ML-Driven Jamming Detection for Critical Communication Systems with SMOTE-Based Optimization

Bhuvaneswari Yennapusala, Sohith Bukka
Department of Computer Engineering and Artificial Intelligence, Marwadi University, Rajkot, Gujarat, India
bhuvaneswariyennapusala.116514@marwadiuniversity.ac.in, bukkasohith.118516@marwadiuniversity.ac.in

***Abstract*—In the age of interconnected systems, wireless signal jamming has evolved as a significant threat to critical communication networks, including those used in emergency services, national security, and transportation. Jamming attacks are often carried out using inexpensive and easily accessible equipment to interfere with communication systems, including military communications and GPS at key airports. These attacks, combined with the challenge of hidden jammers, make detection and enforcement difficult. This issue has collected attention at various international security forums, such as the Munich Security Conference, the UN Counter-Terrorism Week, and CyCon. Despite ongoing advancements, jamming detection remains a significant challenge in wireless security. In this research, we propose a machine learning-based approach for jamming detection, enhanced by SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance. We evaluate several classification models, including K-Nearest Neighbors, Random Forest, Support Vector Machine, Logistic Regression, and XGBoost. Experimental results show that SMOTE significantly enhances model performance, with XGBoost outperforming all other models in terms of precision-recall curves, log loss score, accuracy, and cross-validation stability, followed by Random Forest. These results confirm that our approach provides an effective solution for jamming detection in critical communication environments.**

Index Terms—Jamming Detection, Machine Learning, Critical Communication Systems, SMOTE, Data Imbalance, Wireless Security, Classification Models

I. Introduction

In an era of seamlessly integrated technologies, wireless signal jamming has emerged as a significant threat to communication networks across critical sectors such as emergency services, national security, and transportation [1] [2] [3]. These attacks are often done using cheap and easily available tools, making it easy for attackers to disturb important systems, including emergency response and surveillance [2]. High-profile incidents have involved military communication blackouts during conflicts and GPS interference at major airports [4], [5]. The covert nature of jamming devices further complicates enforcement and detection. This concern has gained international attention at forums such as the Munich Security Conference, which emphasized hybrid warfare and electronic threats [6]; the UN Counter-Terrorism Week, addressing the exploitation of emerging technologies by terrorist groups [7]; and the International Conference on Cyber Conflict (CyCon), which highlighted advancements in cyber defence strategies [8]. Despite ongoing advancements, jamming detection remains a complex and unresolved challenge in the field of wireless network security.

Traditional methods for detecting jamming usually rely on spotting unusual patterns in data and analyzing the physical signals being transmitted. Arjoune et al. [9] introduced a machine learning-based framework incorporating Random Forest, Support Vector Machines, and Neural Networks for jamming detection in wireless communications, achieving high accuracy and low false alarm rates, with Random Forest yielding the best performance. The importance of tackling jamming threats has been highlighted at major global security events like the Munich Security Conference, the UN Counter-Terrorism Week, and the International Conference on Cyber Conflict [6] [7] [8]. Despite these advancements, detection remains difficult under diverse jamming scenarios. Zhu [10] proposed a Markov Decision Process-based adaptive anti-jamming technique that enhances both resilience and energy efficiency. Furthermore, Del-Valle-Soto and Mex-Perera [11] suggested a distributed detection mechanism aimed at improving packet delivery and robustness against attacks. However, conventional approaches continue to face notable challenges, including limited adaptability to novel jamming strategies, high computational requirements, and scalability constraints in real-world deployments.

Recent advancements in artificial intelligence (AI) have enabled the development of more resilient and adaptable jamming detection frameworks. Building upon traditional methods, Hussain et al. [12] proposed a TinyML-based edge AI solution for IoT networks, demonstrating high accuracy and low latency; however, the approach was limited by device-specific tuning and poor scalability. For narrowband interference detection, Robinson et al. [13] employed convolutional neural networks (CNNs), achieving 99% accuracy, though the method demanded extensive training data and lacked interpretability. In 5G UAV networks, Viana and Farkhari [14] proposed a hybrid transformer-PCA model, which performed well in both Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) conditions, though it came with high computational costs. While AI-powered methods make detection faster, smarter, and more adaptable, but they also come with challenges like higher resource demands, scalability issues, and a lack of

transparency.These problems shows us why we need smart and balanced systems that can work well in all kinds of changing wireless environments.

Motivated by the need to enhance the security and reliability of critical communication systems, we developed an AI-driven framework for jamming detection. This framework leverages machine learning (ML) algorithms, including K-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), and XG-Boost, to identify jamming signals by analyzing key features such as signal strength, signal-to-noise ratio (SNR), and error patterns [15] [16]. To address the challenge of class imbalance in detecting rare jamming events, we incorporate the Synthetic Minority Over-sampling Technique (SMOTE), which enhances model performance by generating synthetic jamming data [17]. This optimization ensures the system's robustness across various communication protocols and environments, even under challenging signal conditions.

A. Research Contribution

Following the major research contributions of the proposed framework, we introduced an AI-based framework for jamming detection, optimized with SMOTE, to enhance the reliability and accuracy of communication systems under attack. We utilized a large-scale dataset with features from real-world communication systems, representing both jamming and non-jamming signals across various scenarios [18]. After thorough preprocessing of the dataset, we trained five classification models, including KNN, RFC, SVM, LR, and XGBoost, for jamming detection, and compared their performance in terms of accuracy, log loss, precision-recall curves, and cross-validation. From the in-depth analyses, including precision-recall curves and cross-validation,We concluded that the XG-Boost model outperformed other classifiers, demonstrating its robustness and precision, particularly when combined with SMOTE-based optimization [19].

B. Organization

The rest of the paper is organized into the following sections. Section II discusses the system model and problem formulation for the proposed framework. In Section III, we define and detail the layers of the proposed framework. Section IV discusses the experimental setup and results of the performance analysis. Section V provides the conclusion.

II. System Model and Problem Formulation

A. System Architecture

We model the jamming detection system as a layered analytical architecture that transforms raw RF signal statistics into actionable intelligence. Formally, let the set of wireless transmissions be denoted as:

$$\mathcal{F} = \{f_i | i = 1, 2, \ldots, N\}, f_i \in \mathbb{R}^{39} \tag{1}$$

Each instance f_i includes temporal and physical-layer features such as RSSI, SINR, PKT_SIZE, POWER, NOISE, etc. These are mapped to an outcome variable $y_i \in \{0, 1\}$, where 1 denotes the presence of a jamming attack.

B. Mathematical Formulation of Dataset

Let $\mathcal{D} = \{(f_i, y_i)\}_{i=1}^{N}$ denote the dataset, with $N = 92{,}486$ instances. Each f_i lies in the high-dimensional feature space $\mathbb{R}^d$, where $d = 39$.

We represent feature partitions as:

$$f_i = [\text{RSSI}_i, \text{SINR}_i, \text{PKT\_SIZE}_i, \text{POWER}_i, \text{NOISE}_i, \ldots] \in \mathbb{R}^{39} \tag{2}$$

and the label:

$$y_i = \Phi(f_i) + \varepsilon_i \tag{3}$$

where Φ is a nonlinear mapping induced by the jamming pattern, and ε_i is the latent noise from channel randomness.

To stabilize the feature distribution, we apply a transformation $\mathcal{T}$:

$$f_i' = \mathcal{T}(f_i) = \frac{f_i - \mu}{\sigma}, \quad \mu, \sigma \in \mathbb{R}^{39} \tag{4}$$

which yields the standardized dataset $\mathcal{D}' = \{(f_i', y_i)\}$ with zero mean and unit variance across all features.

C. Class Imbalance Handling and SMOTE Mapping

Let the prior distribution of labels be imbalanced such that:

$$P(y = 1) \ll P(y = 0) \tag{5}$$

To address this, we construct a synthetic expansion $\mathcal{D}_{\text{SMOTE}}$ using k-nearest neighbors in $\mathbb{R}^{39}$ for the minority class. For each minority point f_i:

$$f_i^{(\text{new})} = f_i + \lambda(f_i^{(k)} - f_i), \quad \lambda \sim \mathcal{U}(0, 1) \tag{6}$$

where $f_i^{(k)}$ is a random neighbor and $\mathcal{U}$ denotes a uniform distribution. The final balanced training set is:

$$\mathcal{D}_{\text{train}} = \mathcal{D}' \cup \mathcal{D}_{\text{SMOTE}} \tag{7}$$

This method, known as Synthetic Minority Over-sampling Technique (SMOTE), improves classifier sensitivity on minority classes [**?**].

D. Classifier Abstraction and Training

Let $\mathcal{M}_j : \mathbb{R}^{39} \to \{0, 1\}$ denote the classifier function trained on (f_i', y_i). We define the learning transformation as:

$$\hat{y}_i = \mathcal{M}_j(f_i') = \Theta_j^\top \phi(f_i') + b_j \tag{8}$$

where ϕ is the feature projection (kernelized for SVM/XGB), and Θ_j and b_j are model-specific learnable parameters.

For ensemble-based models such as RFC and XGB, the final hypothesis function is:

$$\mathcal{M}_{\text{XGB}}(f_i') = \sum_{t=1}^{T} \alpha_t h_t(f_i'), \quad h_t \in \mathcal{H} \tag{9}$$

where T is the number of base learners and $\mathcal{H}$ is the set of decision trees. XGBoost is particularly well-suited for structured datasets due to its regularization and scalability features [**?**].

E. Evaluation Dynamics

Model performance is measured using several evaluation tools. The loss distribution per fold in 3-fold cross-validation is:

$$\mathcal{L}_j^{(k)} = \frac{1}{|D_k|} \sum_{(f,y)\in D_k} \ell(y, \hat{y}), \quad k = 1, 2, 3 \tag{10}$$

where ℓ is the negative log-likelihood loss:

$$\ell(y, \hat{y}) = -\left[y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})\right], \quad \hat{p} = \mathcal{M}_j(f) \tag{11}$$

Additionally, for the PR curve, precision and recall pairs $\{(\text{Pr}_\tau, \text{Re}_\tau)\}_{\tau=0}^{1}$ are extracted to form:

$$PR_{\mathcal{M}_j} = \left\{ \left(\frac{TP_\tau}{TP_\tau + FP_\tau}, \frac{TP_\tau}{TP_\tau + FN_\tau} \right) \middle| \tau \in [0, 1] \right\} \tag{12}$$

Let CV_j denote the cross-validation curve of classifier j:

$$CV_j = \left\{ \mathcal{L}_j^{(1)}, \mathcal{L}_j^{(2)}, \mathcal{L}_j^{(3)} \right\} \tag{13}$$

We also define the expected accuracy over the validation set for model $\mathcal{M}_j$ as:

$$\text{Acc}_j = \frac{1}{|\mathcal{D}_{val}|} \sum_{(f,y)\in\mathcal{D}_{val}} \mathbb{I}[\mathcal{M}_j(f') = y] \tag{14}$$

where $\mathbb{I}$ is the indicator function returning 1 if the prediction matches the ground truth, and 0 otherwise.

F. Experimental Results Interpretation

Empirical experiments reveal the following ranking in terms of log loss and detection robustness under SMOTE:

$$\mathcal{L}_{\text{XGB}}^{\text{SMOTE}} < \mathcal{L}_{\text{RFC}}^{\text{SMOTE}} < \mathcal{L}_{\text{LR}}^{\text{SMOTE}} < \mathcal{L}_{\text{SVM}}^{\text{SMOTE}} < \mathcal{L}_{\text{KNN}}^{\text{SMOTE}} \tag{15}$$

with similar trends for accuracy. The decision boundary learned by XGB under balanced conditions exhibits sharp discrimination in high-dimensional $\mathbb{R}^{39}$ space. Models trained on $\mathcal{D}_{\text{SMOTE}}$ exhibit higher precision-recall areas and lower variance in CV loss, validating their reliability for real-time jamming detection in critical wireless networks.

III. Proposed Framework

The proposed ML-based framework for jamming detection in critical communication systems consists of three layers.

A. Data Layer

This layer processes the dataset by handling missing values, encoding, and scaling. SMOTE addresses class imbalance by generating synthetic minority samples:

$$x_{\text{new}} = x_i + \delta \cdot (x_{\text{nn}} - x_i), \quad \delta \sim U(0, 1)$$

where x_i is a minority class instance and x_{nn} is its nearest neighbor.

B. AI Layer

Five machine learning models—logistic regression, KNN, random forest, SVM, and XGBoost—are trained on the balanced data. Cross-validation identifies XGBoost as the most effective model due to its accuracy, robustness, and ability to handle complex patterns with regularization,

C. Application Layer

The trained XGBoost model classifies transmissions as normal or jammed, enabling fast detection and response, such as rerouting traffic or alerting personnel. The layer includes a user interface for real-time monitoring and integrates with existing communication infrastructure for seamless operation.

D. Dataset Description

This study utilizes the Book2.csv dataset, derived from a wireless communication monitoring system, which contains 92,486 records, each with 39 features. These features include signal-based metrics (e.g., mean frequency, standard deviation, maximum amplitude) and statistical properties (e.g., energy, entropy) that characterize the state of the communication channel. The binary target variable, "attack," is labeled 1 for jamming events (such as interference caused by gunshots) and 0 for normal transmissions. Due to the class imbalance in the dataset, where jamming events are underrepresented, SMOTE was applied to oversample the minority class (jamming), helping to improve the model's ability to generalize across both classes.

E. Dataset Preprocessing

To prepare the dataset for modeling, several preprocessing steps were applied. Missing values were removed, and categorical features were encoded using label encoding or one-hot encoding, depending on the type of variable. A Pearson correlation heatmap was used to identify and eliminate highly correlated features, reducing redundancy and ensuring that only the most relevant features remained. All features were then normalized using min-max scaling, ensuring that each feature contributed equally to the model training. To address the class imbalance, SMOTE was used to synthetically generate more samples of the minority class, thereby balancing the dataset and improving the model's ability to detect jamming events.

F. Model Motivation

After preprocessing, these five machine learning models, namely Logistic Regression, KNN, Random Forest, SVM, and XGBoost—were trained and evaluated. XGBoost consistently outperformed the other models in terms of accuracy, log loss, precision-recall curves, and cross-validation stability. It demonstrated strong performance in handling complex patterns, was less prone to overfitting due to its built-in regularization, and showed robustness even when applied to the SMOTE-balanced dataset. Given its superior performance and ability to generalize effectively, XGBoost was selected as the final model for detecting jamming events in wireless communication systems.

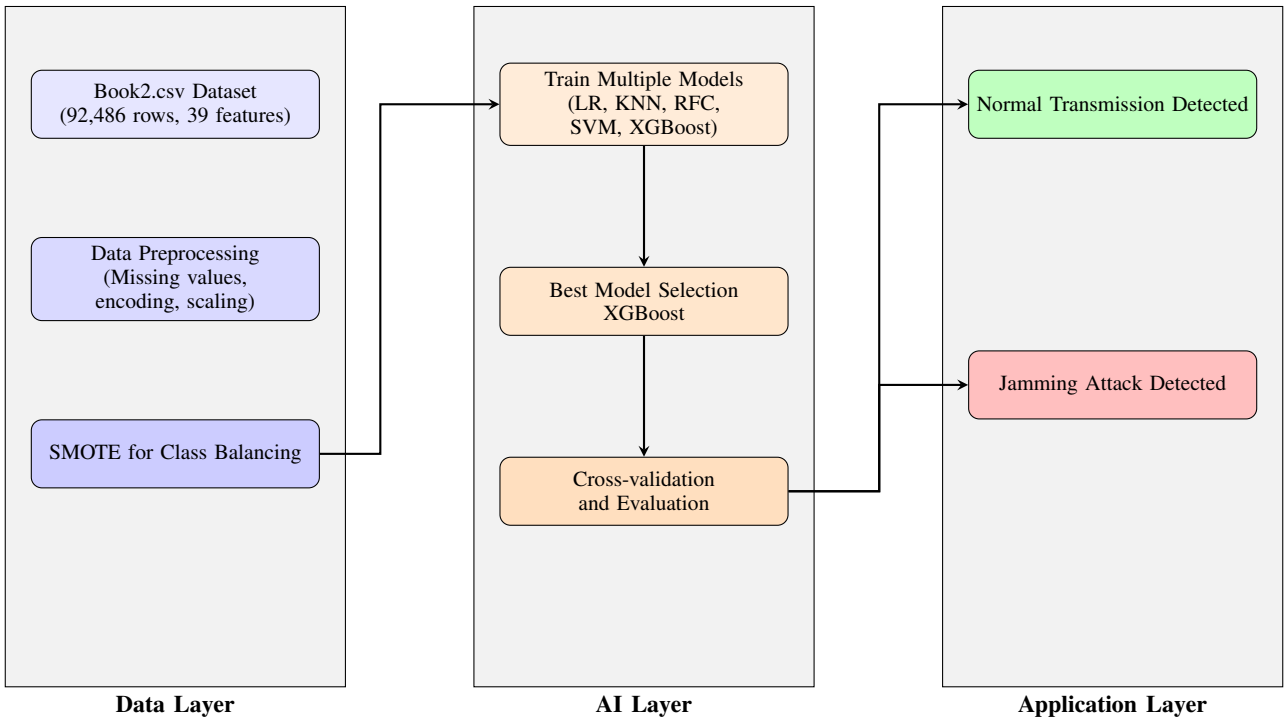

Fig. 1: Proposed Jamming Detection Framework

Fig.1 depicts the proposed jamming detection framework with three layers: Data, AI, and Application. The Data Layer processes the dataset through cleaning, encoding, scaling, and SMOTE for class balancing. The AI Layer trains multiple ML models, with XGBoost emerging as the best performer after cross-validation. The Application Layer uses the trained XGBoost model to classify transmissions as normal or jammed, enabling effective detection of wireless signal disruptions.

IV. Results and Discussion

A. Experimental Setup and Parameters

The experiments were conducted on a personal computer equipped with an Intel Core i7-11390H processor (3.40 GHz), 16 GB RAM, and Windows 11 Home. Development was carried out using Jupyter Notebook (Anaconda 2.4.0) and Google Colab. Key Python libraries included pandas and numpy for data manipulation, matplotlib and seaborn for visualization, scikit-learn for machine learning implementation and evaluation, xgboost for gradient boosting, and imblearn for handling class imbalance via SMOTE. The main hyperparameters were: KNN with 5 neighbours, SVM with an RBF kernel (C=1.0), logistic regression with the 'lbfgs' solver and 1000 iterations, random forest with 100 estimators, and XGBoost optimized using the 'logloss' metric. These settings were selected to balance model performance and efficiency.

B. Model Performance Metrics

We evaluated our classifiers using four key metrics: log loss, accuracy, precision-recall (PR) curves, and cross-validation stability.

TABLE I: Classifier Hyperparameters

Classifier	Key Parameters
KNN	n_neighbors = 5
SVM	kernel = 'rbf', C = 1.0
LR	solver = 'lbfgs', max_iter = 1000
RFC	n_estimators = 100
XGBoost	eval_metric = 'logloss'

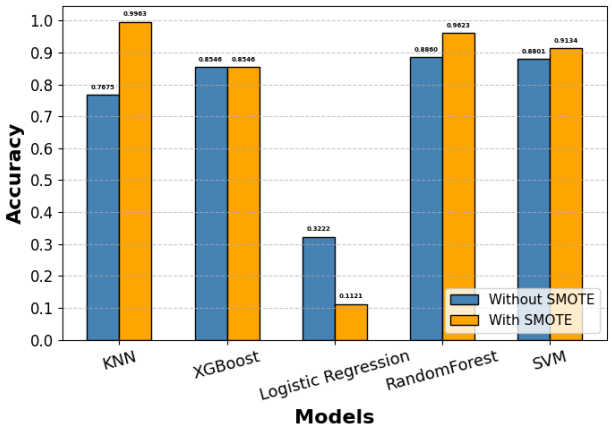

Fig. 1: Accuracy Comparison of Models With and Without SMOTE

1) Accuracy Analysis: Figure 1 shows how the accuracy of each model changed before and after applying SMOTE. K-Nearest Neighbors (KNN) saw the biggest improvement, with its accuracy jumping from 76.75% to 99.63%. This big

boost highlights how sensitive KNN is to class imbalances and how much SMOTE helps in improving its performance. Random Forest also improved a lot, with accuracy going up from 88.60% to 96.23%, showing that ensemble methods like Random Forest work well when the data is balanced. Support Vector Machine (SVM) had a smaller improvement, moving from 88.01% to 91.34%. This suggests that while SVM benefits from SMOTE, it's less affected by class imbalance compared to KNN or Random Forest. XGBoost, on the other hand, showed pretty stable performance, with accuracy staying around 85.46%, meaning it's already quite robust to class imbalance. Surprisingly, Logistic Regression didn't follow the same pattern. Its accuracy dropped sharply from 32.22% to 11.21% after SMOTE. This drop suggests that Logistic Regression might struggle with synthetic data from SMOTE, possibly because its linear decision boundaries don't work well with the artificially created data points.

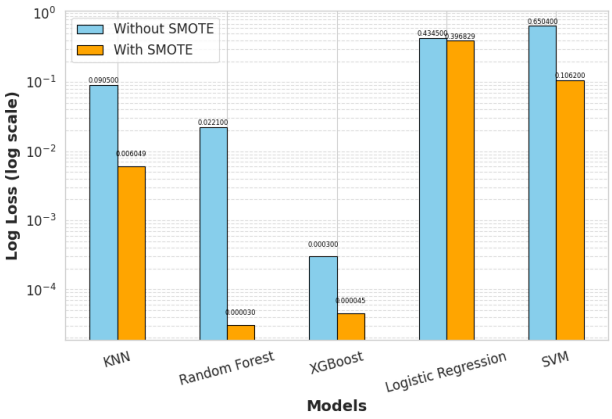

Fig. 2: Log Loss Comparison of Models With and Without SMOTE

2) Log Loss Analysis: As shown in Figure 2, the use of SMOTE had a clear impact on the log loss of the models. For Random Forest, the log loss dropped dramatically from 0.0221 to almost zero (0.00003), indicating that the model became much more confident and accurate in its predictions. Similarly, KNN saw a sharp improvement in log loss, going from 0.0905 to 0.0060, which shows that KNN performed better when the data was balanced and became more reliable. XGBoost, which was already performing well, showed a small improvement in log loss, decreasing from 0.0003 to 0.000045, indicating even more precise predictions. SVM also experienced a significant reduction in log loss, dropping from 0.6504 to 0.1062, meaning SMOTE improved its confidence in predictions, although it still didn't perform as well as Random Forest. On the other hand, Logistic Regression only showed a modest improvement, with its log loss changing from 0.4345 to 0.3968, suggesting that it struggled more with synthetic data from SMOTE. Overall, SMOTE improved predictive certainty across most models, with the most significant benefits observed in tree-based models like Random Forest and instance-based models like KNN.

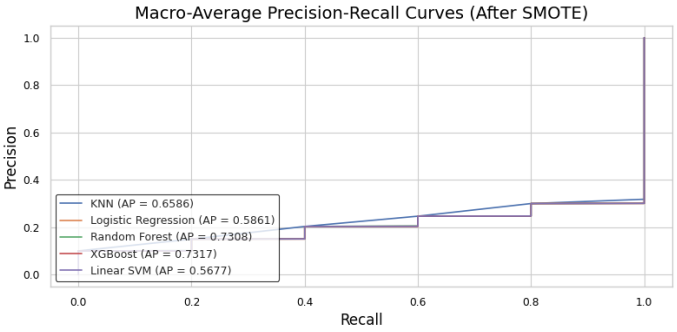

Fig. 3: Macro-Average Precision-Recall Curves of All Models After SMOTE

3) Precision-Recall Curve Analysis After SMOTE: Figure 3 presents the macro-average Precision-Recall (P-R) curves for all classifiers after applying SMOTE. XGBoost achieved the highest AP score of 0.7317, followed by Random Forest (0.7075) and KNN (0.6586), indicating effective learning of minority classes. Logistic Regression and Linear SVM yielded moderate scores of 0.5861 and 0.5677, respectively. In terms of improvement, KNN recorded the highest AP gain—from 0.3545 to 0.6586 (+0.304). XGBoost improved by +0.055, Random Forest by +0.031, SVM by +0.053, and Logistic Regression by +0.040. KNN benefitted most from SMOTE, likely due to enhanced neighborhood structure. Tree-based models also improved due to better data balance. Linear models saw limited gains, constrained by their simplicity. Overall, SMOTE enhanced class balance learning across all models, especially KNN and ensemble methods.

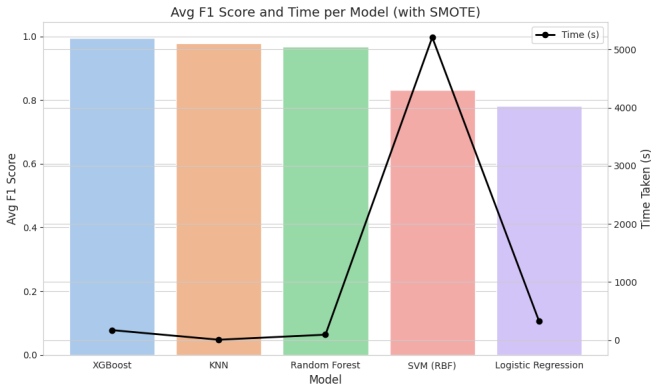

Fig. 4: Average Cross-Validation Score and Time per Model (with SMOTE)

4) Cross-Validation Score and Time Analysis (After SMOTE): Figure 4 presents a comparison of average cross-validation (CV) scores and training times across different models after the application of SMOTE. XGBoost achieved the highest CV score, demonstrating strong generalization capability, followed closely by Random Forest and KNN. Although KNN exhibited slightly lower accuracy, it was the fastest to train, indicating superior computational efficiency. In contrast,

SVM (RBF) resulted in the lowest CV score and the longest training time, highlighting its poor scalability for the given dataset. Logistic Regression, while computationally efficient, showed only moderate predictive performance. Overall, the results suggest that tree-based models, particularly XGBoost and Random Forest, offer the best trade-off between accuracy and training time when SMOTE is used for class balancing.

C. Performance Analysis of the Proposed Framework

We evaluated five classifiers using key metrics—accuracy, log loss, average precision (AP) score, cross-validation score, training time, and overfitting risk—both before and after applying SMOTE. XGBoost exhibited the highest performance with an accuracy of 99.68%, an AP score of 0.7317, a cross-validation score of 0.9946, and a log loss of 0.000045. Although XGBoost required a moderate training time of 57.69 seconds, it necessitated careful hyperparameter tuning to mitigate overfitting risks. Random Forest achieved an accuracy of 97.05%, an AP score of 0.7075, the shortest training time of 11.54 seconds, a cross-validation score of 0.9705, and a log loss of 0.00003, offering a good balance between efficiency and accuracy. KNN demonstrated significant improvement after SMOTE, with the AP score increasing from 0.3545 to 0.6586, a cross-validation score of 0.9760, and a log loss reduction from 0.0905 to 0.0060, although it remained sensitive to noise.

SVM (RBF) showed moderate performance with an AP score of 0.5677, a cross-validation score of 0.9100, and a log loss of 0.1062, but suffered from an excessively long training time of 358 seconds, limiting its scalability. Logistic Regression underperformed across all metrics, showing minimal improvement after SMOTE, with an accuracy decline from 32.22% to 11.21% and only a slight reduction in log loss from 0.4345 to 0.3968. Overall, the application of SMOTE significantly enhanced the performance of all classifiers by improving accuracy, AP scores, and reducing log loss. XGBoost and KNN exhibited the most notable improvements, Random Forest maintained a strong balance between performance and overfitting risk, SVM's long training time remained a major drawback, and Logistic Regression proved to be the least effective model for jamming detection under the given conditions.

V. Conclusion

This paper presents a data-driven framework for detecting jamming attacks, including those caused by gunshot interference, in critical communication systems. The framework leverages machine learning algorithms and the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance. Evaluating five classifiers—Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, and XGBoost—our results show significant performance improvements, particularly in recall and prediction accuracy. Among the models, XGBoost achieved the best results, with 99.68 percent accuracy, a precision-recall area of 0.7317, and strong cross-validation stability. Random Forest also performed well with low overfitting risk, while KNN showed improvements after applying SMOTE, although it remained sensitive to noise. These results highlight the importance of model selection and data preprocessing for reliable jamming detection. This research contributes a scalable, interpretable ML approach to enhance the security of critical infrastructure against wireless signal interference, with future work focusing on deep learning, real-time deployment, and advanced evaluation metrics like ROC-AUC and detection latency.

References

[1] H. Pirayesh *et al.*, "Jamming attacks and detection in wireless networks," *IEEE Communications Surveys Tutorials*, 2018.

[2] ——, "A survey on wireless jamming and anti-jamming techniques," *IEEE Communications Surveys Tutorials*, 2022.

[3] A. Al-Shaihk *et al.*, "Review of jamming in communication systems," *International Journal of Wireless Communications*, 2019.

[4] O. GROUP. (2024) Gps interference and spoofing reports. [Online]. Available: https://ops.group

[5] A. Jaitly *et al.*, "Impacts of gps jamming on aviation," *Journal of Aviation Technology*, 2017.

[6] "Munich security conference 2025." Munich Security Conference, 2025. [Online]. Available: https://securityconference.org

[7] "Un counter-terrorism week." United Nations, 2023. [Online]. Available: https://www.un.org/counterterrorism

[8] "International conference on cyber conflict (cycon)." NATO CCDCOE, 2023. [Online]. Available: https://ccdcoe.org

[9] Y. Arjoune, N. Kaabouch, H. El Ghazi, and J. Ma, "A novel machine learning approach for classification and detection of jamming attacks in wireless sensor networks," *Computers & Security*, vol. 89, p. 101660, 2020.

[10] Q. Zhu and T. Basar, "Optimal jamming-resistant communication over an arbitrarily varying channel," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1865–1878, 2012.

[11] C. Del-Valle-Soto and C. Mex-Perera, "A new distributed detection method for jamming attacks in wireless sensor networks," *Sensors*, vol. 19, no. 16, p. 3573, 2019.

[12] F. Hussain, M. Khan, and V. Sharma, "Jamming detection in iot networks using edge ai with tinyml framework," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 160–13 170, 2022.

[13] K. Robinson, Y. Chen, and D. Lee, "Narrowband interference detection using cnns in wireless spectrum monitoring," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2023, pp. 1–6.

[14] P. Viana and S. Farkhari, "A transformer-pca hybrid framework for jamming detection in 5g uav networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 1, pp. 118–130, 2024.

[15] D. Kosmanos, D. Karagiannis, A. Argyriou, S. Lalis, and L. Maglaras, "Rf jamming classification using relative speed estimation in vehicular wireless networks," *arXiv preprint arXiv:1812.11886*, 2018.

[16] Y. Arjoune, F. Salahdine, M. S. Islam, C. Ghribi, and N. Kaabouch, "A novel jamming attacks detection approach based on machine learning for wireless communication," *arXiv preprint arXiv:2003.07308*, 2020.

[17] Y. Zhang, L. Wang, and Y. Xie, "Wireless sensor networks intrusion detection based on smote and the random forest algorithm," *Sensors*, vol. 19, no. 1, p. 203, 2019.

[18] E. Testi, L. Arcangeloni, and A. Giorgetti, "Machine learning-based jamming detection and classification in wireless networks," in *Proceedings of the 2023 ACM Workshop on Wireless Security and Machine Learning (WiseML)*. ACM, 2023, pp. 39–44.

[19] T. O. Omotehinwa, M. Onoja, and D. Oyewola, "Oversampling-adaptive search optimization of extreme gradient boost model on an imbalanced spam email dataset," in *Proceedings of the International Conference on Communication and E-Systems For Economic Stability (CeSES'2023)*, 2023. [Online]. Available: https://www.researchgate.net/publication/376636770_Oversampling-Adaptive_Search_Optimization_of_Extreme_Gradient_Boost_Model_on_an_Imbalanced_Spam_Email_Dataset