Jurnal Komputer

Vol. 1, No. 1, October 2026

SAH PUBLISHER

DDoS Attack Detection with LightGBM in DART Mode: Benchmarking GridSearchCV Against Optuna Hyperparameter Tuning

Immanuel Yosua Surianto[1*]

[1]Universitas Sugeng Hartono, Sukoharjo, Indonesia

Email: yosuasurianto72@gmail.com

Abstract. Distributed Denial of Service (DDoS) attacks continue to threaten servers, websites, and critical infrastructure, with SYN flood remaining one of the most common attack types. Manual detection through firewall rules is slow, non-adaptive to new patterns, and dependent on scarce security expertise. This paper presents a machine learning approach that classifies network traffic as benign or SYN flood attack using LightGBM in DART boosting mode, and benchmarks two hyperparameter tuning strategies: GridSearchCV and Optuna. Experiments use the CIC-DDoS2019 dataset, specifically the SYN flood training subset of 3,517 samples with 37.3% benign and 62.7% attack traffic. Preprocessing removes identity columns that cause data leakage, handles infinite values, encodes labels, and keeps numeric features, while class imbalance is addressed with balanced class weights. The model is evaluated with stratified 5-fold cross-validation using precision, recall, F1-score, and F1-macro. GridSearchCV exhaustively tests 4 discrete combinations of max_depth and num_leaves, while Optuna explores a wider continuous space over 8 trials with Bayesian TPE, keeping the learning rate fixed at 0.1 for a fair comparison. GridSearchCV achieves an F1-macro of 0.9912 in 38.44 seconds, while Optuna achieves 0.9909 in 30.15 seconds, making Optuna approximately 21% faster with a negligible F1 difference of 0.0003. Confusion matrices show near-perfect detection with only 29 false negatives in both models, and feature importance analysis identifies packet size, flow duration, and inter-arrival time as the key discriminating signals. The main contribution is empirical evidence that Bayesian hyperparameter optimization matches exhaustive grid search quality while substantially reducing tuning time for intrusion detection models.

Received August 2026 / Revised September 2026 / Accepted October 2026

This work is licensed under a Creative Commons Attribution 4.0 International License.

Keywords: DDoS detection; LightGBM; DART; hyperparameter tuning; Optuna; GridSearchCV; CIC-DDoS2019

INTRODUCTION

Distributed Denial of Service (DDoS) attacks overwhelm servers, websites, and critical infrastructure with massive traffic volumes, causing service disruption and financial loss. Among attack types, SYN flood exploits the TCP handshake mechanism to exhaust server resources until the target crashes [1]. Traditional defenses based on manually configured firewall rules are slow to update, cannot adapt to novel attack patterns, and require specialized security expertise. Machine learning offers an automatic and adaptive alternative by learning traffic behavior patterns from data [2].

Public benchmark datasets have enabled systematic progress in this area. The CIC-DDoS2019 dataset provides realistic DDoS attack captures with a detailed taxonomy [3], building on the earlier CIC-IDS2017 intrusion detection dataset [4]. Deep learning approaches such as DDoSNet and LUCID have shown strong detection performance on these datasets [5][6], while ensemble methods including CNN ensembles and classical classifiers remain competitive [7][8]. Gradient boosting frameworks are particularly attractive for tabular network traffic features: XGBoost demonstrated effective DDoS detection in software-defined networks [9], XGBoost-based intrusion detection systems perform well on benchmark data [10], and CatBoost offers unbiased boosting with categorical features [11]. Feature engineering and model optimization further improve classification quality [12].

LightGBM, a highly efficient gradient boosting decision tree framework, is well suited to large-scale tabular data [13]. Its DART mode brings the dropout technique from deep learning into gradient boosting: at each iteration a portion of trees is randomly dropped so that new trees correct the errors of the remaining ones, preventing early trees from dominating and reducing overfitting [14]. Hyperparameter

tuning critically affects such models. Grid search exhaustively evaluates discrete combinations but wastes computation on poor regions [15], while Bayesian optimization learns from previous trials to focus on promising areas [16], with tree-structured Parzen estimators providing an efficient surrogate model [17]. Optuna implements this approach in a modern, define-by-run optimization framework [18].

This paper benchmarks GridSearchCV against Optuna for tuning a LightGBM classifier in DART mode on SYN flood detection. The contributions are (1) a leakage-aware preprocessing pipeline for the CIC-DDoS2019 SYN subset, (2) a controlled comparison of exhaustive versus Bayesian tuning with the learning rate fixed, and (3) empirical evidence on detection quality, tuning speed, confusion matrices, and feature importance. Class imbalance is handled through balanced class weights [19], and evaluation uses stratified cross-validation [20] with precision-recall based metrics appropriate for imbalanced security data [21][22].

METHODS

Dataset

The experiments use the CIC-DDoS2019 dataset from the Canadian Institute for Cybersecurity at the University of New Brunswick, which provides realistic DDoS attack captures with a documented taxonomy [3]. The specific file used is the SYN flood training subset (Syn-Training.parquet), containing 3,517 samples representing 5% of the original data. Class composition is shown in Table 1: 1,313 benign samples (37.3%) and 2,204 SYN flood attack samples (62.7%). The dataset is imbalanced with attacks dominant, which is addressed through balanced class weights rather than resampling [23][24].

Table 1. Class Composition of the Training Data

Class	Count	Percentage
Benign (Normal)	1,313	37.3%
SYN Flood Attack	2,204	62.7%
Total Samples	3,517	100%

Preprocessing

Preprocessing follows a leakage-aware pipeline illustrated in Fig. 1. Data leakage occurs when features reveal information that would be unavailable at prediction time, causing the model to memorize rather than learn attack behavior. Table 2 lists the removed columns: Flow ID, Source IP, and Destination IP are identity attributes rather than traffic behavior features; Timestamp is irrelevant for classification; and SimillarHTTP is unrelated to the task. Additional cleaning replaces infinite values in numeric features with NaN and drops rows containing NaN, encodes the categorical labels numerically (benign as 0, SYN flood as 1), and retains only numeric features to ensure consistent input types.

Table 2. Removed Columns for Data Leakage Prevention

Column	Reason for Removal
Flow ID, Source IP, Destination IP	Specific identities, not traffic behavior features
Timestamp	Not relevant for classification
SimillarHTTP	Column not relevant to the classification task

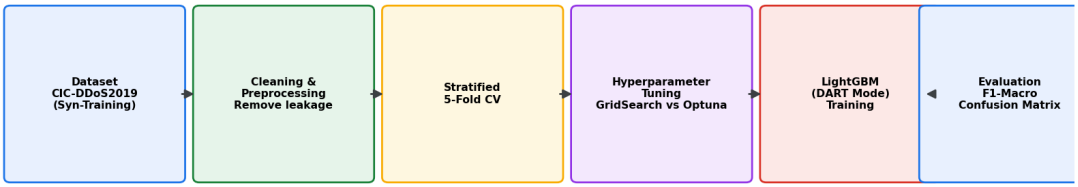


Figure 1. Experiment Pipeline from Dataset to Evaluation

LightGBM with DART Boosting

LightGBM is a gradient boosting framework based on decision trees that builds trees sequentially, with each new tree correcting the errors of the previous ones, and is highly efficient for large tabular data [13]. DART (Dropouts meet Multiple Additive Regression Trees) adapts the dropout technique to boosting: at each iteration a fraction of existing trees is randomly dropped, a new tree is built to correct the errors of the remaining trees, and the dropped trees are reintroduced, which prevents early trees from dominating the ensemble and mitigates overfitting [14]. The fixed hyperparameters are listed in Table 3: 50 estimators, a dropout rate of 0.2, balanced class weights to handle the imbalanced classes, and a binary objective for the two-class problem.

Table 3. Fixed Hyperparameters of the LightGBM Model

Parameter	Value	Function
n_estimators	50	Total number of trees built
drop_rate	0.2	Dropout rate in DART mode
class_weight	balanced	Handles class proportion imbalance
objective	binary	Binary classification (2 classes)

Hyperparameter Tuning

Two tuning strategies are compared under controlled conditions, with the search spaces summarized in Table 4. GridSearchCV performs brute-force evaluation of 4 discrete combinations: max_depth in {5, 10} and num_leaves in {20, 40}, with no adaptation between trials [15]. Optuna performs Bayesian optimization with the tree-structured Parzen estimator [17][16]: it learns from previous trials, avoids poor regions, and explores a wider continuous space over 8 trials with max_depth in [5, 12] and num_leaves in [20, 50] [18]. The learning rate is fixed at 0.1 in both methods so that the comparison isolates the search strategy for max_depth and num_leaves rather than confounding it with learning rate differences.

Table 4. Hyperparameter Search Spaces

Parameter	GridSearchCV	Optuna
max_depth	{5, 10}	suggest_int(5, 12)
num_leaves	{20, 40}	suggest_int(20, 50)
learning_rate	0.1 (fixed)	0.1 (fixed)

Strategy	Brute-force, 4 combinations	Bayesian TPE, 8 trials

Evaluation Protocol

Model evaluation uses stratified 5-fold cross-validation: the data is split into 5 folds, each preserving the benign-to-attack class ratio, and the model is trained and validated 5 times in rotation, yielding a stable performance estimate unbiased by any single split [20]. The metrics are precision (TP/(TP+FP)), recall (TP/(TP+FN)), F1-score (2PR/(P+R)), and F1-macro, the average of per-class F1 scores, which evaluates fairly without bias toward the majority class. In cybersecurity, recall is the most critical metric because missed attacks (false negatives) are far more dangerous than false alarms (false positives) [22].

RESULT AND DISCUSSION

Benchmarking Results

Table 5 compares the two tuning strategies. GridSearchCV achieves an F1-macro of 0.9912 in 38.44 seconds, while Optuna achieves 0.9909 in 30.15 seconds. Optuna is approximately 21% faster, the F1 difference of 0.0003 is practically insignificant, and Optuna explores a wider parameter space in less time. The speed advantage comes from the search strategy: GridSearchCV must test every combination including poorly performing ones with no mechanism to learn from previous results, while Optuna's Bayesian TPE avoids bad trials and concentrates the search on promising regions of the parameter space.

Table 5. Benchmarking Comparison of Tuning Strategies

Method	Execution Time (s)	F1-Macro
GridSearchCV	38.44	0.9912
Optuna (Bayesian TPE)	30.15	0.9909

Classification Performance

Table 6 reports the classification reports for both models. Precision and recall are high for both classes: benign traffic reaches 0.98 precision and 1.00 recall, while SYN attacks reach 1.00 precision and 0.99 recall, with macro averages of 0.99 across the board. The near-identical reports confirm that the choice of tuning strategy does not sacrifice prediction quality.

Table 6. Classification Reports

Class	GridSearch CV Precision	GridSearch CV Recall	GridSearch CV F1	Optuna Precision	Optuna Recall	Optuna F1
Benign	0.98	1.00	0.99	0.98	1.00	0.99
SYN Attack	1.00	0.99	0.99	1.00	0.99	0.99
Macro Avg	0.99	0.99	0.99	0.99	0.99	0.99

Confusion Matrix Analysis

Fig. 2 shows the confusion matrix for the GridSearchCV-tuned model: 1,313 true negatives (benign correctly detected), 2,175 true positives (SYN flood correctly detected), 29 false negatives (attacks missed), and 0 false positives (no false alarms). Fig. 3 shows the Optuna-tuned model: 1,312 true

negatives, 2,175 true positives, 29 false negatives, and 1 false positive. Both models miss the same 29 attack samples, representing 1.3% of total attacks that evade detection, which remains a concern in real security deployments where every missed attack matters. The single false positive of the Optuna model is negligible by comparison.

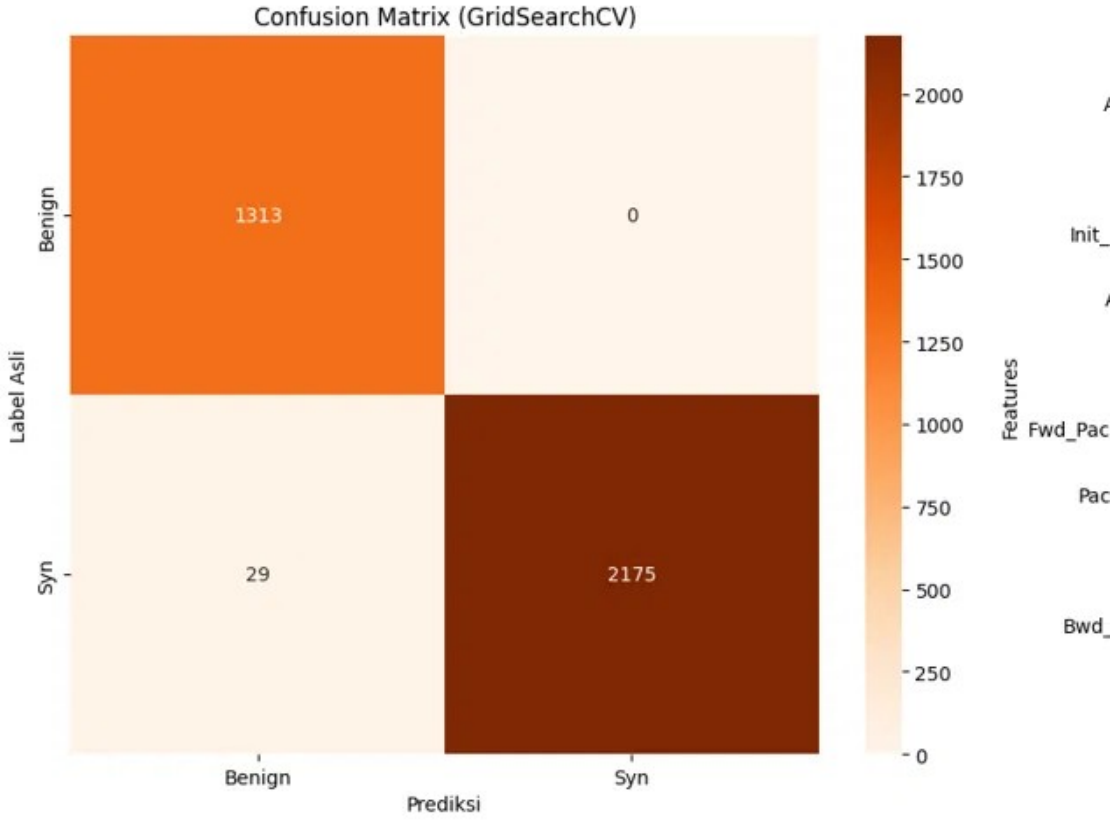


Figure 2. Confusion Matrix of the GridSearchCV-Tuned Model

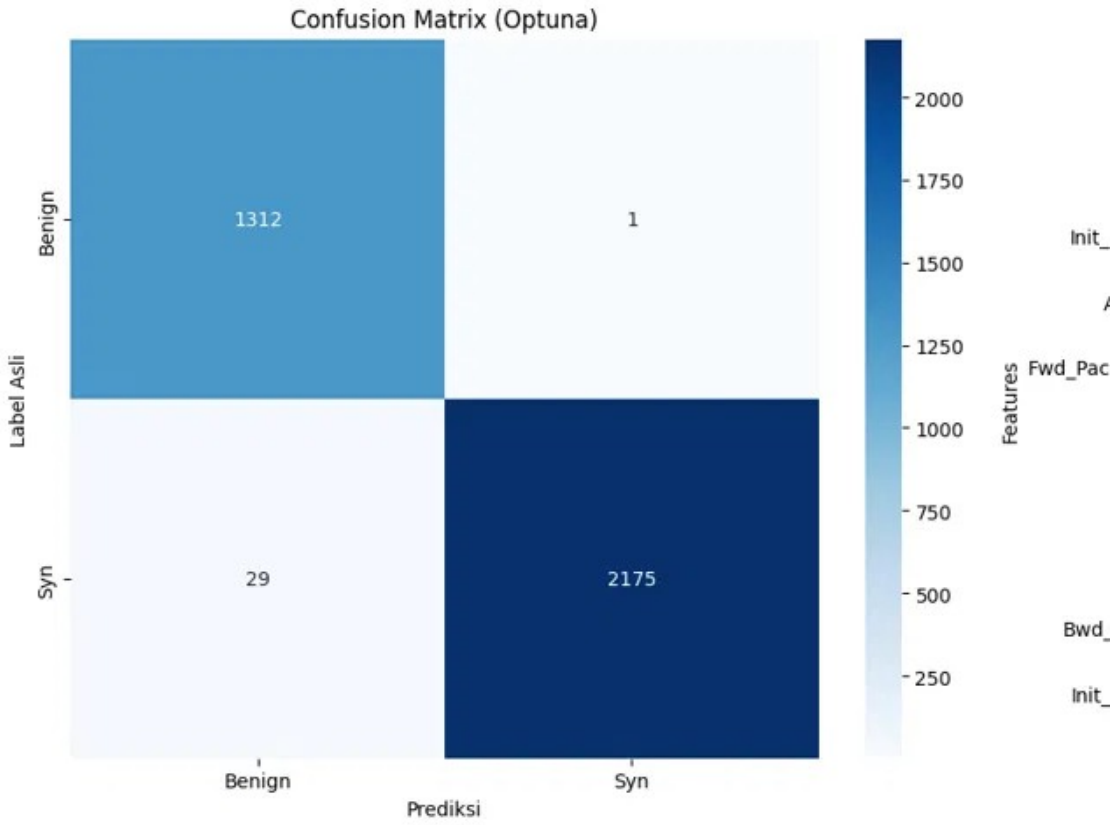


Figure 3. Confusion Matrix of the Optuna-Tuned Model

Feature Importance

Fig. 4 shows the top 10 feature importances for the GridSearchCV model, led by Avg_Packet_Size (70), Fwd_IAT_Min (63), and Init_Bwd_Win_Bytes (56). Fig. 5 shows the Optuna model, led by Flow_Duration (164), Fwd_IAT_Min (110), and Init_Bwd_Win_Bytes (83). Both models agree on the key insight: packet size, flow duration, and inter-arrival time are the primary signals distinguishing normal traffic from SYN flood attacks. The difference in the top-ranked feature reflects the different hyperparameter configurations found by each tuner, yet the discriminating feature family remains consistent, which increases confidence that the models learn genuine attack behavior rather than artifacts.

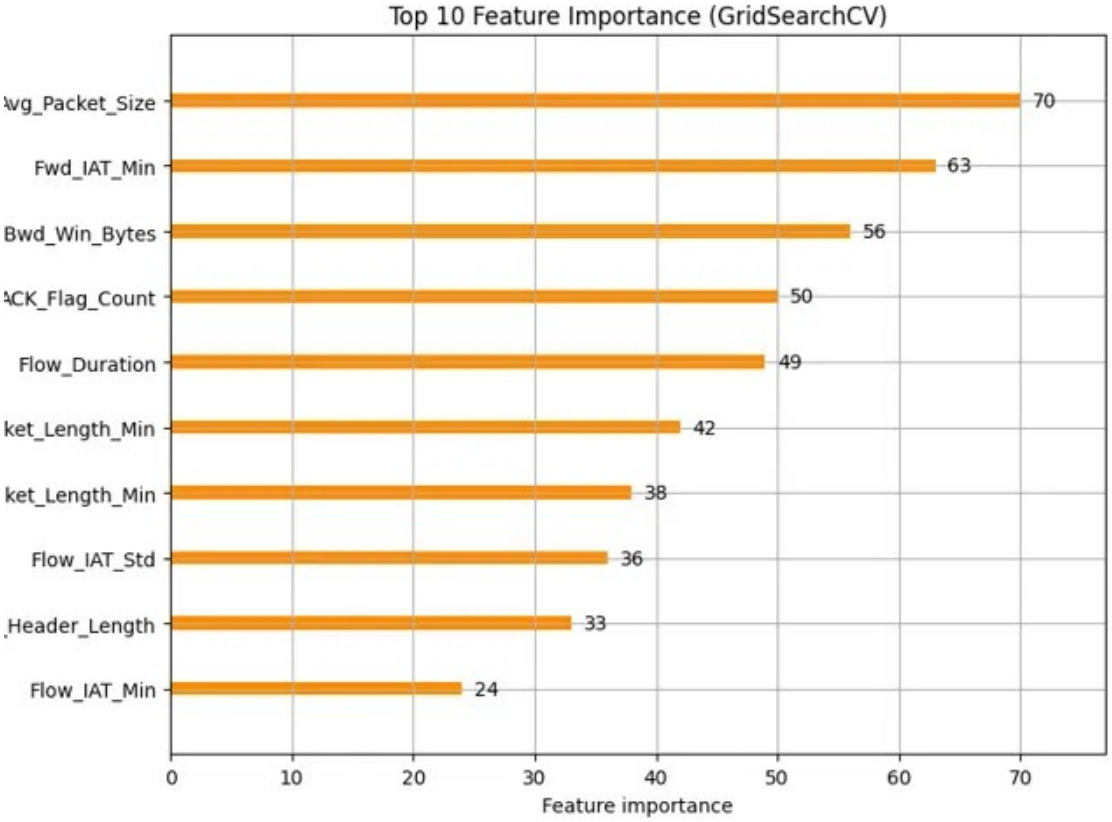


Figure 4. Top 10 Feature Importance of the GridSearchCV-Tuned Model

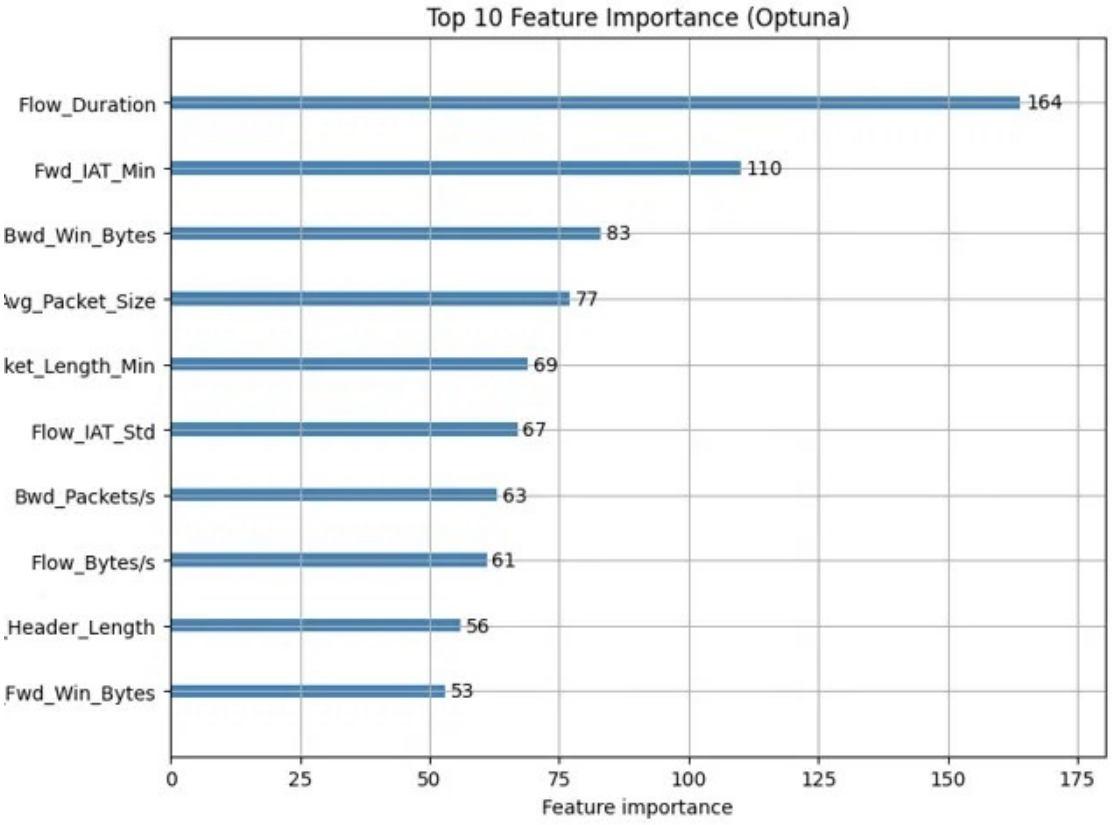


Figure 5. Top 10 Feature Importance of the Optuna-Tuned Model

Discussion

The results align with prior evidence that gradient boosting is highly effective for intrusion detection on tabular flow features [9][10]. The DART mode's dropout regularization contributes to the strong generalization observed, consistent with its design goal of preventing early-tree dominance [14]. Compared with deep learning detectors such as DDoSNet and LUCID [5][6], the LightGBM approach achieves comparable detection quality with far lower training cost and full feature-level interpretability. The tuning comparison echoes the broader hyperparameter optimization literature: Bayesian methods reach grid-comparable optima with fewer evaluations [17][16]. A limitation is that experiments cover only the SYN flood subset; extending to other DDoS attack types such as DNS, LDAP, and UDP, comparing against XGBoost, CatBoost, and Random Forest [25][11][26], and integrating the model into a real-time network monitoring system remain future work. Evaluation on the full dataset rather than the 5% subset would further strengthen the findings [27]. Alternative imbalance treatments such as adversarial SMOTE variants [28] and deep architectures tailored to imbalanced network traffic [29] were not explored here, and single-threshold summaries such as the Matthews correlation coefficient [30] could complement the reported precision-recall metrics in future evaluations.

CONCLUSION

This paper presented SYN flood detection using LightGBM in DART mode with a controlled benchmark of GridSearchCV against Optuna hyperparameter tuning on the CIC-DDoS2019 dataset. Both strategies achieve an F1-macro of 0.99 with near-perfect confusion matrices, while Optuna is approximately 21% faster and explores a wider parameter space. Feature importance analysis identifies packet size, flow duration, and inter-arrival time as the main discriminating signals. The practical implication is that for large-scale machine learning, Bayesian optimization is preferable to grid search: it delivers equal quality with substantially less tuning time. Future work includes extending coverage to other DDoS attack types, comparing additional algorithms, and deploying the model for real-time network monitoring. Comparative studies of machine-learning algorithms for DDoS prediction likewise report accuracies above 99% for ensemble methods on large network-traffic datasets [31], and AI-based intrusion detection has been evaluated for related campus-network threat scenarios [32].

REFERENCES

[1] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," IEEE Commun. Surv. Tutorials, vol. 15, no. 4, pp. 2046-2069, Fourth Quarter 2013. DOI: 10.1109/SURV.2013.031413.00127

[2] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," Cybersecurity, vol. 2, no. 1, p. 20, 2019. DOI: 10.1186/s42400-019-0038-7

[3] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in Proc. IEEE 53rd Int. Carnahan Conf. Security Technology, Chennai, India, 2019. DOI: 10.1109/CCST.2019.8888419

[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), Funchal, Madeira, Portugal, 2018, pp. 108-116. DOI: 10.5220/0006639801080116

[5] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," in Proc. IEEE 21st Int. Symp. "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland, 2020, pp. 391-396. DOI: 10.1109/WoWMoM49955.2020.00072

[6] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del-Rincon, and D. Siracusa, "LUCID: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection," IEEE Trans. Netw. Service Manag., vol. 17, no. 2, pp. 876-889, Jun. 2020. DOI: 10.1109/TNSM.2020.2971776

[7] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K.-K. R. Choo, and J. Iqbal, "A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks," IEEE Access, vol. 8, pp. 53972-53983, 2020. DOI: 10.1109/ACCESS.2020.2976908

[8] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," IEEE Access, vol. 6, pp. 33789-33795, 2018. DOI: 10.1109/ACCESS.2018.2841987

[9] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," in Proc. 2018 IEEE Int. Conf. Big Data and Smart Computing (BigComp), Shanghai, China, 2018, pp. 251-256. DOI: 10.1109/BigComp.2018.00044

[10] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective Intrusion Detection System Using XGBoost," Information, vol. 9, no. 7, p. 149, 2018. DOI: 10.3390/info9070149

[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in Proc. 32nd Conf. Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 2018, pp. 6638-6648.

[12] Y. Zhang and Z. Wang, "Feature Engineering and Model Optimization Based Classification Method for Network Intrusion Detection," Appl. Sci., vol. 13, no. 16, p. 9363, 2023. DOI: 10.3390/app13169363

[13] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Proc. 31st Conf. Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 2017, pp. 3146-3154.

[14] K. V. Rashmi and R. Gilad-Bachrach, "DART: Dropouts meet Multiple Additive Regression Trees," in Proc. 18th Int. Conf. Artificial Intelligence and Statistics (AISTATS), ser. Proceedings of Machine Learning Research, vol. 38, San Diego, CA, USA, 2015, pp. 489-497.

[15] J. Bergstra and Y. Bengio, "Random Search for Hyper-parameter Optimization," J. Mach. Learn. Res., vol. 13, pp. 281-305, 2012.

[16] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in Proc. 26th Annu. Conf. Neural Information Processing Systems (NIPS 2012), Lake Tahoe, NV, USA, 2012, pp. 2960-2968.

[17] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kegl, "Algorithms for Hyper-parameter Optimization," in Proc. 25th Annu. Conf. Neural Information Processing Systems (NIPS 2011), Granada, Spain, 2011, pp. 2546-2554.

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD '19), Anchorage, AK, USA, 2019, pp. 2623-2631. DOI: 10.1145/3292500.3330701

[19] C. X. Ling and V. S. Sheng, "Cost-Sensitive Learning and the Class Imbalance Problem," in Encyclopedia of Machine Learning, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer, 2011, pp. 231-235.

[20] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," in Proc. 14th Int. Joint Conf. Artif. Intell. (IJCAI), Montreal, QC, Canada, 1995, pp. 1137-1143.

[21] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in Proc. 23rd Int. Conf. Mach. Learn. (ICML), Pittsburgh, PA, USA, 2006, pp. 233-240. DOI: 10.1145/1143844.1143874

[22] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," PLoS ONE, vol. 10, no. 3, e0118432, 2015. DOI: 10.1371/journal.pone.0118432

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321-357, 2002. DOI: 10.1613/jair.953

[24] S. M. H. Mirsadeghi, H. Bahsi, R. Vaarandi, and W. Inoubli, "Learning From Few Cyber-Attacks: Addressing the Class Imbalance Problem in Machine Learning-Based Intrusion Detection in Software-Defined Networking," IEEE Access, vol. 11, pp. 140428-140442, 2023. DOI: 10.1109/ACCESS.2023.3341755

[25] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, 2016, pp. 785-794. DOI: 10.1145/2939672.2939785

[26] L. Breiman, "Random Forests," Mach. Learn., vol. 45, no. 1, pp. 5-32, 2001. DOI: 10.1023/A:1010933404324

[27] M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," Procedia Comput. Sci., vol. 89, pp. 117-123, 2016. DOI: 10.1016/j.procs.2016.06.016

[28] X. Ma and W. Shi, "AESMOTE: Adversarial Reinforcement Learning With SMOTE for Anomaly Detection," IEEE Trans. Netw. Sci. Eng., vol. 8, no. 2, pp. 943-956, 2021. DOI: 10.1109/TNSE.2020.3004312

[29] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Abumallouh, "Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic," IEEE Sensors Lett., vol. 3, no. 1, pp. 1-4, 2019. DOI: 10.1109/LSENS.2018.2879990

[30] B. W. Matthews, "Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme," Biochim. Biophys. Acta, vol. 405, no. 2, pp. 442-451, 1975. DOI: 10.1016/0005-2795(75)90109-9

[31] S. Sunardi and S. Suyahman, "Analisis Komparasi Prediksi Serangan DDoS Menggunakan Machine Learning," in Proc. Informatics Collaborations and Dissemination Meeting, vol. 1, no. 1, pp. 84-91, 2025.

[32] F. Fitriyadi, S. Suyahman, and S. A. Asif, "Evaluation of Campus Wi-Fi Network Security Against Man-in-the-Middle Attacks Using AI-Based Intrusion Detection Systems," J. Electrical, Computer and Informatics, 2025.