

An Imbalance-Aware Machine Learning Approach for Improving Minority-Class Cyberattack Detection

Disha Patel¹, Dr. Vipulkumar Dabhi², Dr. S. A. Nagalpara³

¹Research Scholar, Department of Computer Science, Gokul Global University, Gujarat.
dishapatel1693@gmail.com

²Assistant Professor, Dept. of Computer Science, Gokul Global University, Gujarat.
dean.tech@gokuluniversity.com

³Assistant Professor, Maktabah Jafariyah BCA College, HNGU, Patan, Gujrat, India.
sirajali.bca@maktabahjafariyah.org
0009-0003-3274-5365

Abstract—Intrusion detection models trained on skewed network data favor frequent traffic and miss rare, high-impact attacks. We study imbalance-aware learning on NSL-KDD, where U2R attacks constitute 0.04% of the training records, and propose a hybrid pipeline: random-forest feature selection, capped under-sampling with Borderline-SMOTE and Tomek-link cleaning, and a cost-sensitive LightGBM and Random Forest ensemble. We compare it with five standard classifiers and seven imbalance-aware alternatives under a single leakage-free protocol (5-fold cross-validation, a 30% hold-out, and the official KDDTest+ file). In cross-validation, the pipeline achieves the highest macro-F1 (0.951 ± 0.010) and a U2R recall of 0.775 ± 0.083 , against 0.936 and 0.582 for XGBoost; its gain over class-weighted LightGBM (0.944) is small and not significant. On KDDTest+, which contains unseen attack types, all methods detect few rare attacks (proposed: R2L recall 0.12, U2R 0.13). Imbalance handling clearly helps over unweighted learners, but the full hybrid adds only a modest benefit over simple cost-sensitive learning.

Keywords — intrusion detection, imbalanced classification, minority-class detection, SMOTE, cost-sensitive learning, NSL-KDD

I. INTRODUCTION

Machine learning is widely used to detect network intrusions that signature-based tools miss [1], but intrusion data are highly imbalanced: benign flows and denial-of-service attacks dominate, while privilege-escalation and remote credential attacks are rare. A classifier tuned for overall accuracy can achieve very high scores yet still miss most rare attacks [2]. In NSL-KDD [3], U2R has only 52 of 125,973 training records (imbalance ratio 1,295:1), and R2L is 0.79%, yet these are the events with the highest consequence. Remedies include data-level (resampling), algorithm-level (cost-sensitive learning), or hybrid approaches [2], [4], but they are rarely compared under a single leakage-free protocol with per-class reporting, an external test set, and significance testing. The contributions are: (1) a hybrid pipeline combining feature selection, three-step resampling, and a cost-sensitive ensemble; (2) a like-for-like benchmark of 13 methods; (3) minority-aware evaluation with statistical tests; and (4) an ablation and an honest account of where the design does not help.

II. RELATED WORK

SMOTE interpolates between minority neighbors [5]; Borderline-SMOTE restricts this to samples near the decision boundary [6], and ADASYN generates more samples where learning is hardest [7]. Tomek-link removal and undersampling clean or shrink the majority side, and all are available in imbalanced-learn [8]. Cost-sensitive learning reweights errors by class [4] and is built into Random Forest [9], XGBoost [10], and LightGBM [11]; the focal loss [12] reshapes the loss for the same purpose. Recent IDS studies pair resampling with deep learning or feature selection on NSL-KDD [13], [14] and report high aggregate scores, but accuracy masks rare-class failures; their splits differ from ours, and we do not compare numbers directly. Datasets such as UNSW-NB15 [15] and CIC-IDS2017 [16] are more recent, though CIC-IDS2017 needs label corrections [17]. We evaluate with macro-averaged metrics, MCC [18], PR-AUC [19], and McNemar's test [20].

III. DATASET

We use NSL-KDD [3] (KDDTrain+: 125,973 records; KDDTest+: 22,544), with 41 features (3 categorical). Neither file contains duplicates, missing, or infinite values; the constant column num_outbound_cmds is dropped, leaving 121 features after one-hot encoding. The 23 (train) and 38 (test) attack types are merged into Normal, DoS, Probe, R2L, and U2R. KDDTest+ is deliberately harder: R2L rises to 12.8%, and 17 attack types appear only in the test file, so it measures generalization to unseen variants (Table I).

TABLE I. NSL-KDD class distribution and imbalance ratio (against the largest class).

Class	KDDTrain+	%	70 % train	30 % hold-out	KDDTest+	%	Imbalance ratio
Normal	67,343	53.46	47,140	20,203	9,711	43.08	1.0:1
DoS	45,927	36.46	32,149	13,778	7,460	33.09	1.5:1
Probe	11,656	9.25	8,159	3,497	2,421	10.74	5.8:1
R2L	995	0.79	697	298	2,885	12.80	67.7:1
U2R	52	0.04	36	16	67	0.30	1295.1:1

IV. PROPOSED METHODOLOGY

The pipeline (Fig. 1) is applied only to training data. Preprocessing: one-hot encoding and standardization are fitted on the training set within each cross-validation fold. Feature selection: a class-balanced Random Forest (50 trees, depth 12) ranks the 121 encoded features, and the top 40 are retained. Hybrid resampling: (1) random undersampling

caps each class at 15,000 samples; (2) Borderline-SMOTE ($k = 5$, $m = 10$) raises R2L and U2R to 3,000 samples each; (3) Tomek-link removal cleans the boundary. This yields 44,107 training records, compared with 235,700 for full-balance SMOTE. Cost-sensitive ensemble: LightGBM with class weights equal to the square root of the inverse class ratio and a balanced-subsample Random Forest (100 trees) are combined by averaging their class probabilities. All targets and settings were fixed in advance; nothing was tuned on test data.

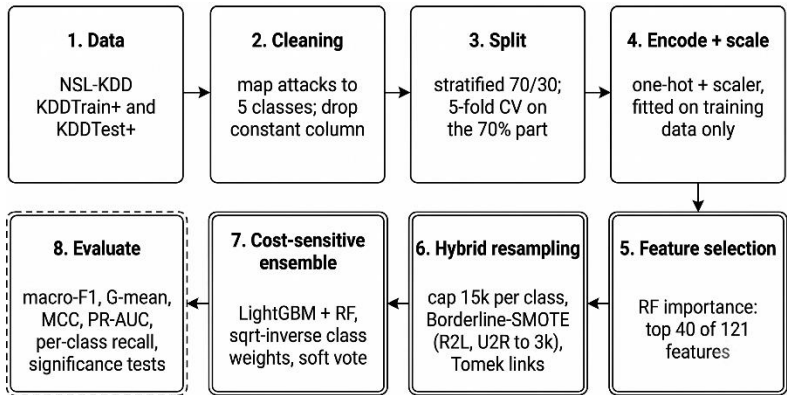


Fig. 1. Proposed pipeline (blue: common preprocessing; violet: proposed components).

V. EXPERIMENTAL SETUP

KDDTrain+ is split once, stratified, into 70% training (88,181) and 30% hold-out (37,792) with seed 42. 5-fold stratified cross-validation runs on the 70% part, with preprocessing and resampling refitted per fold. Every method is also evaluated on the full KDDTest+ file. Baselines are Logistic Regression, Decision Tree, Random Forest, XGBoost, and LightGBM (default settings, 100 trees where applicable). Imbalance-aware comparators are class-weighted Random Forest and LightGBM, and LightGBM after random undersampling, SMOTE, Borderline-SMOTE, ADASYN, or SMOTE-Tomek (imbalanced-learn defaults, which equalize all classes). A focal-loss variant was not implemented. Metrics are accuracy, macro-F1 (primary), G-mean, MCC, macro ROC-AUC, macro PR-AUC, and per-class recall and F1. We compare the proposed method with the best baseline and best comparator using the exact McNemar test, bootstrap intervals for the macro-F1 difference (1,000 resamples), and paired t-tests across folds. Software: Python 3.13.16, scikit-learn 1.9.1, imbalanced-learn 0.14.2, XGBoost 3.4.1, LightGBM 4.7.0.

VI. RESULTS AND DISCUSSION

A. Overall and rare-class performance

Accuracy saturates (at least 0.9985 across five methods) and cannot distinguish among the strong models, so we focus on macro-F1 and rare-class measures (Table II, Fig. 2). In cross-validation, the proposed pipeline has the highest macro-F1 (0.951 ± 0.010), macro-recall, and G-mean (0.943 ± 0.019), ahead of class-weighted LightGBM (0.944 ± 0.012) and XGBoost (0.936 ± 0.006); the margin over class-weighted LightGBM is smaller than one standard deviation. Default LightGBM is unstable (0.753 ± 0.053) because it often misses U2R entirely, and class weighting lifts it to 0.944. SMOTE, Borderline-SMOTE, ADASYN, and SMOTE-Tomek give 0.917, 0.925, 0.927, and 0.922, so full-balance oversampling did not beat simple class weighting here. Random under-sampling fails: it finds U2R (0.914 ± 0.128) but keeps only 36 records per class, and its macro-precision collapses to 0.607 (macro-F1 0.637).

TABLE II. 5-fold cross-validation results (mean $\pm$ SD; best per column in bold, training time in seconds per fold).

Method	Macro-F1	G-mean	MCC	PR-AUC	R2L recall	U2R recall	Time (s)
LR	0.845 $\pm$ 0.035	0.763 $\pm$ 0.086	0.9800 $\pm$ 0.0010	0.862 $\pm$ 0.047	0.769 $\pm$ 0.055	0.389 $\pm$ 0.186	10.8
DT	0.902 $\pm$ 0.026	0.872 $\pm$ 0.056	0.9956 $\pm$ 0.0010	0.846 $\pm$ 0.028	0.953 $\pm$ 0.036	0.554 $\pm$ 0.161	1.7
RF	0.912 $\pm$ 0.025	0.853 $\pm$ 0.053	0.9974 $\pm$ 0.0008	0.949 $\pm$ 0.031	0.941 $\pm$ 0.020	0.496 $\pm$ 0.139	4.8
XGB	0.936 $\pm$ 0.006	0.888 $\pm$ 0.010	0.9977 $\pm$ 0.0006	0.954 $\pm$ 0.017	0.956 $\pm$ 0.024	0.582 $\pm$ 0.024	6.3
LGBM	0.753 $\pm$ 0.053	0.608 $\pm$ 0.351	0.9478 $\pm$ 0.0383	0.710 $\pm$ 0.062	0.786 $\pm$ 0.116	0.304 $\pm$ 0.270	4.5
RF + class weights	0.901 $\pm$ 0.022	0.819 $\pm$ 0.052	0.9972 $\pm$ 0.0009	0.954 $\pm$ 0.036	0.924 $\pm$ 0.027	0.414 $\pm$ 0.128	5.2
LGBM + class weights	0.944 $\pm$ 0.012	0.930 $\pm$ 0.008	0.9981 $\pm$ 0.0006	0.967 $\pm$ 0.025	0.966 $\pm$ 0.026	0.721 $\pm$ 0.016	7.1
LGBM + RUS	0.637 $\pm$ 0.015	0.908 $\pm$ 0.023	0.8457 $\pm$ 0.0334	0.820 $\pm$ 0.029	0.891 $\pm$ 0.011	0.914 $\pm$ 0.128	0.3
LGBM + SMOTE	0.917 $\pm$ 0.033	0.882 $\pm$ 0.067	0.9980 $\pm$ 0.0007	0.938 $\pm$ 0.027	0.966 $\pm$ 0.026	0.579 $\pm$ 0.183	15.0
LGBM + Borderline-SMOTE	0.925 $\pm$ 0.020	0.897 $\pm$ 0.039	0.9981 $\pm$ 0.0005	0.951 $\pm$ 0.033	0.966 $\pm$ 0.024	0.611 $\pm$ 0.119	25.9
LGBM + ADASYN	0.927 $\pm$ 0.011	0.901 $\pm$ 0.018	0.9981 $\pm$ 0.0004	0.947 $\pm$ 0.024	0.976 $\pm$ 0.021	0.611 $\pm$ 0.062	29.0
LGBM + SMOTE-Tomek	0.922 $\pm$ 0.032	0.890 $\pm$ 0.070	0.9980 $\pm$ 0.0007	0.936 $\pm$ 0.024	0.966 $\pm$ 0.026	0.607 $\pm$ 0.192	208.4
Proposed	0.951 $\pm$ 0.010	0.943 $\pm$ 0.019	0.9979 $\pm$ 0.0008	0.957 $\pm$ 0.026	0.968 $\pm$ 0.027	0.775 $\pm$ 0.083	14.5

U2R is the discriminating class. The proposed pipeline achieves a cross-validated U2R recall of 0.775 ± 0.083 and an F1 of 0.787 ± 0.055 , compared with 0.721 ± 0.016 and 0.744 ± 0.053 for class-weighted LightGBM and 0.582 ± 0.024 and 0.713 ± 0.026 for XGBoost; under-sampling yields higher recall but an F1 of only 0.036 ± 0.010 . Most methods detect R2L well (0.968 ± 0.027 for the proposed method). Against XGBoost, paired t-tests over folds favor the proposed method for macro-F1 (mean difference +0.015, $p = 0.016$), R2L recall ($p = 0.036$), and U2R recall (+0.193, $p = 0.004$); against class-weighted LightGBM, none of the three differences is significant ($p = 0.095$, 0.647, and 0.179). These p-values use five folds, are uncorrected for multiple tests, and are indicative only. On the hold-out, the McNemar test over R2L and U2R samples favors the proposed method over XGBoost (6 versus 0 discordant records, $p = 0.031$), but the bootstrap interval for the macro-F1 difference includes zero for both comparators on both test sets.

B. Hold-out and external test

TABLE III. Internal hold-out (HO) and external KDDTest+ (Ext) results; best per column in bold.

Method	HO macro-F1	HO U2R recall	Ext accuracy	Ext macro-F1	Ext R2L recall	Ext U2R recall	Ext PR-AUC
LR	0.830	0.312	0.757	0.566	0.012	0.239	0.712
DT	0.918	0.625	0.753	0.518	0.092	0.239	0.433
RF	0.889	0.312	0.745	0.486	0.034	0.015	0.755
XGB	0.925	0.625	0.775	0.547	0.075	0.104	0.762
LGBM	0.781	0.312	0.721	0.471	0.047	0.104	0.442
RF + class weights	0.927	0.500	0.738	0.477	0.002	0.030	0.733
LGBM + class weights	0.960	0.812	0.765	0.569	0.058	0.209	0.746
LGBM + RUS	0.625	0.938	0.757	0.569	0.218	0.657	0.633
LGBM + SMOTE	0.955	0.812	0.772	0.568	0.094	0.149	0.714
LGBM + Borderline-SMOTE	0.957	0.750	0.769	0.575	0.080	0.209	0.716
LGBM + ADASYN	0.945	0.812	0.785	0.595	0.120	0.179	0.708
LGBM + SMOTE-Tomek	0.945	0.812	0.777	0.578	0.107	0.149	0.709
Proposed	0.954	0.812	0.762	0.563	0.118	0.134	0.776

On the hold-out (Table III), macro-F1 is 0.954 for the proposed method, 0.925 for XGBoost, and 0.960 for class-weighted LightGBM; with only 16 U2R records, these differences correspond to one or two records. On KDDTest+, every method degrades: accuracy is 0.721–0.785 and macro-F1 is 0.471–0.595. The proposed method achieves 0.563; the best value is 0.595 (LGBM + ADASYN), so it is not the top method under distribution shift. Its R2L and U2R recalls (0.118 and 0.134) are higher than XGBoost's for R2L (0.075) and U2R (0.104), but still low: most R2L and U2R records are labeled Normal (Fig. 3). Restricted to rare-class samples, the proposed method is correct more often than XGBoost (154 versus 26 discordant samples) and then class-weighted LightGBM (185 versus 16), whereas over all samples it is correct less often (279 versus 589 and 354 versus 433): it flags more rare attacks at the price of more errors on Normal and DoS traffic. No tested method solves rare-class detection under distribution shift, because resampling and weighting cannot supply knowledge of attack types absent from training (Fig. 5).

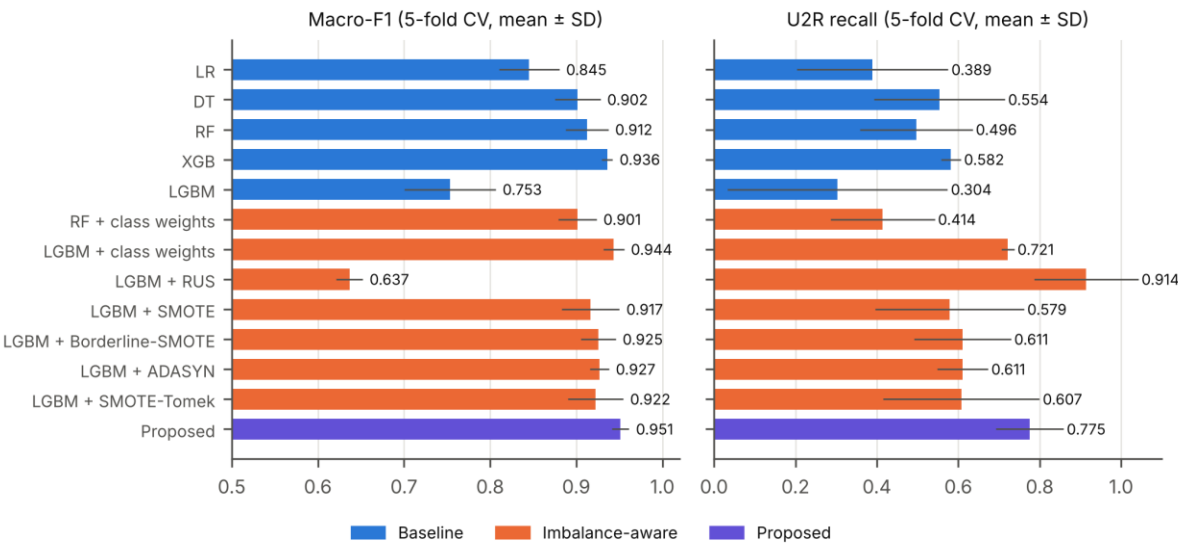


Fig. 2. Cross-validated macro-F1 and U2R recall of all methods (5-fold, mean ± SD).

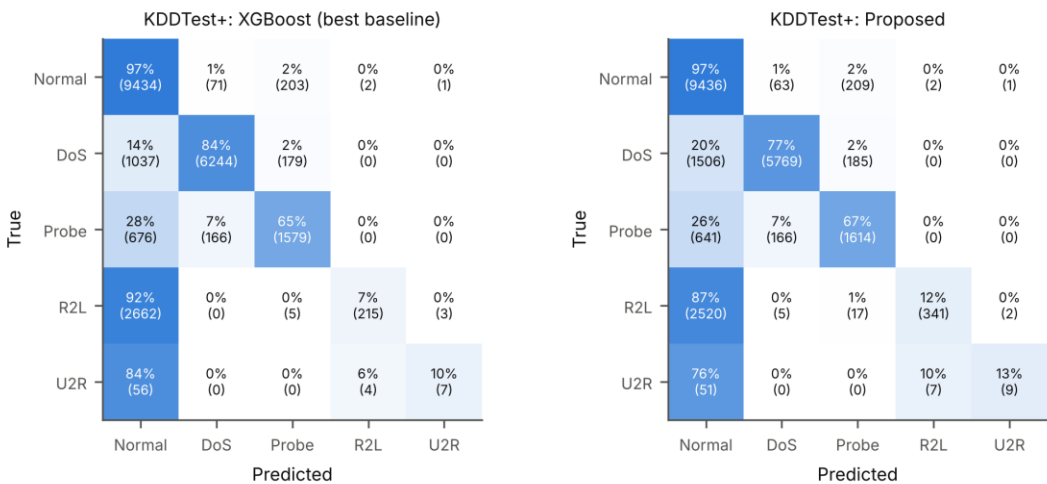


Fig. 3. Row-normalised confusion matrices (with counts) on KDDTest+ for XGBoost and the proposed method.

C. Ablation and cost

TABLE IV. Ablation of the proposed pipeline (5-fold CV: mean ± SD; HO: hold-out; Ext: KDDTest+).

Variant	CV macro-F1	CV G-mean	CV R2L recall	CV U2R recall	HO macro-F1	Ext macro-F1
Full pipeline	0.9513 ± 0.0100	0.943 ± 0.019	0.968 ± 0.027	0.775 ± 0.083	0.954	0.563
– feature selection	0.9286 ± 0.0147	0.916 ± 0.015	0.973 ± 0.024	0.668 ± 0.066	0.936	0.587

Variant	CV macro-F1	CV G-mean	CV R2L recall	CV U2R recall	HO macro-F1	Ext macro-F1
– resampling	0.9362 ± 0.0271	0.904 ± 0.040	0.971 ± 0.023	0.636 ± 0.135	0.971	0.550
– Tomek cleaning	0.9390 ± 0.0219	0.929 ± 0.026	0.968 ± 0.025	0.721 ± 0.102	0.965	0.558
– cost-sensitive weights	0.9480 ± 0.0151	0.936 ± 0.020	0.970 ± 0.024	0.746 ± 0.072	0.943	0.541
– ensemble (LightGBM only)	0.9475 ± 0.0185	0.935 ± 0.031	0.971 ± 0.027	0.746 ± 0.124	0.949	0.562

Removing any component lowers cross-validated macro-F1 from 0.9513 (Table IV), with the largest drop from feature selection (0.9286, $p = 0.039$), followed by resampling (0.9362) and Tomek cleaning (0.9390). Only removing feature selection yields a paired-test p -value below 0.05; the other drops fall within fold-to-fold noise (SD about 0.010). The hold-out and KDDTest+ do not reproduce this ordering (the best hold-out macro-F1 belongs to the variant without resampling), which is expected given 16 U2R records and a shifted test file. We therefore conclude that each component probably helps a little in-distribution, but we cannot rank their individual contributions with confidence. Training takes 14.5 s per fold for the proposed pipeline, 14 times faster than SMOTE-Tomek (208 s) and slower than class-weighted LightGBM (7.1 s).

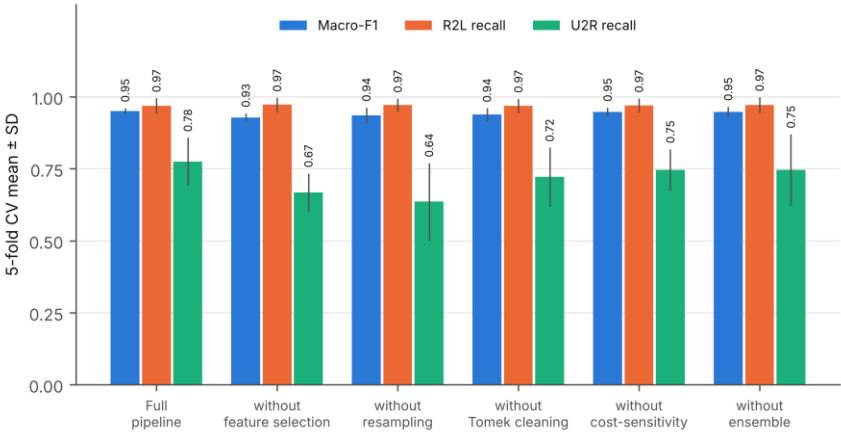


Fig. 4. Ablation: cross-validated macro-F1, R2L recall and U2R recall.

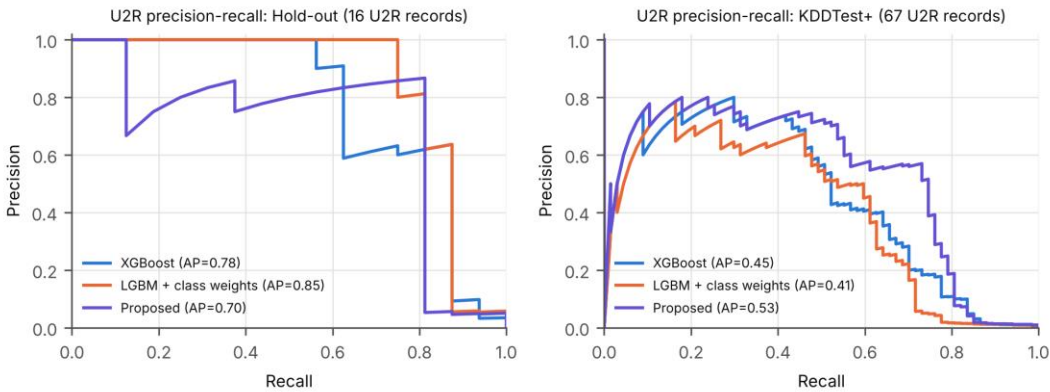


Fig. 5. U2R precision-recall curves (AP = average precision).

VII. LIMITATIONS

NSL-KDD is derived from 1998-era simulated traffic and should be complemented by UNSW-NB15 [15] or a corrected CIC-IDS2017 [16], [17]. U2R has only 16 hold-out and about 7 validation records per fold, so its estimates are coarse. KDDTest+ contains 17 unseen attack types, so low recall for rare classes reflects both distribution shift and imbalance. SMOTE interpolates one-hot columns; categorical-aware generators may behave differently. No hyperparameters were tuned; results use a single seed; resampling comparators share a single learner; and published NSL-KDD figures [13], [14] are not directly comparable to ours.

VIII. CONCLUSION

A hybrid imbalance-aware pipeline achieved the best cross-validated macro-F1 (0.951 ± 0.010) and U2R recall (0.775 ± 0.083) on NSL-KDD, significantly above the best unweighted baseline, but only marginally and not significantly above the class-weighted LightGBM (0.944). Under the KDDTest+ distribution shift, all methods missed most R2L and U2R attacks. Cost-sensitive learning is therefore a cheap and effective first step against imbalance; hybrid resampling adds a modest in-distribution benefit for the rarest class; and robustness to novel attacks requires other tools. Future work: validation on UNSW-NB15 and corrected CIC-IDS2017, nested-CV tuning, SMOTE-NC, a focal-loss variant, multiple seeds, threshold calibration, and open-set detection.

REFERENCES

[1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, 2016, doi: 10.1109/COMST.2015.2494502.

[2] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[3] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl. (CISDA)*, 2009.

[4] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2001, pp. 973–978.

[5] 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.

- [6] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing (ICIC 2005)*, Lecture Notes in Computer Science, vol. 3644. Berlin, Germany: Springer, 2005, pp. 878–887, doi: 10.1007/11538059_91.
- [7] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2008, pp. 1322–1328, doi: 10.1109/IJCNN.2008.4633969.
- [8] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017.
- [9] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, doi: 10.1145/2939672.2939785.
- [11] 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 *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017, pp. 3146–3154.
- [12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.
- [13] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *J. Supercomput.*, vol. 79, pp. 10611–10644, 2023, doi: 10.1007/s11227-023-05073-x.
- [14] Y. G. Damteaw and H. Chen, "SMMO-CoFS: Synthetic multi-minority oversampling with collaborative feature selection for network intrusion detection system," *Int. J. Comput. Intell. Syst.*, vol. 16, Art. no. 12, 2023, doi: 10.1007/s44196-022-00171-9.
- [15] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, doi: 10.1109/MilCIS.2015.7348942.
- [16] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP)*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.
- [17] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: The CICIDS2017 case study," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2021, pp. 7–12, doi: 10.1109/SPW53761.2021.00009.
- [18] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, Art. no. 6, 2020, doi: 10.1186/s12864-019-6413-7.
- [19] 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, Art. no. e0118432, 2015, doi: 10.1371/journal.pone.0118432.
- [20] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, pp. 1895–1923, 1998, doi: 10.1162/089976698300017197.