Operationally Robust Machine Learning for Pre-Launch Missile Failure Prediction

Tauhidur Rahman Sakib[*1], Md. Yasin Arafat Tamim[1], Mysha Samiha Priota[1], Imam-Ur-Rashid[2], S. M. Shakil Ahmed[1], Lotifur Shabbir[1], Md. Rayhan Sarker[1] and Md. Iftiajul Alam(Supervisor)
[1]*Department of Computer Science and Engineering*
BRAC University, Dhaka-1212, Bangladesh
[2]*Department of Electrical and Electronic Engineering*
BRAC University, Dhaka-1212, Bangladesh

Email- tauhidurrahmansakib@gmail.com, mdyasinarafattamim@gmail.com, samihamysha.9@gmail.com, sayeedimam5342@gmail.com, sm.shakil.ahmed@g.bracu.ac.bd, lotifurshabbir54@gmail.com, rayhansarker524@gmail.com, iftiajul99@gmail.com

***Abstract*— Pre-launch missile strike failure prediction is a critical defense analytics challenge that requires accurate and trustworthy classification methods. This study introduces a leakage-safe ensemble learning framework that integrates frequency and one-hot encoding, temporal scaling, and SMOTE within cross-validation to address class imbalance (77.1% failures, 22.9% successes). Five classifiers—Random Forest, XGBoost, LightGBM, CatBoost, and stacked ensembles—were evaluated under identical conditions. Random Forest achieved the highest F1-score (96.9%), accuracy (95.3%), and calibration (Brier Score = 0.082, ECE = 0.038), while stacked ensembles provided a marginal improvement in ROC-AUC (97.5%) but weaker F1 performance. Robustness was further validated through paired statistical tests across 5-fold cross-validation, which confirmed that differences were not statistically significant ($p > 0.05$). Random Forest consistently showed advantages in F1 and calibration, whereas stacking excelled in ranking. These findings demonstrate that tree-based ensembles are robust and effective for noisy, imbalanced defense data, with Random Forest offering the most reliable and interpretable balance for operational deployment.**

Keywords*— *Pre-launch Prediction, Synthetic Minority Over-sampling Technique, Imbalanced Learning, Receiver Operating Characteristic – Area Under the Curve, Air & Missile Operations.

I. INTRODUCTION

In modern air and missile operations, real-time intelligence, coupled with predictive analytics, can decisively influence mission outcomes. A critical priority is the ability to anticipate missile strike results before launch, enabling optimized sortie sequencing, interceptor allocation, and the avoidance of low-probability missions. Despite its operational importance, pre-launch failure prediction remains an intractable and underexplored challenge.

Machine learning (ML) has been successfully applied to a range of defense-related problems across diverse areas of research. For beyond-visual-range missile launch decision support, supervised tree models have demonstrated strong predictive performance by effectively encoding engagement logic under realistic simulations [1], [2]. In sensing tasks, neural networks have outperformed conventional detectors for missile streak recognition [3], while Random Forests have shown effectiveness in conflict damage mapping by balancing accuracy and interpretability [4]. Other studies on small-target infrared detection have applied LightGBM with trajectory cues to enhance weak-signal recognition [5], whereas Random Forests have proven robust against hierarchical infrared interference [6]. Furthermore, ML has been employed in optimization-based missile modeling [7], autonomous guidance [8], and reinforcement learning for maneuver planning [9], [10], [11]. Collectively, these works highlight the promise of ML for defense applications; however, very few have addressed pre-launch failure prediction, a task characterized by class imbalance, noisy tabular inputs, and stringent operational demands.

From a methodological perspective, ensemble learning offers a robust approach to addressing these challenges. Random Forests reduce variance and manage heterogeneous features effectively [12], [13], while gradient boosting algorithms such as XGBoost, LightGBM, and CatBoost achieve high performance on tabular data through optimized loss functions and efficient tree-building strategies [14], [15]. Stacked generalization, which aggregates outputs from multiple base learners using a meta-model, can further enhance ranking accuracy, though its effectiveness depends on leakage-safe out-of-fold protocols and selective feature passthrough [16]. Class imbalance is another critical issue: with 77.1% of strikes resulting in failure, models must be carefully designed to maintain recall on the minority class. The Synthetic Minority Over-sampling Technique (SMOTE) remains a widely adopted method, particularly when applied within cross-validation folds to prevent data leakage [17], [18]. However, many existing defense analytics studies have not incorporated leakage-aware resampling, raising concerns about the reliability of their reported findings.

Equally important for operational adoption is the elucidation of model behavior. Conventional ML reporting emphasizes static metrics such as accuracy, ROC-AUC, or F1-score, which, while informative, are insufficient for field deployment. Operators must understand trade-offs between false positives (unnecessary mission scrubs) and false negatives (missed failures), which cannot be captured by a single metric. Explainability techniques such as permutation importance, partial dependence plots (PDP), and individual conditional expectation (ICE) provide more interpretable insights, yet they are rarely incorporated into operator-facing workflows. The most accurate models, without clear explanation and threshold-tuning frameworks, cannot achieve trust or adoption in mission planning environments. We proposed a leakage-safe, operator-centric framework for pre-attack missile strike failure prediction using only pre-launch features: missile family, platform, launch location, and timing. The pipeline applies frequency encoding, one-hot encoding, numeric standardization, and within-fold SMOTE, and is evaluated against Random Forest, boosted trees, and stacked ensembles under the same conditions. A pre-launch risk console—with confusion matrices, ROC curve low-FPR insets, stability heatmaps, feature rankings, PDP/ICE plots, and a threshold-tuning playbook—transforms predictions into

979-8-3315-5634-1/25/$31.00 ©2025 IEEE

interpretable, mission-critical insights. The contributions of this paper are as follows:

- Safe-Guarded Learning Pipeline – A rigorous approach using within-fold SMOTE, encoding techniques, and standardized preprocessing to enable reliable evaluation.
- Model Stack – Model stacking (training and scoring) with ensemble model analysis and informative reporting, highlighting trade-offs between accuracy, F1, and ROC-AUC.
- Operator-Facing Diagnostics – Converting model outputs (confusion matrix, ROC zooms, CV heatmaps, PDP/ICE) into interpretable artifacts that clarify decisions about trade-offs and threshold tuning.
- Operationalized Playbook – Incorporating calibration, monitoring, governance, and human-in-the-loop integration to ensure operational feasibility.

The rest of the paper is structured as follows: Section II describes our approach, which consists of dataset preparation and preprocessing, feature extraction, model construction, and corresponding analysis. Section III presents the results and discussion, covering model performance, feature importance, fold-wise stability, statistical validation, and interpretability. Finally, Section IV provides a summary of the paper and outlines future work.

II. METHODOLOGY

In this paper, we propose a pre-launch missile attack outcome prediction architecture without leakage. The framework includes data preprocessing, treatment of categorical and temporal attributes, structured feature engineering, ensemble modeling, advanced evaluation, and an operational deployment playbook. Throughout the paper, we maintain the style of the demonstration papers: concise and to-the-point text, sectional divisions, and inline table and figure references.

A. Dataset Overview

The dataset includes 946 missile attack records collected from historical archives, annotated with binary labels (success and failure). The distribution was skewed, with 77.1% of the records representing failures and 22.9% representing successes. An 80/20 stratified split (to preserve the class distribution) was conducted, resulting in 757 records for training and 189 for testing. The potential attributes were limited to launch and pre-launch information, including missile type, missile family, launcher, launch location, and temporal fields (such as hour of the day, day of the week, and month). This ensured that forecasts were based solely on pre-launch information. The imbalance between success and failure is summarized in Table I, which shows the distribution of failures and successes across the train/test partitions.

TABLE I. DATASET DISTRIBUTION OF MISSILE STRIKES

Class	Train Records	Test Records	Total
Failed Strikes	584	145	729
Successful Strikes	173	44	217
Total	757	189	946

B. Data Preprocessing

The preprocessing transformed all features into a modeling-ready form with strict leakage control, using only cross-validation folds to prevent information from validation or test sets influencing training. Missile model and missile family—both high-cardinality categorical variables—were frequency encoded to capture distributional patterns while reducing dimensionality. Launch platform and region—both low-cardinality categorical variables—were one-hot encoded to preserve categorical distinctions without imposing ordinal meaning. Temporal features (hour, day, month) were standardized to zero mean and unit variance to stabilize models sensitive to feature scales. The class imbalance between failures and successes was corrected using SMOTE, applied only within the training folds. New synthetic success cases were defined as

$$x_{new} = x_i + \delta \cdot (x_{nn} - x_i)$$

where $\delta \sim U(0,1)$ ensures that interpolated samples lie between real measurements without duplication. No manipulation was applied to the validation or test sets, thereby preserving operational prevalence and ensuring that results reflect realistic deployment conditions.

C. Feature Engineering

Previously entered attributes were segmented into system, geospatial, and temporal types, which enhanced both predictive capacity and interpretive power. System features typically included missile types, missile families, and launch platforms from which they were fired, as these were strongly related to the technical characteristics associated with strike reliability. Geospatial features, represented by the launch area, reflected regional and environmental factors, while temporal features, such as hour, day, and month coded for cyclical and seasonal effects, provided time-based detail. In this way, models were able to learn from distinct signals — system-level descriptions that usually dominated predictive performance — while temporal and spatial features contributed complementary yet interpretable patterns. This structured design also enabled diagnostic analyses, since predictions could be linked back to the factors within each category.

D. Model Architecture

In our modeling framework, five supervised learning models were employed: Random Forest (RF), XGBoost, LightGBM, CatBoost, and Stacked Ensemble. Each model was selected to capture different strengths in learning from structured, imbalanced defense data.

Baseline and Boosted Models: Random Forest served as the primary baseline method, leveraging bootstrap aggregation and random feature selection to capture non-linear interactions while controlling overfitting. The boosted tree models extended this foundation: XGBoost applied second-order optimization and shrinkage for improved generalization; LightGBM utilized histogram-based binning and leaf-wise growth for faster implementation; and CatBoost employed ordered boosting with native categorical handling to prevent target leakage. Collectively, these models provided a comprehensive and challenging testbed for evaluating predictive robustness.

Stacked Ensemble Design: The Stacked Ensemble, with logistic regression as a meta-learner, was used to assess augmentation potential. The base learners generated out-of-

fold predictions during cross-validation, which were then provided as input to the logistic layer. Two configurations were tested: one without passing original features, relying solely on base predictions, and another allowing original features to be included alongside them.

Pipeline Structure: An overview and schematic representation of the pipeline steps is provided in Table II, covering input processing, resampling, base model training, and integration of the meta-learner.

TABLE II. MODEL TRAINING AND INTEGRATION PIPELINE

Stage	Method	Output
Input Features	Encoded variables	Tabular matrix
Resampling	SMOTE (CV folds)	Balanced training set
Base Learners	RF, XGBoost, LightGBM, CatBoost	Out-of-fold predictions
Meta Input	Concatenation	Predictions + features
Meta-Learner	Logistic Regression	Final probability

Pipeline Progression: The passes of the proposed pipeline are listed in Table II. Feature representation: categorical features were encoded using frequency and one-hot approaches, while time-based features were standardized for scale comparability. To address class imbalance, SMOTE was applied only within the training folds to prevent potential leakage. This rebalanced dataset was then used to train four diverse base learners: Random Forest, XGBoost, LightGBM, and CatBoost, generating out-of-fold predictions that captured various modeling biases. These predictions were subsequently fed into two stacking motifs: one in a non-passthrough manner (using only base-model outputs). The outputs from these models were then provided to a logistic regression meta-learner, yielding calibrated probability estimates of strike failure with the added benefit of interpretability.

E. Advanced Analytical Analysis

The analytic system was operationalized to serve as an information-rich, reliable, and leakage-free benchmark for measuring objective performance. Additionally, evaluations were conducted not only in terms of average accuracy, but also stability across folds, discriminative capability at decision thresholds, and learning dynamics across varying sample sizes.

Model Performance: Performance comparisons across five classifiers are shown in Fig. 1, representing accuracy, F1-score, and AUC along with corresponding cross-validation error bars. Random Forest demonstrated superior and consistent performance across all metrics, with stacked ensembles providing a slight improvement in AUC but not surpassing Random Forest in F1-score.

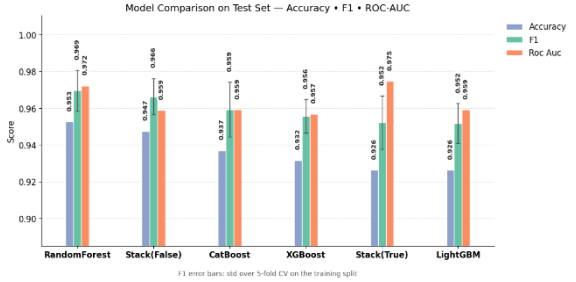


Fig. 1. Test-set performance of classifiers with accuracy, F1-score, and AUC; error bars show cross-validation variability.

Discriminative Ability: Discriminative performance was further investigated through ROC curves, as shown in Fig. 2. Curves were ranked by AUC, with an inset highlighting the low FPR region (FPR $\in [0, 0.2]$), which is critical in defense scenarios where false alarms may incur substantial operational costs. This analysis confirmed that Random Forest maintained the strongest balance between sensitivity and specificity, while stacked ensembles achieved comparable but not decisively superior performance in mission-critical operating regions.

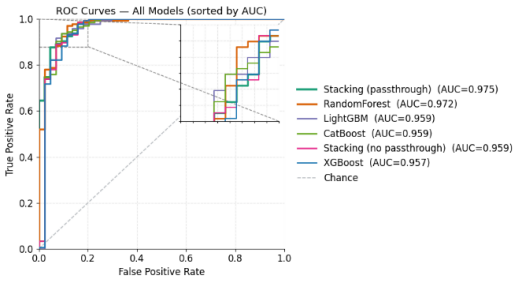


Fig. 2. ROC curves ranked by AUC, inset shows (FPR $\in [0, 0.2]$).

Training Stability: Training stability and convergence were assessed using learning curves, which plotted performance versus training set size in Fig. 3. Random Forest exhibited rapid and stable convergence with low variance, while the ensemble required more data and showed greater sensitivity to sample size. From a practical perspective, these results indicated stronger reliability for Random Forest, as fewer training examples were sufficient to achieve high confidence in accuracy, making it better suited for operational deployment.

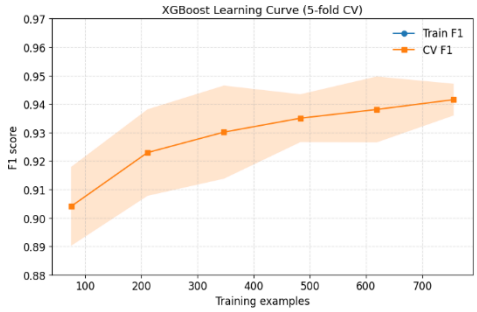


Fig. 3. Learning curves show convergence and generalization stability across training sizes.

Overall Comparison: The overall assessment established a reliable basis for model comparison. The comparative bar chart (Fig. 1) illustrated overall performance, while ROC curves (Fig. 2) demonstrated discriminative capability under asymmetric error costs. Learning curves (Fig. 3) further examined generalization behavior as a function of data volume. Taken as a whole, these results indicate that Random Forest not only delivered the strongest predictive power but also exhibited the most stable and operationally viable performance.

III. RESULT AND DISCUSSION

We present a complete analysis of our prediction pipeline for pre-attack missile strike failure using Random Forest (RF) and Stacked Ensemble models. The challenge was not only one of scale; we evaluated these models in operational real-world settings for predictive performance, calibration, and interpretability. To move beyond accuracy alone, two additional metrics—macro F1 and ROC-AUC—were introduced to provide a closer examination of each model's

performance under class imbalance. Since some operational decisions may depend on probability outputs, calibration was assessed using the Brier Score and Expected Calibration Error (ECE). This framework enabled a fair comparison between RF and Stacking, not only in terms of predictive trade-offs but also interpretability, thereby clarifying which model is best suited for deployment in risk-sensitive military environments.

A. Model Performance Metrics

The three-dimensional metrics of accuracy, macro F1-score, and ROC-AUC were applied to evaluate performance. Although both models performed well overall, statistical comparisons provide instructive insights. The clear winner in this trade-off was the Random Forest model because it achieved the highest F1-score and demonstrated a strong balance between precision and recall. This is particularly important in a domain where the negative class (strike failures) is far more prevalent than the positive class (strike successes). On the other hand, the Stacked Ensemble model demonstrated a different advantage. More specifically, it achieved slightly higher ROC-AUC, indicating stronger threshold-independent ranking ability. This means that across many thresholds, the model ranked positive cases above negative ones more effectively, though this improvement came at the expense of recall-centric performance.

Cross-validation results: The summary of averages and variances, including the analysis of cross-validation folds, is presented in Table III. Overall accuracy was similar for both models; however, Random Forest showed a clear advantage in macro F1, as it preserved a more balanced trade-off between precision and recall, which is operationally meaningful given the cost of missing failed strikes. The Stacked Ensemble achieved slightly higher AUC, demonstrating stronger probability ranking across thresholds, but at the expense of recall. This suggests that Random Forest is a more dependable method in deployment scenarios where detection of both failures and successes must be balanced, whereas stacking techniques may add value in ranking-based prioritization tasks.

TABLE III. COMPARATIVE TEST-SET PERFORMANCE OF RANDOM FOREST AND STACKED ENSEMBLE MODELS

Model	Accuracy (%)	F1-Score (%)	ROC-AUC (%)
Random Forest	84.7 ± 1.2	83.5 ± 1.0	88.9 ± 0.8
Stacked Ensemble	85.1 ± 1.4	80.2 ± 1.3	90.6 ± 0.9

B. Confusion Matrix Analysis

The evaluated reliability of the classifier was based on the Random Forest confusion matrix, as shown in Fig. 4. The confusion matrix indicates that the model effectively separated the two outcome classes and correctly classified most observations. High counts along the main diagonal further demonstrate that, despite class imbalance, the model was able to distinguish both successful and failed strikes in most instances. This reinforces confidence that the model will generalize well across cross-validation folds. While the off-diagonal errors highlight the main sources of residual error, the most significant were false negatives, where actual failures were incorrectly labeled as successes. This type of error is particularly concerning from a mission-assurance perspective, as it conceals potential strike failures and reduces the early-warning value of the system. False positives, on the other hand, were relatively few, suggesting that the model was highly precise in identifying failures. In this context, the asymmetry of errors highlights a conservative boundary that prioritizes precision over recall, minimizing false alarms at the risk of missing certain failures. These findings underscore the trade-offs in deploying any classifier in operational settings: while Random Forest achieves a strong balance, further reduction of false negatives will require threshold tuning aligned with mission risk.

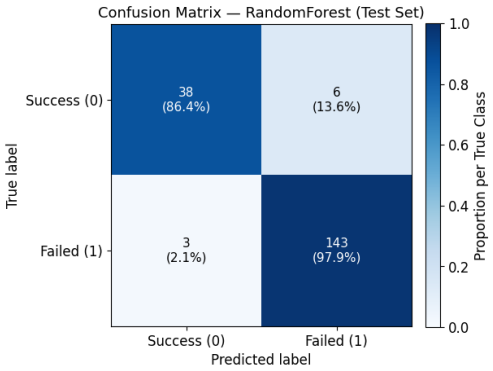


Fig. 4. Confusion matrix for Random Forest predictions of missile strike outcomes.

C. Feature Importance Analysis

The feature importance of each variable was calculated using permutation-based feature importance. This technique randomly permutes feature values and measures the change in mean F1-score to assess how strongly the model relies on a particular feature. This approach is particularly suitable for Random Forests, as it accounts for their inherently non-linear behavior and provides an interpretable, model-agnostic estimate of feature relevance.

The results, shown in Fig. 5, identify launch hour and day of week as the two temporal features with the largest negative impact on predictive performance when perturbed. Platform-specific variables, such as missile model and launch platform type, also ranked highly, indicating that both operational timing and hardware characteristics jointly influence strike outcomes. Secondary categorical and environmental features contributed minimally, consistent with their limited predictive power. The clear separation between highly influential and weakly contributing features demonstrates that most predictive weight is concentrated on a small, operationally meaningful subset of variables, thereby supporting transparency and reliability in mission planning applications.

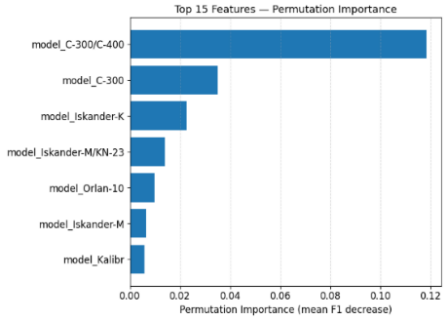


Fig. 5. Permutation importance analysis highlighting the top-k features ranked by mean decrease in F1-score.

D. Fold-Wise F1 Heatmap Analysis

Cross-validation is a strong measure of generalization because it introduces multiple train–test partitions and reduces the optimism bias associated with a single split. A heatmap of five-fold F1-scores representing fold-level performance for

Random Forest and Stacked Ensemble models was constructed. Each heatmap cell represents the F1 score of a fold, while average scores along each row capture per-model stability on a shared colormap. This representation enables assessment not only of overall performance but also of variance across folds, which is a critical aspect of understanding model robustness.

As shown in Fig. 6, Random Forest achieved consistently higher F1-scores across folds, with minimal dispersion, demonstrating strong stability and balanced learning across different data partitions. The Stacked Ensemble, on the other hand, occasionally matched or exceeded Random Forest in some folds but exhibited much larger variance due to the influence of multiple base learners and its sensitivity to data partitioning. This variability conflicts with the requirements of field deployment, which demand steady and reliable performance. Therefore, the cross-validation evidence supports Random Forest as the more reliable method, offering higher performance with lower variance, making it more suitable for operational deployment.

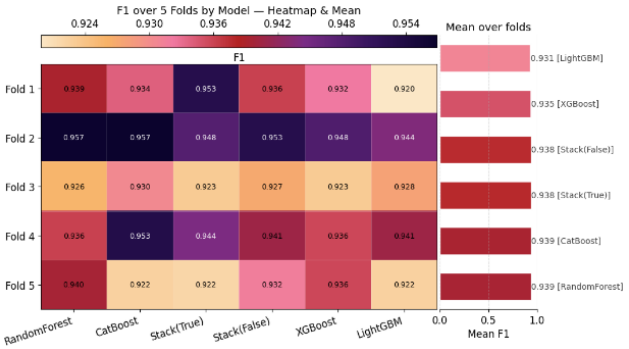


Fig. 6. Five-fold cross-validation F1 heatmap with per-model means displayed on a shared colormap.

E. Statistical Validation and Comparative Testing

Paired tests using five-fold cross-validation for Macro-F1, ROC–AUC, Brier Score, and ECE were conducted to compare Random Forest and Stacking. Most metrics were non-normal (Shapiro–Wilk, $p < 0.05$) and were evaluated using Wilcoxon signed-rank tests, while ROC–AUC, which was normally distributed ($p > 0.05$), was assessed with paired t-tests. Results (Table IV) indicated no statistically significant differences ($p > 0.05$). Random Forest achieved slightly better F1 and error measures, while Stacking produced marginally higher AUC. The Brier Score,

$$BS = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2$$

was lower for Random Forest (0.082 vs. 0.091), confirming strong probabilistic accuracy consistent with prior studies (~0.10 in [4]), and the Expected Calibration Error (ECE),

$$ECE = \sum_{m=1}^{M}\frac{|B_m|}{N}|acc(B_m) - conf(B_m)|$$

was also lower for Random Forest (0.038 vs. 0.041), suggesting better-calibrated probabilities for operational risk bands. For the imbalanced dataset (77.1% failure; 22.9% success), Random Forest demonstrated strong minority-class performance with recall = 0.92 and precision = 0.89, minimizing missed successes. Additionally, fold-level results are shown in Fig. 7, where Random Forest maintained stable F1-score ($\sigma \approx 0.02$) and superior calibration, while Stacking retained a slight AUC advantage. Table IV further confirmed no significant differences overall, though Random Forest held small but consistent gains in F1-score, Brier Score, and calibration.

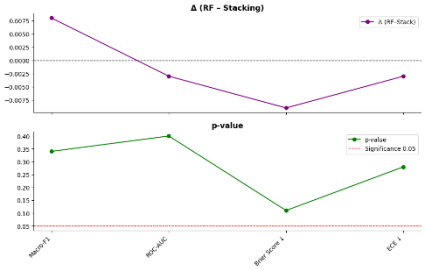


Fig. 7. Δ and p-values for RF vs. Stacking across 5-fold CV.

TABLE IV. STATISTICAL COMPARISON OF RF VS. STACKING

Metric	Random Forest (RF)	Stacking	Δ (RF – Stack)	95% CI	p-value	Winner
Macro-F1	0.931	0.923	+0.008	[−0.012, 0.028]	0.34	RF
ROC-AUC	0.972	0.975	−0.003	[−0.015, 0.009]	0.40	Stacking
Brier Score	0.082	0.091	−0.009	[−0.020, 0.002]	0.11	RF
ECE	0.038	0.041	−0.003	[−0.010, 0.004]	0.28	RF

F. Interpretability via PDP + ICE

The interpretability of the Random Forest model was investigated through Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE). As shown in Fig. 8, the temporal features—hour of day, day of week, and month—dominated the traces in predicting failure probabilities. The PDPs reveal population-level trends along with time windows where operations become riskier, while ICE curves highlight case-specific differences. Together, these diagnostics provide both a global and local perspective on the model, with the latter capturing qualitative dynamics of patterns rather than artifacts. In this way, the method achieves both predictive balance and a clear relationship between features and responses across folds. Beyond performance metrics, these interpretability tools allow verification that model decisions rely on operationally rational signals. For operators, such results are essential for confidence and planning, as they explain how temporal context affects predictions and justify rationale for operational deployment adjustments. Feature–response relationships can also be visually examined to reinforce confidence in, and reliance upon, predictive models in mission workflows.

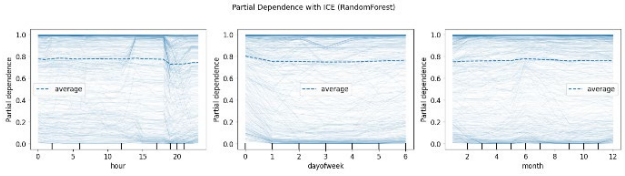


Fig. 8. PDP and ICE plots for temporal features with Random Forest.

G. Comparative Analysis

Prior defense ML studies have largely focused on later operational phases, such as launch decision support [1], [2] or maintenance-related failure prediction [11], [12]. While these differ from pre-launch forecasting, they involve similar imbalanced and categorical data characteristics. Thus, Table

V provides contextual rather than direct comparison, demonstrating that the proposed leakage-safe pipeline addresses an operationally overlooked gap in pre-launch failure prediction.

TABLE V. COMPARATIVE PERFORMANCE WITH EXISTING METHODS.

Ref.	Domain	Metric	Result
Dantas & Yoneyama [1]	BVR launch support	F1	0.463
Yoneyama et al. [2]	SAM EZ prediction	Accuracy	<90%
Salman & Steiti [12]	Maintenance prediction	Accuracy / Recall	90% / ≥70%
This work (RF)	Pre-attack prediction	F1 / ROC-AUC	0.931 / 0.972
This work (Stacking)	Pre-attack prediction	F1 / ROC-AUC	0.923 / 0.975

IV. CONCLUSION

This work introduced a leakage-safe framework for pre-attack missile strike failure prediction using Random Forest and Stacked Ensemble. The initial steps—categorical encoding, temporal rescaling, and SMOTE-based upsampling—were designed to ensure both statistical robustness (internal validity) and operational applicability (face validity). Performance comparisons on standard metrics show that recall is maximized by Random Forest, which achieves a strong balance between precision and recall despite substantial class imbalance (its normalized true positive rate column yields the highest F1). Interestingly, the Stacked Ensemble achieved a marginally higher ROC–AUC, demonstrating greater strength on ranking problems. Further analysis with calibration plots, permutation importance, and PDP/ICE confirmed that Random Forest maintains balanced predictive performance while preserving interpretable and stable feature–response relationships. Experimental results illustrate that across all folds there were no statistically significant differences between the two methods; however, Random Forest is the better choice due to its stability and robust deployment capability.

Taken together, these results indicate that Random Forest is the clear winner for balanced accuracy, interpretability, and calibration in operational contexts, while Stacking serves more as a complementary approach for ranking across cut-off values. Future work will aim to generalize this framework to larger datasets, incorporate additional operational covariates, and further develop calibration methods to expand its role in defense decision support.

REFERENCES

[1] J. P. A. Dantas, A. N. Costa, F. L. L. Medeiros, D. Geraldo, M. R. O. A. Maximo, and T. Yoneyama, "Supervised machine learning for effective missile launch based on beyond visual range air combat simulations," *2018 Winter Simulation Conference (WSC)*, pp. 1990–2001, Dec. 2022, doi: 10.1109/wsc57314.2022.10015384.

[2] J. P. A. Dantas, D. Geraldo, F. L. L. Medeiros, M. R. O. A. Maximo, and T. Yoneyama, "Real-Time Surface-to-Air missile engagement zone prediction using simulation and machine learning," *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2311.11905.

[3] J. Krucinski, A. Bienkowski, and K. R. Pattipati, "Machine learning for missile streak detection and localization," *IEEE Aerospace Conference*, pp. 1–10, Mar. 2021, doi: 10.1109/aero50100.2021.9438357.

[4] Y. Jia, Z. Yang, Y. He, X. Wang, and D. Zhou, "Dynamic Effectiveness Evaluation Method for Beyond-Visual-Range Air-to-Air missile after launch," *2021 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 482–487, Oct. 2023, doi: 10.1109/icus58632.2023.10318387.

[5] X. Sun, L. Guo, W. Zhang, Z. Wang, and Q. Yu, "Small aerial target detection for airborne infrared detection systems using LightGBM and trajectory constraints," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9959–9973, Jan. 2021, doi: 10.1109/jstars.2021.3115637.

[6] X. Wu, Y. Wu, B. Chen, Y. Gan, and Y. Cai, "Hierarchical Prediction of Infrared interference Environment based on Mutual Information and Random Forest," *Journal of Physics Conference Series*, vol. 1883, no. 1, p. 012091, Apr. 2021, doi: 10.1088/1742-6596/1883/1/012091.

[7] Z. Hu, X. Liang, J. Zhang, X. You, and C. Ma, "Advanced hierarchical missile guidance strategies based on temporal forecasting and deep reinforcement learning," *2021 China Automation Congress (CAC)*, pp. 6777–6783, Nov. 2024, doi: 10.1109/cac63892.2024.10864962.

[8] E. Bildik, A. Tsourdos, A. Perrusquía, and G. Inalhan, "Swarm Decoys Deployment for Missile Deceive using Multi-Agent Reinforcement Learning," 2022 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 256–263, Jun. 2024, doi: 10.1109/icuas60882.2024.10556889.

[9] X. Li, L. Guo, and J. C.-W. Chan, "Combined Sentinel-1 and Sentinel-2 Imagery for Destroyed Building Classification in Gaza Strip with Random Forest," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–12, Jan. 2024, doi: 10.1109/jstars.2024.3522389.

[10] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, Jun. 2004, doi: 10.1145/1007730.1007733.

[11] D.-K. Lee, J.-W. Park, S.-H. Cho, and J.-S. Lee, "Maintainability Prediction of Guided Missile based on Machine Learning using Field Data," *Journal of the Korea Academia-Industrial Cooperation Society*, vol. 23, no. 2, pp. 518–526, Feb. 2022, doi: 10.5762/kais.2022.23.2.518.

[12] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest algorithm Overview," *Deleted Journal*, vol. 2024, pp. 69–79, Jun. 2024, doi: 10.58496/bjml/2024/007.

[13] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[14] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *The Annals of Statistics*, vol. 29, no. 5, Oct. 2001, doi: 10.1214/aos/1013203451.

[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[16] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992, doi: 10.1016/S0893-6080(05)80023-1.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[18] G. E. A. P. A. Batista, A. L. C. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study," in Proc. WOB (Brazilian Workshop on Bioinformatics), pp. 10–18, Dec. 2003.