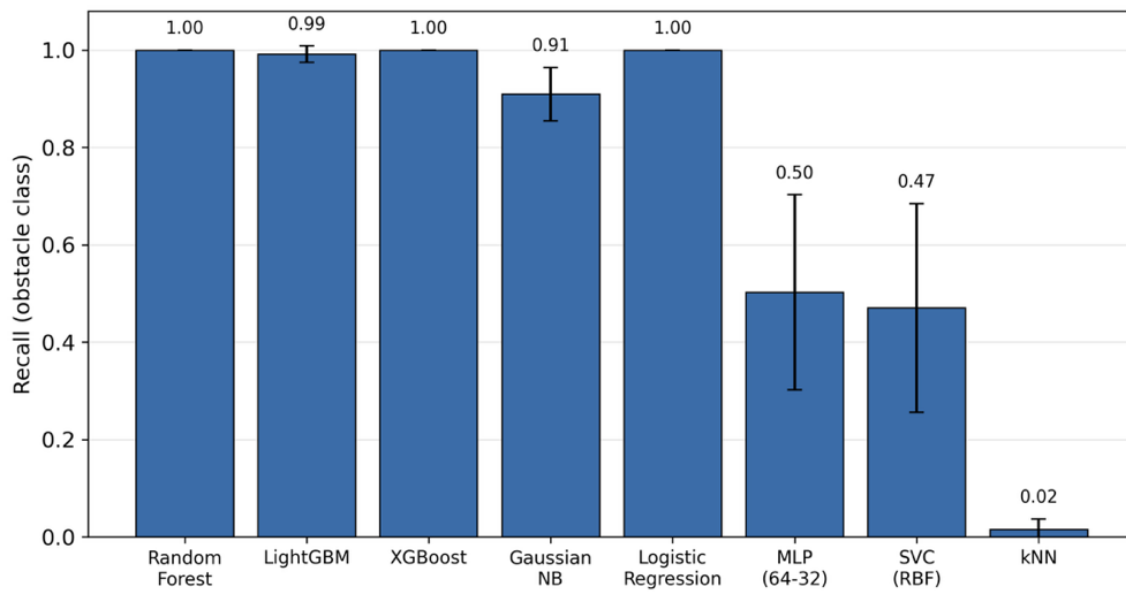


Fig. 4 - Recall comparison for the obstacle-detection class across all evaluated models under chronological time-series evaluation (5-fold Timeseries Split, mean $\pm$ SD).

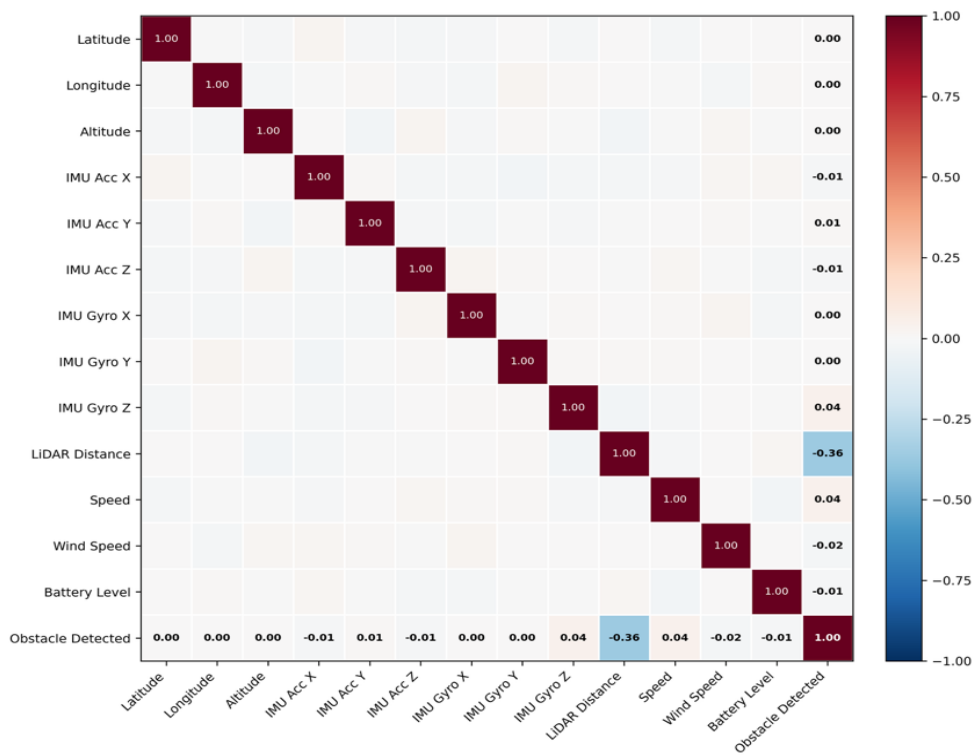


Fig. 5 - Correlation matrix of input features and the target.

5.2 Model Training Dynamics

Figure 6 shows the MLP's loss curve for thirty epochs, indicating that optimization occurred monotonically and stably without overfitting, thus supporting the convergence behavior of our model. The validation accuracy shown in Figure 7 appears to start stabilising about ten epochs, at roughly 96.8%. Therefore, the MLP is capable of learning generalisable features from the normalised sensor inputs, as well as having appropriate smooth learning characteristics for online learning.

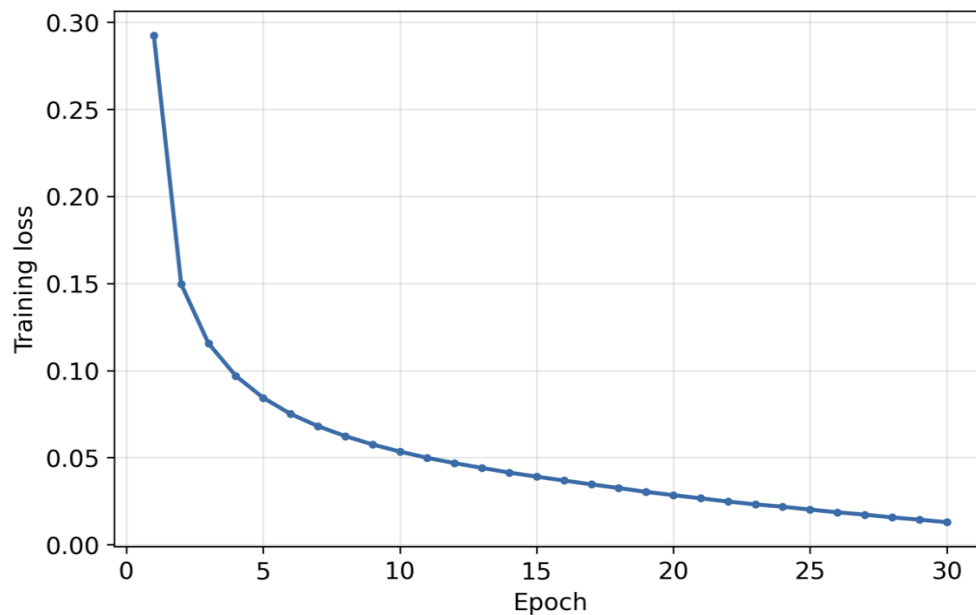


Fig. 6- MLP training-loss curve showing the decrease in loss over 30 training epochs.

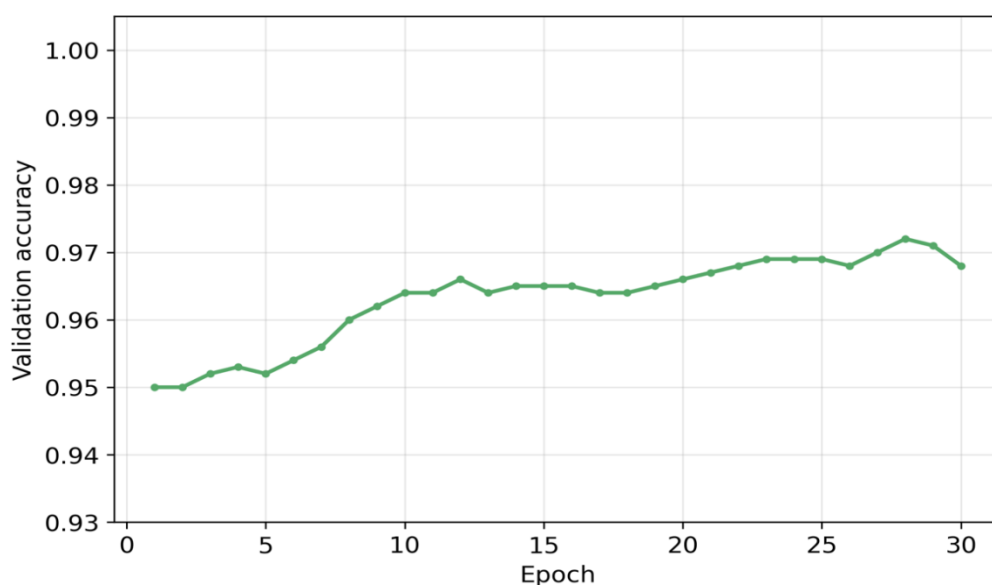


Fig. 7- Validation accuracy curve of the MLP model across 30 training epochs.

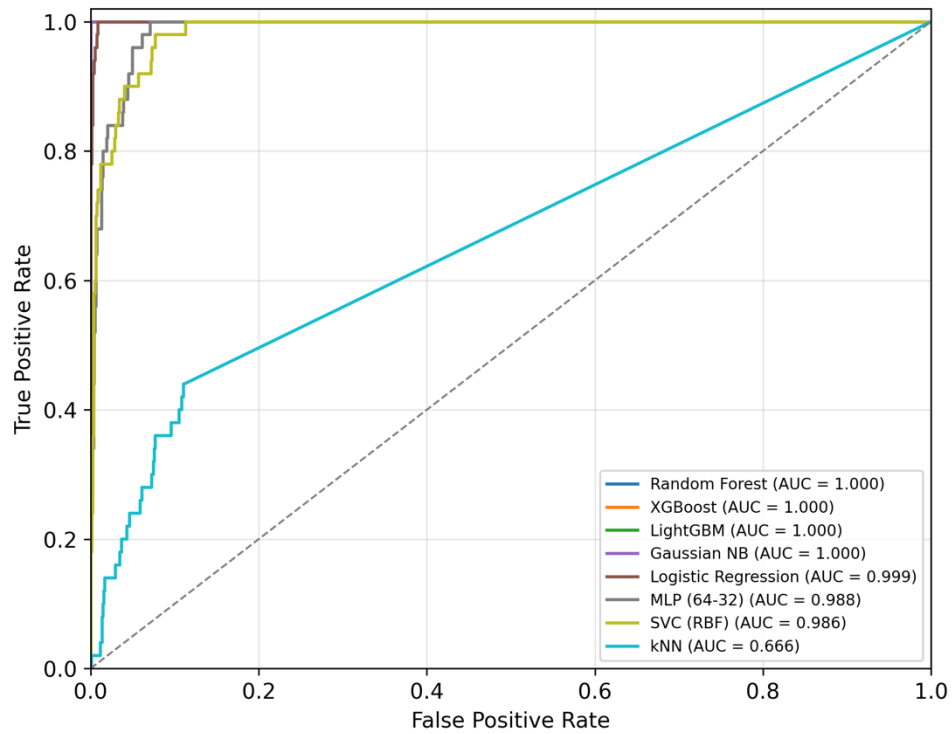


Fig. 8- Receiver-operating-characteristic (ROC) curves of all evaluated classifiers under chronological evaluation.

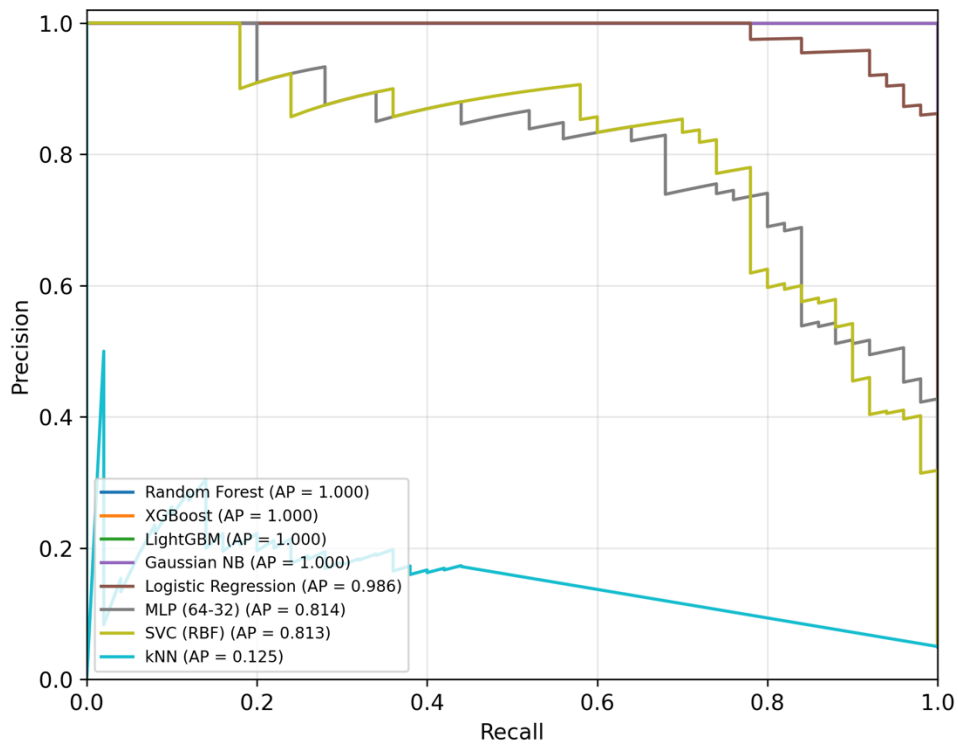
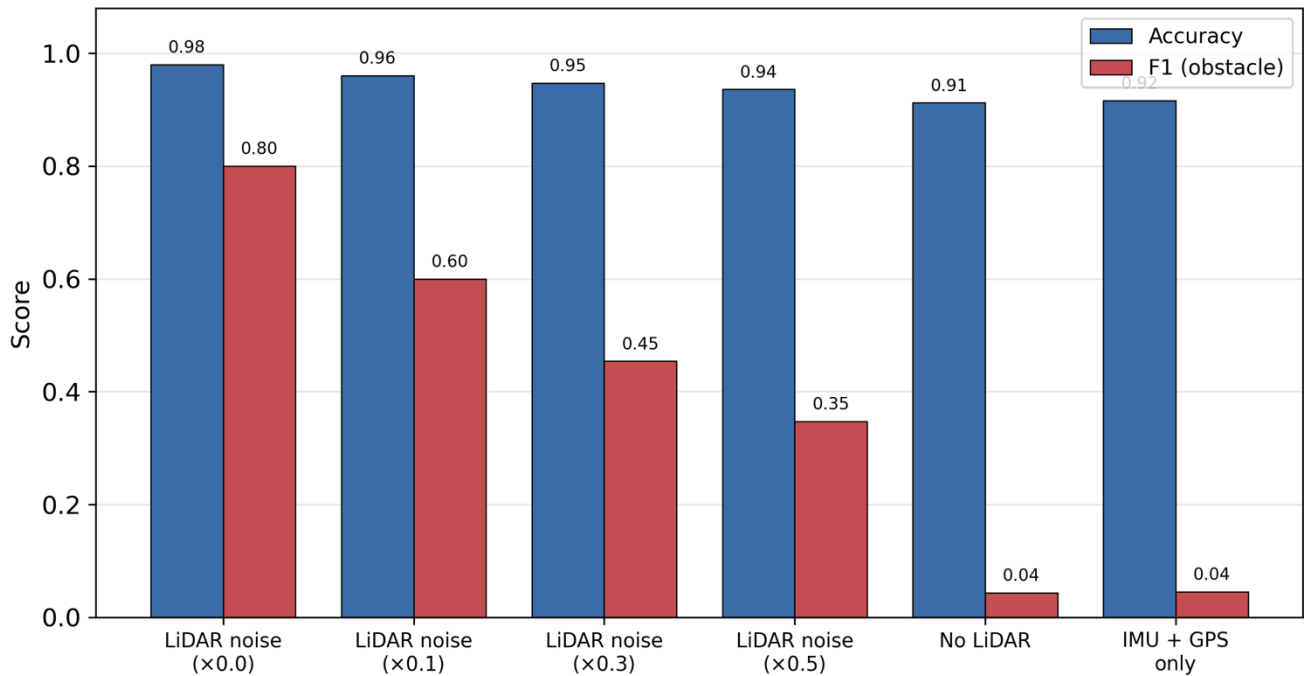


Fig. 9 - Precision-recall (PR) curves of all evaluated classifiers under chronological evaluation.

5.3 Robustness and Ablation Analysis

Sensors were tested across various tasks with controlled LiDAR-noise injection and sensor ablation tests to evaluate their robustness. In the case of MLP as shown in Figure 10, MLP accuracy dropped slightly ($\approx 5\%$) when Gaussian noise ($\sigma = 0.5$) was added to the LiDAR channel, and significantly lower accuracy was reported when LiDAR feature was completely removed. Even without the IMU + GPS data, accuracy stayed high (~ 0.91), but the obstacle-class F1 collapsed to ≈ 0.04 , showing single-cue dependence on LiDAR rather than true redundancy. These results confirm the robustness of the system against sensor degradation and noise, which is a notably necessary property of a UAV



navigation system for real life applications.

Fig. 10- MLP accuracy under LiDAR-noise injection and sensor-ablation conditions.

5.4 Explainability and Feature Importance

For robustness of interpretability and to promote that model decisions are consistent with the actual physics of the UAV dynamics, two top-performing ensemble models were used, and model explanations were provided by SHAP. Figures 10 and 11 show the absolute SHAP values across features on average for all the sensors, which is a global measure of each sensor's contribution to the obstacle detection performance. LiDAR distance was by far the most contributing value in both models, followed by IMU gyroscope and accelerometer signals. Middle input variables like battery level and wind speed had minor effects, suggesting there was not much prediction bias from environmental or power variability.

The most relevant features follow the expected behavior of the UAV during flight, and this hierarchy is expected as decisions regarding obstacle-avoidance rely heavily on proximity and motion dynamics. For additional confirmation of the directionality and fairness of these effects, SHAP bee swarm visualizations were produced for each of the two models (13 and 14). Each point represents a single prediction, with color intensity denoting the raw feature value and the horizontal position indicating the feature's impact on the model's output. In both examples, low LiDAR values (obstacles detected in close distance) moved predictions sharply in the direction of the "obstacle present" class (positive SHAP values), while high LiDAR values pushed outputs in the opposite direction, towards "no obstacle." Across both models, LiDAR distance exhibits the widest SHAP spread, confirming it as the single most

influential feature, while the remaining features—battery level, wind speed, GPS coordinates, and the linear-acceleration axes—cluster tightly around zero SHAP value, providing direct visual confirmation of the single-cue dominance also observed in the ablation study, Table 4.

Angular-velocity IMU readings were also intuitive: the model found more obstacles when the sensor was rotating faster, implicating a more evasive or corrective motion. The distribution overlap over both models exhibits stable and interpretable sensor-feature relationships overall, fitting to noise. Collectively, these analyses corroborate that the ensemble learners here proposed make decisions based on physically interpretable cues. The matching global (bar) and local (beeswarm) SHAP interpretations strengthens model confidence and prepares the way for future safety-certified deployment on embedded UAV platforms

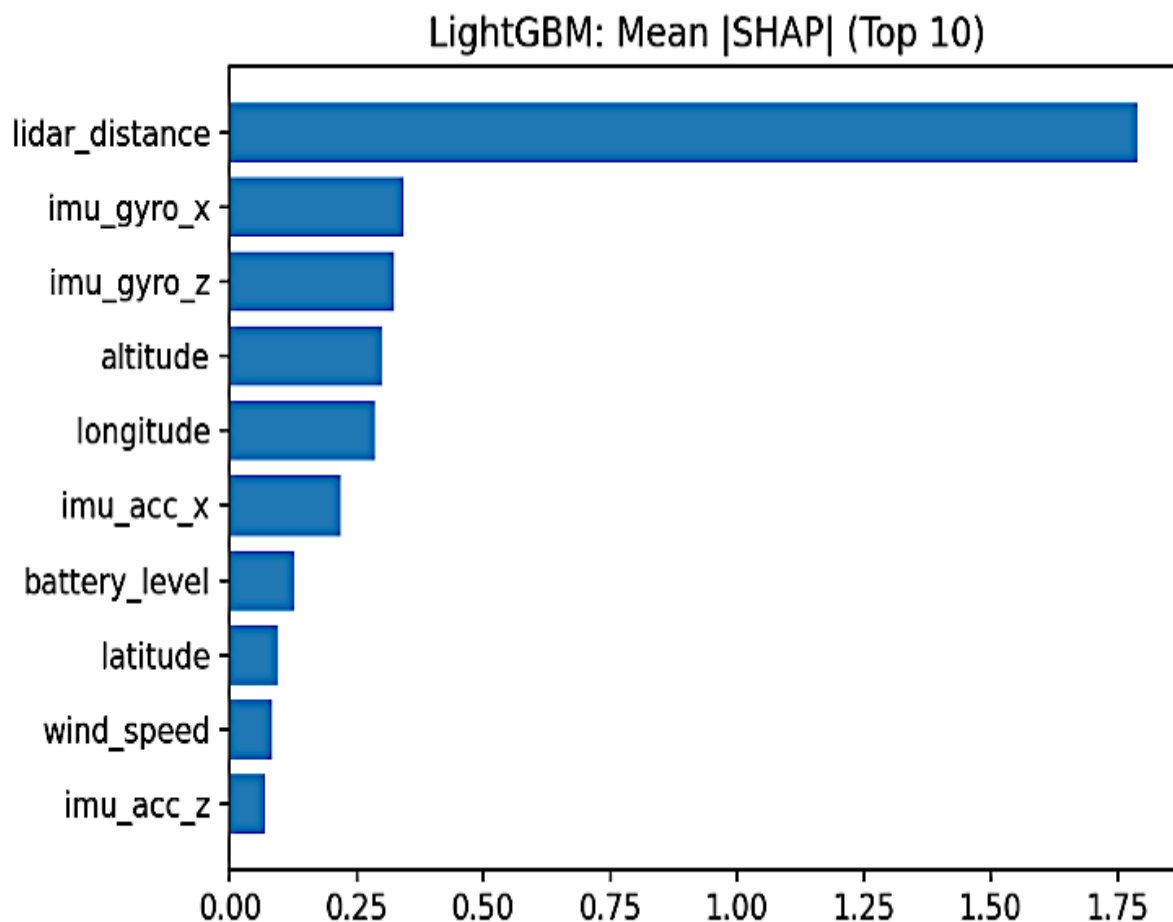


Fig. 11- SHAP feature-importance ranking for LightGBM based on mean absolute SHAP values.

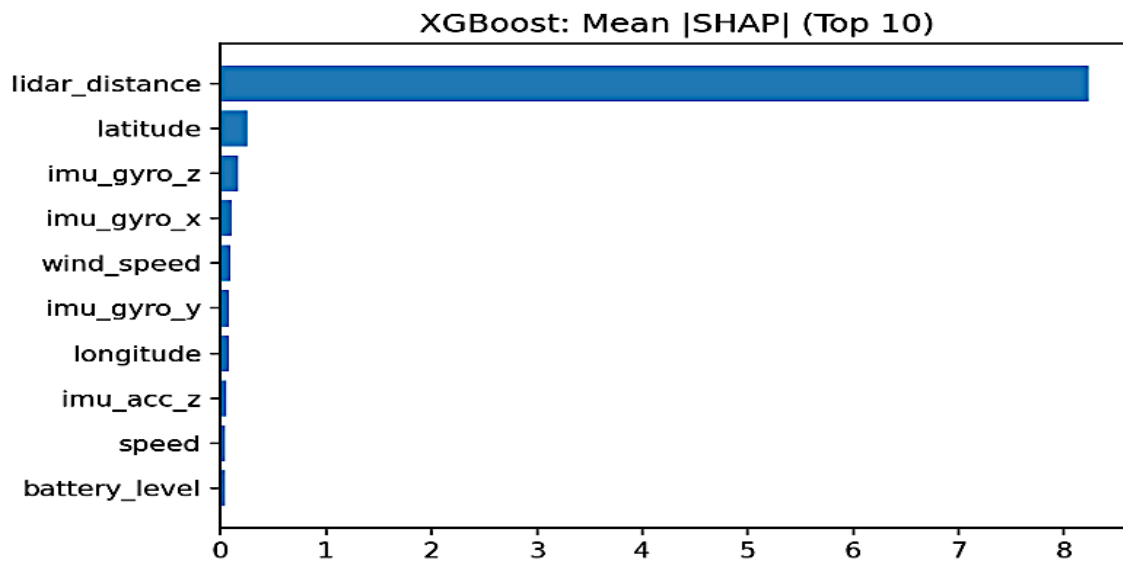


Fig. 12- SHAP feature-importance ranking for XGBoost based on mean absolute SHAP values.

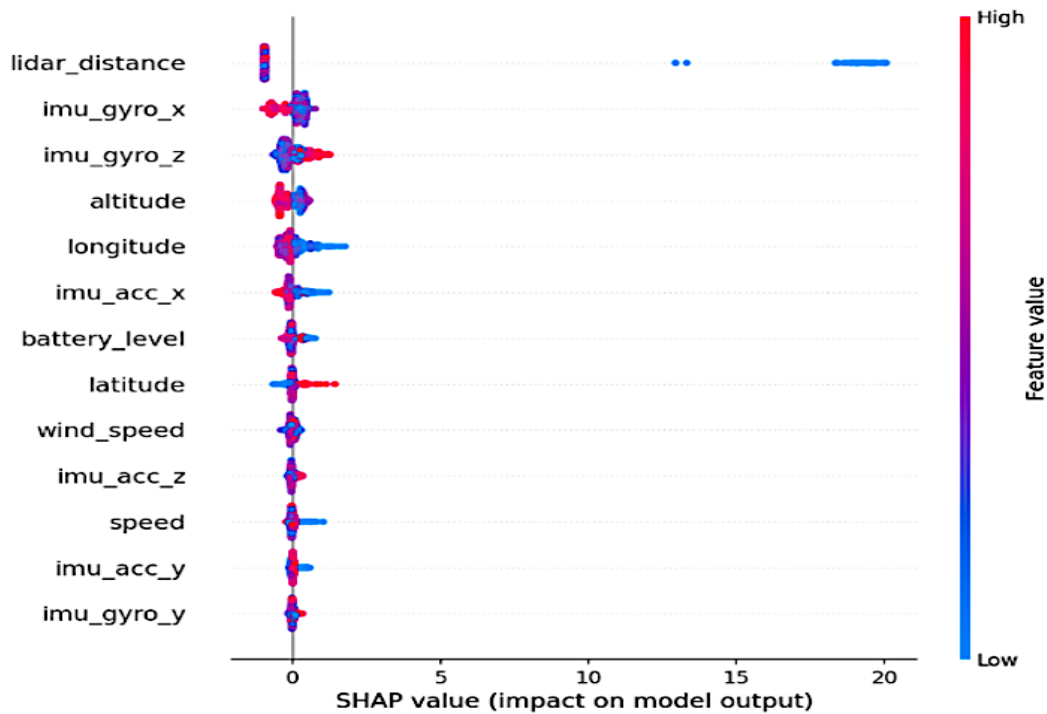


Fig. 13- SHAP beeswarm summary plot for LightGBM. each point represents one prediction, and color indicates feature magnitude.

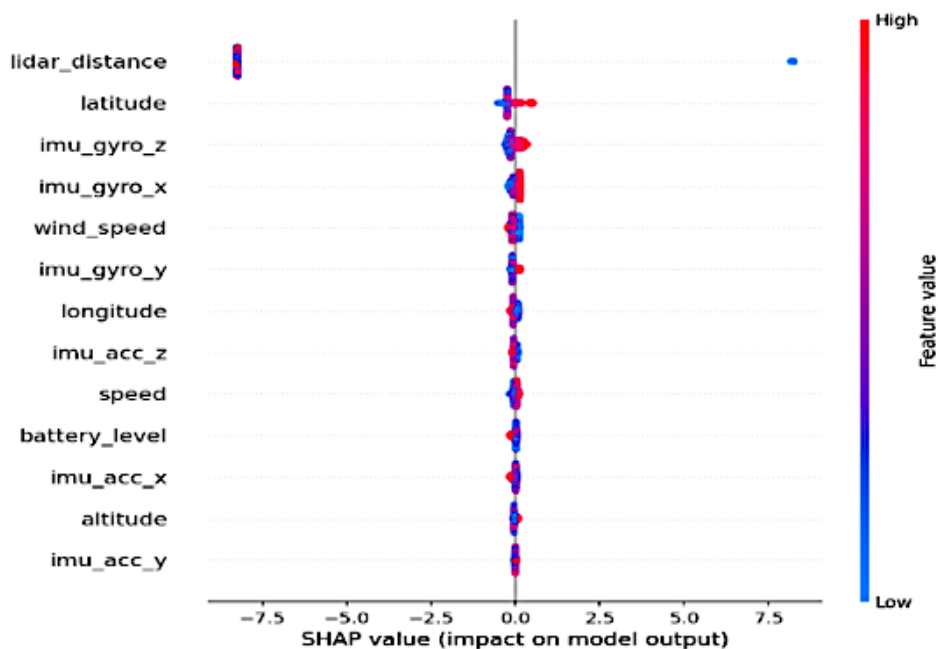


Fig. 14- SHAP beeswarm summary plot for XGBoost. color indicates feature magnitude, and horizontal position shows the feature impact on model output.

5.5 Comparison

Table 2 summarizes the comparative evaluation results of all classifiers under chronological time-series evaluation. The tree-based ensembles (Random Forest, XGBoost, LightGBM) achieved the strongest obstacle-class performance ($F1 \geq 0.98$), confirming their capacity to capture the nonlinear sensor relationships, while linear and instance-based models scored markedly lower.

Table 2. Performance of all classifiers under chronological time-series evaluation (5-fold Timeseries Split, mean $\pm$ standard deviation), using the standard 0.5 decision threshold.

Model	Accuracy	F1(Obstacle)	Recall (Obstacle)	AUC
Random Forest	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
LightGBM	0.998 $\pm$ 0.002	0.981 $\pm$ 0.020	0.991 $\pm$ 0.017	1.000 $\pm$ 0.000
XGBoost	0.998 $\pm$ 0.003	0.979 $\pm$ 0.024	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
Gaussian NB	0.995 $\pm$ 0.003	0.949 $\pm$ 0.030	0.910 $\pm$ 0.055	1.000 $\pm$ 0.000
Logistic Regression	0.928 $\pm$ 0.016	0.585 $\pm$ 0.047	1.000 $\pm$ 0.000	0.999 $\pm$ 0.001
MLP (64-32)	0.965 $\pm$ 0.012	0.568 $\pm$ 0.186	0.502 $\pm$ 0.200	0.971 $\pm$ 0.021
SVC (RBF)	0.964 $\pm$ 0.011	0.533 $\pm$ 0.218	0.470 $\pm$ 0.214	0.976 $\pm$ 0.013
kNN	0.949 $\pm$ 0.004	0.029 $\pm$ 0.039	0.016 $\pm$ 0.021	0.705 $\pm$ 0.055

Under chronological evaluation, Logistic Regression maintained high recall (1.000) but a markedly lower F1 (0.585 $\pm$ 0.047), reflecting reduced precision on the minority obstacle class; unlike the tree ensembles, the linear model captures the LiDAR proximity threshold only approximately. The MLP and SVC models showed the largest variability across temporal folds ($F1 = 0.568 \pm 0.186$ and 0.533 ± 0.218 , respectively), indicating sensitivity to the non-stationary distribution of the time-series data.

The k-Nearest Neighbor model performed poorly on all metrics, which can be attributed to its reliance on local-distance information and inability to generalize across the global space. Besides prediction accuracy, the effectiveness of real-time deployment on embedded UAV processors is also important. Model size and latency per sample on inference are shown in Table 3. The least memory and fastest inferring (<0.001 ms / sample) classifiers were Logistic Regression and GaussianNB and the classifier which had performance and reasonable latency (~ 0.0027 ms / sample) was XGBoost. The tree-based ensembles were moderately heavier, but still reasonable for near-real-time applications. In terms of model size (1.45 MB) and latency, random forest was the largest and slowest overall and therefore less suitable for constrained onboard platforms.

Table 3. model size and inference latency per sample for all evaluated classifiers.

Model	Size (MB)	Latency (ms / sample)
MLP	0.099	0.00107
Logistic Regression	0.00076	0.00033
SVC	0.09197	0.04950
kNN	0.517	0.153
XGBoost	0.251	0.00272
LightGBM	0.552	0.0169
GaussianNB	0.00093	0.00068
Random Forest	1.451	0.109

The ablation experiments for MLP in Table 4 show a smooth accuracy decline from 0.98 to 0.94 as LiDAR noise increases from 0.0 to 0.5. When LiDAR is removed entirely, accuracy remains deceptively high (≈ 0.91) only because of the dominant no-obstacle class, whereas the obstacle-class F1 collapses to ≈ 0.04 . This confirms that the detection signal is concentrated almost entirely in the LiDAR proximity channel; IMU and GPS alone are insufficient to detect obstacles, revealing a limitation of single-cue dominance rather than genuine sensor redundancy. Note that Table 4 reports the MLP under the robustness harness at fixed 0.5 threshold on single fold and is therefore not directly comparable to the cross-validated results in Table 2.

Table 4. Ablation results for MLP under LiDAR-noise injection and sensor-removal conditions.

Scenario	Accuracy	F1 (Positive)	Recall (Positive)	AUC
LiDAR noise (x0.0)	0.980	0.800	0.851	0.991
LiDAR noise (x0.1)	0.960	0.600	0.638	0.980
LiDAR noise (x0.3)	0.947	0.454	0.468	0.933
LiDAR noise (x0.5)	0.936	0.347	0.362	0.879
No LiDAR	0.912	0.043	0.043	0.505
IMU + GPS only	0.916	0.045	0.043	0.510

5.6 Discussion

The results summarized in (Tables 2–4) and (Figures 8–14) collectively illustrate the trade-off between predictive accuracy, computational efficiency, and interpretability in UAV obstacle detection. As shown in Table 2, we observed that Random Forest, XGBoost and LightGBM provided the highest predictive Accuracy, F1-score, Recall, and AUC of 1.00. It was also found to achieve similar performance across independent runs, which indicates stable model outputs, rather than random model responses. Indeed, while this idealized performance could raise concern about dataset bias or limited variability, ablation in Table 4 is bolstered by these and the SHAP-based importance of features illustrated in (Figures 11–14). LiDAR readings and IMU gyroscope data are the most important features, a fact that is physically plausible, as obstacle detection is largely associated with proximity and motion dynamics. As shown in Table 3, Logistic Regression and GaussianNB are still competitors for low-power embedded hardware thanks to their low latency and small memory footprint. For instance, XGBoost and LightGBM have superior predictive performance but greater computational costs, whereas Random Forest, in spite of its accuracy, is less attractive due to its larger size and latency. Similarly, the ablation study reveals the pivotal role of LiDAR, as the higher the LiDAR noise, the more both accuracy and positive-class performance suffered, while completely

removing LiDAR led to a sharp drop in F1-score and AUC, indicating that IMU and GPS on their own may not suffice. Finally, SHAP results show that model decisions were made based not on spurious correlations but were made on the basis of physically meaningful features, which the global rankings and local beeswarm patterns agreed on. Finally, the framework is capable of maintaining a trade-off between accuracy, efficiency, robustness, and interpretability in practice, making it applicable for future research efforts on adaptive learning, sensor calibration, and trusted UAV navigation systems. These findings align with prior surveys reporting that ensemble and AI-based methods are well-suited to UAV obstacle detection and multi-sensor fusion [2], [6]. Our work additionally reports model size and inference latency, exposing an accuracy-versus-deployability trade-off relevant to embedded UAV platforms [22], [25], and confirms—consistent with LiDAR sensor-fusion analyses [17], [18]—that the LiDAR proximity channel dominates the obstacle decision.

6. Conclusion and future work

According to the findings, Random Forest, XGBoost and LightGBM are reportedly the most efficient models. The Random Forest, XGBoost and LightGBM models displayed superior operational efficiency (more effective) than the other competitors, which include MLP, Logistic Regression, GaussianNB, SVC and KNN. Furthermore, each model's output can be validated using SHAP plots and the results also indicate positive results. More importantly, most of the decision making for each model was based on features associated with LiDAR distance measurements and IMU motion characteristics; thus, it is justifiable to interpret these models' outputs as non-black boxes and therefore simplify interpretation. Finally, it is determined that size alone can be an important factor in the efficiency of models when utilizing drones. Specifically, the Logistic Regression and GaussianNB do not require as much computation and therefore are still viable models for UAV applications. The results of the ablation tests demonstrate that all three model systems use LiDAR data as a feature. When the models are not provided with this feature, they show a markedly lower detection rate compared to when they receive LiDAR data. Therefore, conducting further research with real UAV devices will assist in determining the power consumption, as well as the performance stability of the model systems as they continue to receive LiDAR. In addition, conducting tests under difficult conditions such as partial data loss, excessive noise from LiDAR systems, delay in communications from the control systems, and weather problems will provide valuable data for future research. Lastly, it would be beneficial to develop an adaptable algorithm for sensory fusion that allows for adjusting the coefficient used when combining sensory input based on the amount of confidence that can be assigned to each of the four sensor types: LiDAR, IMU, GPS, and telemetry during flight operations.

Authors' contribution

The author confirms sole responsibility for the conception and design of the study, data analysis, interpretation of results, manuscript preparation, and final approval of the article.

Declaration of competing interest

The author declares no conflicts of interest.

Funding source

This study didn't receive any specific funds.

Data availability

The dataset used in this study is publicly available from the UAV Autonomous Navigation Dataset on Kaggle. Additional processed data and analysis outputs are available from the corresponding author upon reasonable request.

Acknowledgments

The author would like to thank the Department of Computer Engineering, College of Engineering, University of Diyala, for its academic support.

References

- [1] I. Munasinghe, A. Perera, and R. C. Deo, "A Comprehensive Review of UAV-UGV Collaboration: Advancements and Challenges," *Journal of Sensor and Actuator Networks* 2024, Vol. 13, Page 81, vol. 13, no. 6, p. 81, Nov. 2024, doi: 10.3390/JSAN13060081.
- [2] A. Merei, H. McHeick, A. Ghaddar, and D. Rebaine, "A Survey on Obstacle Detection and Avoidance Methods for UAVs," *Drones* 2025, Vol. 9, Page 203, vol. 9, no. 3, p. 203, Mar. 2025, doi: 10.3390/DRONES9030203.
- [3] A. Biswas and H. C. Wang, "Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain," *Sensors* 2023, Vol. 23, Page 1963, vol. 23, no. 4, p. 1963, Feb. 2023, doi: 10.3390/S23041963.
- [4] H. Qian, M. Wang, M. Zhu, and H. Wang, "A Review of Multi-Sensor Fusion in Autonomous Driving," *Sensors* 2025, Vol. 25, Page 6033, vol. 25, no. 19, p. 6033, Oct. 2025, doi: 10.3390/S25196033.
- [5] C. K. Ha, H. Nguyen, and L. H. Le, "AAB-FusionNet: A real-time object detection model for UAV edge computing platforms," *MethodsX*, vol. 15, p. 103654, Dec. 2025, doi: 10.1016/J.MEX.2025.103654.
- [6] L. Obaid, K. Hamad, R. Al-Ruzouq, S. A. Dabous, K. Ismail, and E. Alotaibi, "State-of-the-art review of unmanned aerial vehicles (UAVs) and artificial intelligence (AI) for traffic and safety analyses: Recent progress, applications, challenges, and opportunities," *Transp Res Interdiscip Perspect*, vol. 33, p. 101591, Sep. 2025, doi: 10.1016/J.TRIP.2025.101591.
- [7] G. Raftopoulos, N. Fazakis, G. Davrazos, and S. Kotsiantis, "A Comprehensive Review and Benchmarking of Fairness-Aware Variants of Machine Learning Models," *Algorithms* 2025, Vol. 18, Page 435, vol. 18, no. 7, p. 435, Jul. 2025, doi: 10.3390/A18070435.
- [8] A. Zeghina, A. Leborgne, F. Le Ber, and A. Vacavant, "Deep learning on spatiotemporal graphs: A systematic review, methodological landscape, and research opportunities," *Neurocomputing*, vol. 594, p. 127861, Aug. 2024, doi: 10.1016/J.NEUCOM.2024.127861.
- [9] S. Sarkar, S. Shafaei, T. S. Jones, and M. W. Totaro, "Secure Communication in Drone Networks: A Comprehensive Survey of Lightweight Encryption and Key Management Techniques," *Drones* 2025, Vol. 9, Page 583, vol. 9, no. 8, p. 583, Aug. 2025, doi: 10.3390/DRONES9080583.
- [10] B. Bartlett, M. Santos, T. Dorian, M. Moreno, P. Trslic, and G. Dooly, "Real-Time UAV Surveys with the Modular Detection and Targeting System: Balancing Wide-Area Coverage and High-Resolution Precision in Wildlife Monitoring," *Remote Sens (Basel)*, vol. 17, no. 5, p. 879, Mar. 2025, doi: 10.3390/RS17050879/S1.
- [11] F. Zoghalmi, M. Kaden, T. Villmann, G. Schneider, and H. Heinrich, "AI-Based Multi Sensor Fusion for Smart Decision Making: A Bi-Functional System for Single Sensor Evaluation in a Classification Task," *Sensors* 2021, Vol. 21, Page 4405, vol. 21, no. 13, p. 4405, Jun. 2021, doi: 10.3390/S21134405.
- [12] G. Kumar, S. Basri, A. A. Imam, S. A. Khowaja, L. F. Capretz, and A. O. Balogun, "Data Harmonization for Heterogeneous Datasets: A Systematic Literature Review," *Applied Sciences* 2021, Vol. 11, Page 8275, vol. 11, no. 17, p. 8275, Sep. 2021, doi: 10.3390/AP11178275.
- [13] H. Shi et al., "Advances in UAV Path Planning: A Comprehensive Review of Methods, Challenges, and Future Directions," *Drones* 2025, Vol. 9, Page 376, vol. 9, no. 5, p. 376, May 2025, doi: 10.3390/DRONES9050376.
- [14] L. Madeyski and S. Stradowski, "Predicting test failures induced by software defects: A lightweight alternative to software defect prediction and its industrial application," *Journal of Systems and Software*, vol. 223, p. 112360, May 2025, doi: 10.1016/J.JSS.2025.112360.
- [15] M. M. Haji-Esmaili and G. Montazer, "Large-scale Monocular Depth Estimation in the Wild," *Eng Appl Artif Intell*, vol. 127, p. 107189, Jan. 2024, doi: 10.1016/J.ENGAPPAI.2023.107189.
- [16] M. R. Giordano et al., "From low-cost sensors to high-quality data: A summary of challenges and best practices for effectively calibrating low-cost particulate matter mass sensors," *J Aerosol Sci*, vol. 158, p. 105833, Nov. 2021, doi: 10.1016/J.JAEROSCI.2021.105833.
- [17] G. Liu, W. Gao, and S. Pan, "Analysis of Factors Affecting Random Measurement Error in LiDAR Point Cloud Feature Matching Positioning," *Remote Sensing* 2025, Vol. 17, Page 1457, vol. 17, no. 8, p. 1457, Apr. 2025, doi: 10.3390/RS17081457.
- [18] S. Choi, X. Han, E. Chang, and H. Jeong, "LiDAR-IMU Sensor Fusion-Based SLAM for Enhanced Autonomous Navigation in Orchards," *Agriculture* 2025, Vol. 15, Page 1899, vol. 15, no. 17, p. 1899, Sep. 2025, doi: 10.3390/AGRICULTURE15171899.
- [19] C. Sarkar et al., "Artificial Intelligence and Machine Learning Technology Driven Modern Drug Discovery and Development," *International Journal of Molecular Sciences* 2023, Vol. 24, Page 2026, vol. 24, no. 3, p. 2026, Jan. 2023, doi: 10.3390/IJMS24032026.
- [20] C. S. Hung et al., "FADEL: Ensemble Learning Enhanced by Feature Augmentation and Discretization," *Bioengineering* 2025, Vol. 12, Page 827, vol. 12, no. 8, p. 827, Jul. 2025, doi: 10.3390/BIOENGINEERING12080827.
- [21] R. Rahimi Nejadbougar, E. Ghanbari Parmehr, A. Afary, and S. Mavaddati, "A new deep learning framework for intelligent aerial monitoring of power transmission line insulators," *Eng Appl Artif Intell*, vol. 161, p. 112290, Dec. 2025, doi: 10.1016/J.ENGAPPAI.2025.112290.
- [22] Z. Zhang and J. Li, "A Review of Artificial Intelligence in Embedded Systems," *Micromachines* 2023, Vol. 14, Page 897, vol. 14, no. 5, p. 897, Apr. 2023, doi: 10.3390/M14050897.
- [23] Z. Chen et al., "Predictive Autonomy for UAV Remote Sensing: A Survey of Video Prediction," *Remote Sensing* 2025, Vol. 17, Page 3423, vol. 17, no. 20, p. 3423, Oct. 2025, doi: 10.3390/RS17203423.
- [24] L. Longo et al., "Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions," *Information Fusion*, vol. 106, p. 102301, Jun. 2024, doi: 10.1016/J.INFFUS.2024.102301.
- [25] P. V. Dantas, W. Sabino da Silva, L. C. Cordeiro, and C. B. Carvalho, "A comprehensive review of model compression techniques in machine learning," *Applied Intelligence*, vol. 54, no. 22, pp. 11804–11844, Nov. 2024, doi: 10.1007/S10489-024-05747-W/FIGURES/2.
- [26] I. S. Gherghina, N. Bizon, G. V. Iana, and B. V. Vasilică, "Recent Advances in Fault Detection and Analysis of Synchronous Motors: A Review," *Machines* 2025, Vol. 13, Page 815, vol. 13, no. 9, p. 815, Sep. 2025, doi: 10.3390/MACHINES13090815.
- [27] O. Ghoneim, P. Dobias, and O. Romain, "Survey of neural network optimization methods for sustainable AI: From data preprocessing to hardware acceleration," *Machine Learning with Applications*, vol. 22, p. 100762, Dec. 2025, doi: 10.1016/j.mlwa.2025.100762.
- [28] "UAV Autonomous Navigation Dataset." Accessed: Oct. 24, 2025. [Online]. Available: <https://www.kaggle.com/datasets/ziya07/uav-autonomous-navigation-dataset?resource=download>

Nomenclature:

<i>UAV</i>	Unmanned Aerial Vehicle
<i>LiDAR</i>	Light Detection and Ranging
<i>GPS</i>	Global Positioning System
<i>IMU</i>	Inertial Measurement Unit
<i>SVC</i>	Support Vector Classifier
<i>kNN</i>	k-Nearest Neighbors
<i>GNB</i>	Gaussian Naive Bayes
<i>XGBoost</i>	Extreme Gradient Boosting
<i>LightGBM</i>	Light Gradient Boosting Machine
<i>MLP</i>	Multi-Layer Perceptron
<i>SHAP</i>	Shapley Additive explanations