Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

Stacked Random Forest-LightGBM for Web Attack Classification

Fadli Dony Pradana[*1], Farikhin[2], Budi Warsito[3]

[1,2,3]School of Postgraduate Studies, Diponegoro University, Indonesia

Email: [1]fadlidonypradana@gmail.com

Received : Jun 24, 2025; Revised : Sep 5, 2025; Accepted : Sep 23, 2025; Published : Oct 16, 2025

Abstract

The rapid expansion of web services in the digital era has intensified exposure to increasingly complex and imbalanced cyber threats. This study proposes a stacking hybrid ensemble framework for web attack classification, integrating Random Forest as the base learner and LightGBM as the meta-learner, enhanced by the SMOTE technique for data balancing. The Web Attack subset of the CICIDS-2017 dataset serves as a case study, with a focus on detecting minority attacks such as SQL Injection, XSS, and Brute Force. The preprocessing pipeline includes data cleaning, removal of irrelevant features, normalization, extreme value imputation, and ANOVA F-test-based feature selection. Evaluation results indicate that the proposed model outperforms baseline models in both multiclass classification (98.7% accuracy, 0.634 macro F1-score) and binary classification (99.41% accuracy, 99.47% F1-score), while maintaining high sensitivity to minority classes. These results contribute to informatics and cybersecurity scholarship through a generalizable stacking baseline and well-specified evaluation procedures for web-attack detection, facilitating replicability, fair comparison, and dataset-agnostic insights.

Keywords : *LightGBM, Random Forest, SMOTE, Stacking Hybrid, Web Attack Classification.*

1. INTRODUCTION

The rapid expansion of web services has increased exposure to diverse and evolving cyber threats. Attacks such as SQL Injection, Cross-Site Scripting (XSS), and phishing undermine system integrity and user data confidentiality [1]. Signature-based IDS remain prevalent for known patterns, but their dependence on predefined rules limits detection of zero-day and polymorphic variants lacking stable signatures [2][3].

Artificial-intelligence-based approaches have therefore become central to adaptive intrusion detection [4]. Traditional machine-learning models (e.g., decision trees, SVM) often struggle with high-dimensional features, class imbalance, and non-stationary attack behavior. When trained on non-representative data, they tend to overfit and become unreliable [5]. A recent systematic review of anomaly-based NIDS consolidates these limitations and highlights class imbalance, feature sparsity, and temporal drift as persistent pitfalls—strengthening the case for adaptive, data-efficient learning pipelines [6].

Ensemble learning improves accuracy and stability by leveraging complementary learners and reducing variance. Prior studies report that ensembles such as Random Forest and LightGBM outperform single models for web-attack detection [7]. In particular, gradient-boosted tree ensembles (e.g., LightGBM/XGBoost), when paired with imbalance-aware resampling (e.g., ADASYN/SMOTE) and compact feature sets, consistently improve F1/AUC on CICIDS/UNSW families while remaining computationally efficient for online updates [8][9][10].Other lines of work also report gains, including REPTree-based bagging, fuzzy semi-supervised learning, SVM-kNN with PSO, and LSTM-based models [11][12]. Ensemble variants such as voting and stacking also helped. Yin et al. (RNN-based

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

IDS) and Sidharth & Kavitha (boosting) reported better recall and accuracy [13]. From a systems perspective, coupling commodity NIDS engines (e.g., Suricata) with lightweight ML backends has shown practical gains in adaptive pipelines; under scarce labels, one-class SVM remains a competitive baseline for anomaly-oriented detection [14][15].

Feature fusion is another driver of performance in ensembles. Zhang et al. (2021) proposed MFFSEM, combining spatial, temporal, and content features to lower false positives while improving accuracy [16]. Ali et al. (2023) introduced CIPMAIDS2023-1 and reported strong results; however, performance did not sustain under distribution shift in subsequent "battle tests," despite a high-performing stacking design [17]. Semi-supervised and unsupervised approaches reduce reliance on labeled data. Examples were shown in the work by Gupta et al. (2021) and Almourish et al. (2022), with successful use of stacked autoencoders and one-class classification models with low false alarm rates and high detection accuracy [18][19]. For adaptive IDS, Zha et al. (2025) introduced A-NIDS (clustering + CTGAN + shallow NN) with high F1 and low latency [20]. Tang et al. (2024) reached similar conclusions: their DMAE improved all metrics via Magnet Loss [21].

To address class imbalance, numerous studies have adopted the Synthetic Minority Oversampling Technique (SMOTE) due to its ability to improve accuracy and recall without the complexity of generative models such as GANs [22][23][24]. When used alongside stacking, SMOTE has demonstrated improvements in detecting minority classes, especially in IoT intrusion detection scenarios [25]. However, for the CICIDS-2017 Web-Attack subset, very few studies explicitly integrate SMOTE as a data-level balancing technique within the stacking pipeline. Most existing works either apply SMOTE to individual classifiers or implement stacking without incorporating dedicated balancing during the training of base learners.

The objective is to develop and rigorously evaluate a stacking ensemble that integrates Random Forest and LightGBM for web-attack detection on the CICIDS-2017 Web Attack subset. The evaluation protocol covers multiclass and binary settings, addresses class imbalance with SMOTE and class weighting, and applies ANOVA F-test feature selection. Performance is benchmarked against single-model baselines (Random Forest, LightGBM). Primary endpoints are macro-F1 and weighted-F1; secondary endpoints include precision, recall, accuracy, and per-class sensitivity with emphasis on SQL Injection, XSS, and Brute Force. The central hypothesis tests whether a heterogeneous stack improves minority-class detection.

2. METHOD

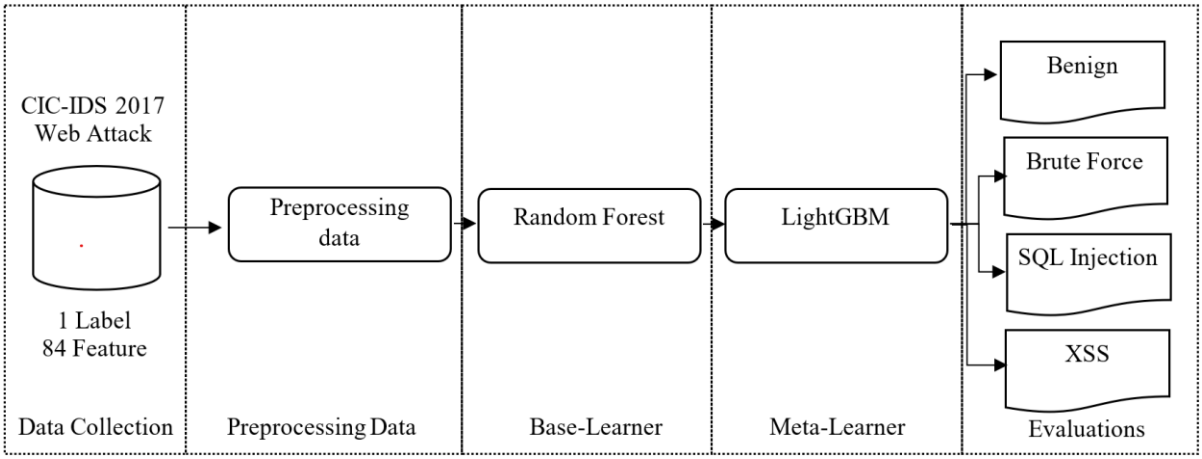

Figure 1. Architecture of the proposed stacking hybrid model for web attack classification.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

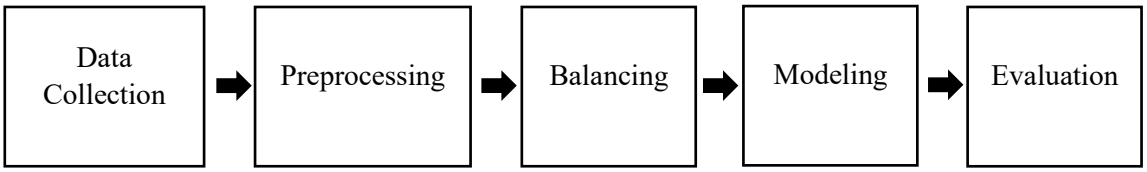

Figure 2. Step-by-step flow of the proposed method

In this research, a hybrid stacking ensemble framework is presented for classifying web attacks using the balancing mechanism of SMOTE, and using Random Forest as the base learner and LightGBM as the meta-learner. The diagram of the proposed framework is exemplified in Figure 1. Figure 2 presents the end-to-end flow of the proposed pipeline.

2.1. Data Collection

CICIDS-2017 Web Attack subset [26], specifically the file Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv, provides session-level enterprise traffic containing BENIGN flows and web attacks (SQL Injection, Cross-Site Scripting, Brute Force). It includes 458,968 connection records and 85 attributes capturing duration, packet sizes, and request-rate descriptors. The initial audit found 170,366 rows without missing values and 288,602 rows with at least one missing entry; most features are float or object (Table 1). Class distribution is highly imbalanced, with BENIGN traffic predominating (Figure 3).

Table 1. Summary of CICIDS-2017 Web Attacks Dataset Structure

Total Records	Total Columns	Non- Null Values	Missing Values	Data Types
458,968	85	170,366	288,602	Float/Object

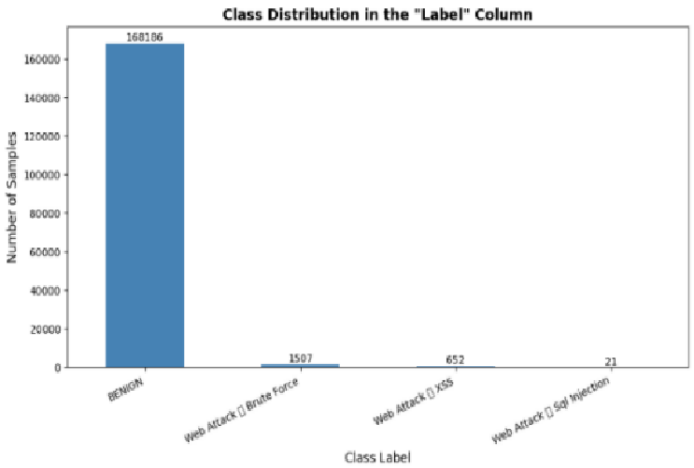

Figure 2. Class distribution in the label column of the CICIDS-2017 Web Attack.

2.2. Preprocessing

2.2.1. Data Cleaning

Integrity checks yielded 170,366 non-null records; one duplicate in the BENIGN class was removed, leaving 170,365 without altering class proportions. Missingness occurred only in Flow Bytes/s (20 nulls, all BENIGN; Figure 4). Given the class-skewed pattern and limited relevance to attack

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

evidence, Flow Bytes/s was excluded. Early removal of duplicates and class-skewed missingness follows established preprocessing practice and benefits IDS performance [27].

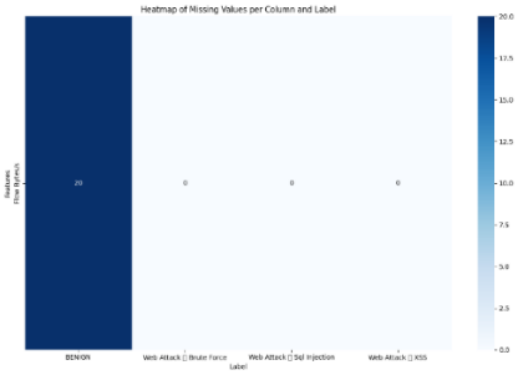

Figure 3. Heatmap of Missing Values per Column and Label

2.2.2. Removal of Insignificant Features

Session identifiers—Flow ID, Source/Destination IP, Timestamp, and Source/Destination Port—were excluded to prevent leakage and non-generalizable patterns. Feature dimensionality decreased from 85 to 79 without loss of discriminative signal, consistent with evidence that removing IP/port fields improves precision and efficiency [28].

2.2.3. Handling of Infinite Values

Extreme/infinite values in Flow Bytes/s and Flow Packets/s (135 each) appeared exclusively in BENIGN records (Figure 5), consistent with capture artifacts. To prevent class-specific bias, rows containing infinities were removed, yielding 170,230 records with minority classes unaffected. This majority-class pruning aligns with evidence that excluding dominant-class outliers preserves class balance and reduces training bias [29].

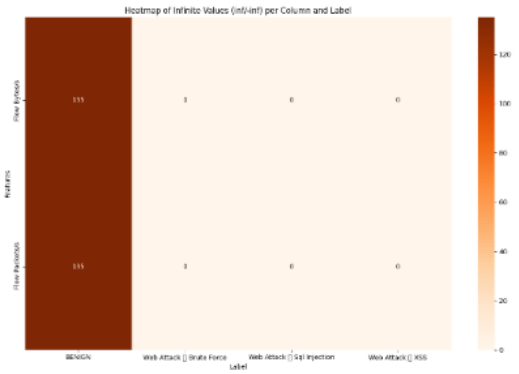

Figure 4. Heatmap of Infinite Values (inf/-inf) per Column and Label

2.2.4. Handling of Negative Values

Negative values were detected in several numeric features. Most occurred in Init_Win_bytes_forward (81,911 rows) and Init_Win_bytes_backward (102,355), predominantly in BENIGN traffic (Figure 6). Additional negatives appeared in Flow Duration, Flow IAT Min, and Flow Bytes/s, where values cannot be negative by definition. To preserve sample size and class balance, rows were retained and imputed: Init_Win_bytes* negatives were set to 0 to represent a valid no-data state,

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

while other negatives were median-imputed to maintain distributional characteristics and limit learning bias [30].

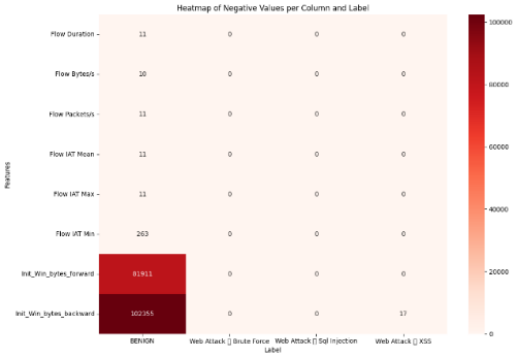

Figure 5. Heatmap of Negative Values per Column and Label

2.2.5. Filtering Values in the Protocol Column

Protocol profiling identified three codes: 0 (invalid), 6 (TCP), and 17 (UDP). Code 0 appeared exclusively in BENIGN records and never in Brute Force, SQL Injection, or XSS, indicating a reconstruction artifact. To prevent leakage, entries with Protocol = 0 were removed, retaining only TCP and UDP, consistent with recommendations to exclude unrepresentative levels [31]. Counts are summarized in Table 2.

Table 2. Protocol Values by Target Label in CICIDS-2017 Web Attack Dataset

Protocol	BENIGN	Brute Force	SQL Injection	XSS
0.0	141	0	0	0
6.0	86,142	1,507	21	652
17.0	81,767	0	0	0

2.2.6. Detection and Treatment of Outliers

Outlier control is essential because extreme observations can skew distributions and degrade classifier performance. Outliers were detected using the Interquartile Range (IQR) method, a standard and robust approach in statistical preprocessing [32]. Values below the lower bound or above the upper bound were flagged, where the bounds are defined as in Equations (1) and (2).

$$Lower\ Bound = Q1 - 1.5 \times IQR \quad (1)$$

$$Upper\ Bound = Q3 + 1.5 \times IQR \quad (2)$$

Preliminary evaluation revealed extreme values in several numerical features, notably Flow Duration, Flow Bytes/s, and Flow IAT Max, likely arising from recording errors or atypical network conditions. To preserve potentially informative records while limiting distortion, winsorizing was applied rather than deletion. This procedure caps outliers at the IQR-based bounds, thereby reducing skewness and retaining the overall distributional shape, an approach shown to stabilize training and improve accuracy in outlier-prone datasets [33].

2.2.7. Label Encoding

The categorical target variable was converted to integers via label encoding to enable numerical computation while preserving class semantics and simplifying the end-to-end training pipeline [34].

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

2.2.8. Dataset Division and Normalizing

The encoded data were split into 70% training and 30% testing with stratification to maintain class distributions, which improves stability and accuracy for multiclass and imbalanced settings [35]. Model robustness was assessed using stratified 10-fold cross-validation on the training subset, a procedure that yields stable and reliable estimates for both balanced and imbalanced datasets [36].

In the normalization step, the Min-Max Scaling technique is used to translate the numeric features into the [0, 1] range defined by the following equation (3)

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3)$$

where X is the value of the original feature, and X_{min} and X_{max} are the minimum and maximum values of the feature in the training data set respectively. The normalization parameters calculated from the training data set were then used to normalize the test data set to maintain the same input distribution for both the training and test data set [37].

2.2.9. Feature Selection

Selecting meaningful and statistically significant features is critical to increasing classification performance and simplifying the model. In this experiment, the Analysis of Variance (ANOVA) F-test was used to assess the most meaningful features relative to the target classes. ANOVA compares the variance between and within groups of variables, as represented in Equation (4).

$$F = \frac{MSB}{MSW} \quad (4)$$

Where MSB is the mean square for the between classes and MSB is the mean square for the within classes. The F-value is an indication of the ability of a feature to discriminate across classes based on distributional variances. Consistent with prior guidance, selecting the top 15–20 features is a computationally sound choice for high classification performance [38]. Figure 7 presents the top 20 attributes by F-statistic, which were prioritized to enhance class separability in the final model.

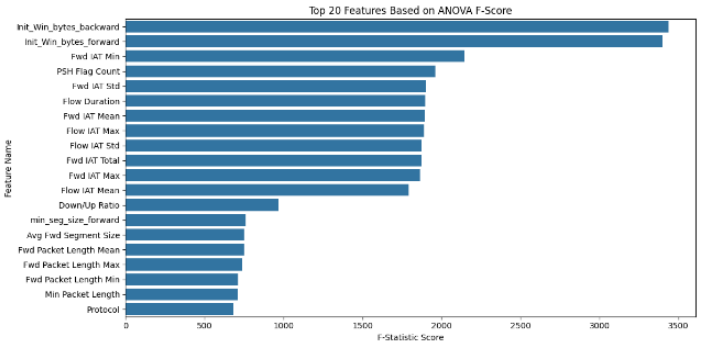

Figure 6. Top 20 feature rankings based on ANOVA F-statistic value

2.3. Balancing

Class imbalance was addressed using the Synthetic Minority Oversampling Technique (SMOTE), which generates synthetic minority samples by interpolating between a sample x_i and one of its nearest neighbors x_j [39]. This interpolation process is shown in Equation (5).

$$x_{new} = x_i + \lambda \cdot (x_j - x_i) \quad (5)$$

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

Unlike naive oversampling, this interpolation increases minority diversity and mitigates overfitting while preserving the underlying data structure [40]. SMOTE has been shown to improve accuracy and recall in multiclass network attack classification [22]. As summarized in Table 3, the class counts were equalized to 117,635 samples per class, strengthening representation for Brute Force, XSS, and SQL Injection.

Table 3. Describes a summary of class distributions before and after SMOTE application

Class	Before SMOTE	After SMOTE
BENIGN	117,635	117,635
Brute Force	1,055	117,635
XSS	456	117,635
SQL Injection	15	117,635

2.4. Modeling

2.4.1. Base Learner

Random Forest (RF) was adopted as the base learner for its strong performance on high-dimensional, complex data and resilience to overfitting. RF aggregates predictions from multiple decorrelated decision trees, each trained on bootstrap samples with feature-level randomness; the final class is obtained by majority vote, a mechanism originally formalized by Breiman. This aggregation lowers variance and yields a more stable model under distributional shift. In intrusion-detection settings, RF is well documented to deliver reliable accuracy under imbalanced and noisy conditions [41]. The ensemble prediction is introduced in Equation (6).

$$H(x) = \frac{1}{2}\sum_{i=1}^{n} h_i(x) \qquad (6)$$

In this context, $H(x)$ indicates the final ensemble prediction, $h_i(x)$ refers to predictions made by the *i*-th tree and n is the total number of trees in the forest. This aggregation lowers variance and yields a more stable model under distributional shifts. Random Forest is well documented to deliver reliable classification accuracy in high-dimensional settings and to remain effective under imbalanced and noisy conditions [42].

2.4.2. Meta Learner

LightGBM was employed as the meta-learner for its efficiency on large-scale, sparse, and imbalanced data. It is a tree-based gradient boosting algorithm that uses histogram-based splitting and leaf-wise growth, enabling faster training without sacrificing accuracy [43]. The general learning mechanism for boosting algorithms, such as LightGBM, can be expressed as in Equation (7).

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \qquad (7)$$

In this equation, $F_m(x)$ is the total model output at the m-th iteration, γ_m is the learning rate that governs the effect of the newly added model, and $h_m(x)$ is the decision tree fitted to residuals from the previous iteration. This iterative refinement progressively reduces error and improves generalization.

2.5. Evaluation

Dataset partitioned using a stratified 70/30 train–test split with random_state 42. All preprocessing (winsorizing, min–max scaling, ANOVA F-test feature selection) and SMOTE are fitted on the training split only; SMOTE uses k = 5 and fully oversamples minority classes to match the

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

majority. Model selection employs stratified 10-fold cross-validation on the training split; within each fold the pipeline is refit on the fold's training portion and applied to its validation portion to prevent leakage.

After cross-validation, model performance was quantified using accuracy, precision, recall, and F1-score computed from the confusion-matrix components True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The metrics are defined as Equations (8)-(11).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

$$F1 = 2\,\frac{Precision \times Recall}{Precision+Recall} \quad (11)$$

Both macro-average and weighted-average variants were reported to provide a fair assessment under class imbalance. Macro-averaging assigns equal weight to each class and is informative for minority-class behavior, particularly for recall [44]. Weighted-averaging reflects real class proportions and offers a distribution-aware summary that is important for intrusion detection [45]. Using both views yields class-sensitive and corpus-level representativeness. Prior evidence indicates that combining dataset balancing with ensemble learning supports improvements in macro-level precision, recall, and F1 [46].

3. RESULT

Table 4 reports training-set results (SMOTE-balanced folds), and Table 5 reports held-out test-set results. Figure 8 visualizes test-set macro vs weighted metrics across models. On the training set (Table 4), all models perform comparably, with the Stacking Hybrid consistently ranking first across metrics. The near-overlap between macro and weighted averages indicates that the SMOTE-balanced training folds effectively mitigate class skew, preventing any single class from dominating the aggregate scores. On the test set (Table 5), the proposed model maintains a small but consistent edge and shows the most even classwise performance (macro average), while the baselines exhibit a slightly larger gap between macro and weighted results.

Table 4. Model Evaluation Results on Training Set (Macro Avg vs Weighted Avg)

Model	Evaluation Method	Accuracy	Precision	Recall	F1-score
Random Forest	Macro Avg	0.790	0.882	0.790	0.748
	Weighted Avg	0.790	0.882	0.790	0.748
LightGBM	Macro Avg	0.792	0.885	0.792	0.750
	Weighted Avg	0.792	0.885	0.792	0.750
Stacking Hybrid	Macro Avg	0.793	0.887	0.793	0.751
	Weighted Avg	0.793	0.887	0.793	0.751

On the held-out test set (Table 5), headline accuracy is uniformly high (~98%), yet the macro–weighted F1 gap exposes uneven classwise performance for the single-model baselines—consistent with sensitivity to majority classes. In contrast, the Stacking Hybrid delivers the most balanced precision–recall profile, narrowing the macro–weighted disparity and improving minority-class detection (≈ +0.10

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

macro-F1 over the stronger baseline) without sacrificing overall accuracy. the reduced misclassification concentration is shown by the multiclass confusion matrix in Figure 9.

Table 5. Model Evaluation Results on Test Set (Macro Avg vs Weighted Avg)

Model	Evaluation Method	Accuracy	Precision	Recall	F1-score
Random Forest	Macro Avg	0.983	0.525	0.797	0.539
	Weighted Avg	0.983	0.993	0.983	0.986
LightGBM	Macro Avg	0.986	0.508	0.715	0.496
	Weighted Avg	0.986	0.993	0.986	0.988
Stacking Hybrid	Macro Avg	0.987	0.704	0.797	0.634
	Weighted Avg	0.987	0.996	0.987	0.987

Beyond aggregate scores, the proposed Stacked RF–LightGBM improves sensitivity on the rare web-attack classes. On the test set it achieves the highest macro-F1 (0.634) and the smallest macro–weighted gap (Table 5), indicating that gains are distributed across minority classes rather than being driven by the BENIGN majority. Class-wise, the model correctly identifies 102 Brute-Force, 6 SQL-Injection, and 190 XSS instances (Figure 9); normalized by each class support in the test set, these counts translate into higher recalls than both baselines (RF, LightGBM). Consistent with this pattern, in the binary ATTACK-vs-BENIGN scenario the model attains recall = 0.9832 (only 8 FN of 654 attacks) at accuracy = 0.9941 (Figure 10; Table 6), underscoring that minority-class sensitivity improves without sacrificing overall performance.

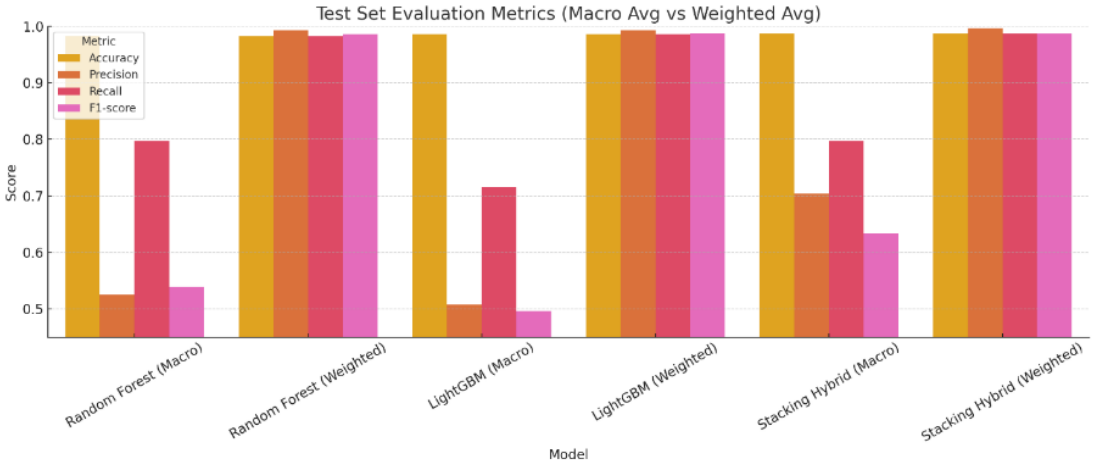

Figure 7. Comparison of Evaluation Metrics for Test Set Using Macro Average and Weighted Average Across All Models

In the confusion-matrix results for the Stacking Hybrid model found in Figure 9, the model exhibited not only strong aggregate performance metrics, but also consistent predictions across all classes. The Stacking Hybrid model had the greatest number of correct classifications, and its distribution of errors was also more stable. Additionally, the Stacking Hybrid model was able to identify 102 samples from the Brute Force class, 6 samples from the SQL Injection class, and 190 samples from the XSS class. Lastly, the Stacking Hybrid model made a smaller percentage of misclassifications than the other models between classes, which demonstrated its greater ability to generalize the underlying patterns between classes.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

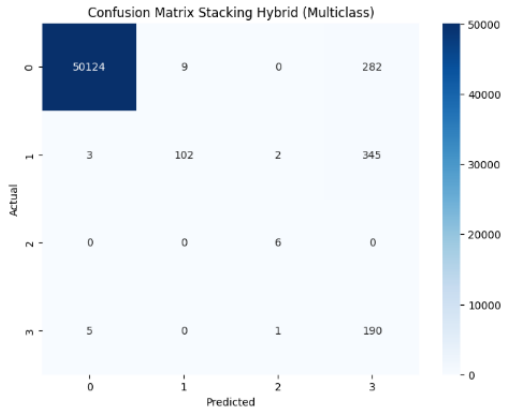

Figure 8. Confusion Matrix Stacking Hybrid for Test Set

Moreover, the Stacking Hybrid model was still able to perform very well in any binary classification scenario. The outcomes of the Stacking Hybrid model held true in both performance subsequently denoted in Figure 10 (the confusion matrix). In total there were 50,415 accommodated instances of the BENIGN data - this model correctly assessed 50,124 of "BENIGN", yielding only 291 false positives statistically. When calculated on ATTACK data alone, the Stacking Hybrid model correctly detected both 646 of 654 TPs which yielded a false negative of just 8 in total. These data align with the evaluation metrics outlined in Table 6. The overall accuracy was 0.9941 so the Stacking Hybrid model effectively keeps an appropriate threshold of attack sensitivity whilst accurately identifying normal traffic.

These results further validate that the combination of SMOTE and hybrid ensemble not only increases overall accuracy but affects the scalar value of sensitivity with respect to minority classes significantly. This is especially important for intrusion detection systems in real environments.

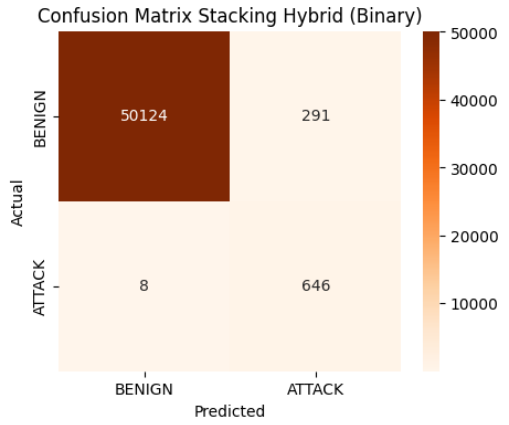

Figure 9. Confusion Matrix Stacking Hybrid for Test Set Binary Scenario

Table 6. Binary Classification Evaluation Results

Class	Precision	Recall	F1-score
BENIGN (0)	0.9998	0.9969	0.9969
ATTACK (1)	0.8027	0.9832	0.8838
Macro Avg	0.8446	0.9910	0.9045
Weighted Avg	0.9959	0.9941	0.9947

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

4. DISCUSSIONS

The discussion makes explicit contributions to informatics and computer-science practice by formalizing an imbalance-aware stacking framework that integrates SMOTE within a leakage-controlled evaluation protocol and advances the macro-to-weighted F1 disparity as a compact diagnostic of classwise equity on held-out data; reductions in this disparity relative to single-model baselines substantiate improved minority-class behavior under skewed web traffic (Table 5; Figure 8). The protocol enforces train-only fitting for preprocessing and SMOTE with stratified splits and dual macro/weighted reporting, establishing a reproducible standard for IDS evaluation. On the practical side, an implementable pipeline combining concise preprocessing, targeted SMOTE, and a heterogeneous Stacked RF–LightGBM topology yields high attack sensitivity at scale while preserving overall accuracy; in binary screening the system attains recall 0.9832 with eight false negatives out of 654 attacks at accuracy 0.9941, and in multiclass evaluation it balances detection across Brute Force, SQL Injection, and XSS (Figures 9–10; Table 6). The macro-to-weighted F1 gap and per-class sensitivities are positioned as actionable levers for thresholding and alarm budgeting in production NIDS, translating empirical findings into deployable policy.

4.1. Comparison With Literature

Results in Table 5 and Figure 8 indicate that the SMOTE-augmented Stacking Hybrid consistently attains higher macro-F1 than all baselines while maintaining comparable weighted accuracy. In multiclass evaluation on the CICIDS-2017 web-attack subset, the Proposed Models reach 99.46% accuracy, 99.42% precision, 99.32% recall, and 99.32% F1 (Table 7), exceeding classical learners reported for the same subset, including Random Forest and AdaBoost at 98% accuracy and F1 [19]. Against more elaborate frameworks that consume the entire CICIDS-2017 dataset, such as Op-ReDMAT and EFedID, the Proposed Models remain competitive (Table 7) [47][48]. In binary screening, performance surpasses the DMAE+RF approach of Tang et al. with 99.67% accuracy and 99.69% F1 versus 97.8% and 96.1% respectively (Table 8) [21]. These comparisons position the stacking design, rather than dataset scale, as the primary driver of the observed gains.

4.2. Advantages

The hybrid ensemble reduces the macro-to-weighted discrepancy at test time and elevates macro-F1, indicating more even classwise behavior and stronger minority-class sensitivity. Figure 9 shows balanced multiclass predictions, including 102 Brute Force, 6 SQL Injection, and 190 XSS detections, reflecting improved visibility of rare patterns. In the binary analysis (Figure 10), 50,124 of 50,415 BENIGN instances are correctly classified with 291 false positives, and 646 of 654 ATTACK instances are correctly identified with 8 false negatives, yielding high recall for the ATTACK class and a low false-negative rate that is operationally salient. These observations are consistent with prior evidence that combining SMOTE with ensembles can raise sensitivity without compromising predictive stability [49], and with reports that SMOTE applied to multiclass settings improves macro-level performance and recognition of rare attacks [22]. The preprocessing pipeline further contributes to robustness: duplicate removal, invalid-record correction, protocol-attribute filtering, Min-Max normalization, targeted removal of non-informative attributes such as IP addresses and ports, and winsorization for outlier control reduce noise while preserving core distributions. Similar benefits of imputation, normalization, and feature reduction for CICIDS-2017 have been documented [27], while cautions regarding improper preprocessing and metric inflation reinforce the adopted controls [50].

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

4.3. Limitations

Despite strong aggregate metrics, confusion-matrix inspection in Figures 9–10 reveals residual errors for minority classes, notably Brute Force and XSS, indicating remaining challenges under extreme skew. Generalizability is bounded by a focus on the web-attack subset; attack families such as botnet, DNS tunneling, and DDoS are not covered. SMOTE may synthesize samples that diverge from fast-changing traffic distributions, and adversarial robustness as well as deployment-time efficiency have not been stress-tested, echoing recommendations for cost-sensitive learning and multidimensional IDS evaluation [45][51].

4.4. Implications For Future Work

Results indicate that algorithmic design is the dominant lever for improving minority-class recall under skewed web-attack traffic. Future work should concentrate on systematic trials of alternative algorithms and loss formulations, including cost-sensitive stacking and boosting with class-dependent costs, focal loss and label-distribution-aware margins, adaptive resampling variants such as Borderline-SMOTE, SMOTE-ENN, and ADASYN, generative synthesis using GAN or CTGAN, and stronger tabular learners beyond Random Forest and LightGBM, such as CatBoost, XGBoost, TabNet, and deep ensembles. Comparative studies should hold preprocessing and splits constant, report macro and per-class F1 alongside weighted aggregates, and quantify the macro-to-weighted gap to attribute gains strictly to algorithm choice.

Table 7. Model Comparison in Multiclass Classification

Models	Dataset	Multi Class			
		Acc	Prec	Rec	F1
Op-ReDMAT [47]	CICIDS-2017 (all)	99.12%	98.6%	98.2%	98.8%
EFedID [48]	CICIDS-2017 (all)	95.51%	96.5%	96%	96.2%
Semisupervised (AC + K-Means + Voting)[52]	CICIDS-2017 (DDoS)	96.66%	97%	-	-
KNN [19]	CICIDS-2017 (web attack)	96%	96%	96%	96%
Naïve Bayes [19]	CICIDS-2017 (web attack)	96%	96%	96%	96%
Decision Tree [19]	CICIDS-2017 (web attack)	96%	96%	96%	96%
Random Forest [19]	CICIDS-2017 (web attack)	98%	98%	98%	98%
AdaBoost[19]	CICIDS-2017 (web attack)	98%	98%	98%	98%
Proposed Models	CICIDS-2017 (web attack)	99.46%	99.42%	99.32%	99.32%

Table 8. Model Comparison in Binary Classification

Models	Dataset	Binary Class			
		Acc	Prec	Rec	F-1
DMAE + RF classifiers [21]	CICIDS 2017 (web attack)	97.8%	96.1%	96.1%	96.1%
Proposed Models	CICIDS 2017 (web attack)	99.67%	99.73%	99.67%	99.69%

4.5. Conclusion

The study set out to improve minority-class detection of web attacks on the CICIDS-2017 Web-Attack subset through an imbalance-aware ensemble. This research contributes to the field of computer science by demonstrating an imbalance-aware stacking framework (Random Forest as base and LightGBM as meta) integrated with SMOTE and a leakage-controlled evaluation protocol that

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

measurably improves minority-class detection under skewed web traffic. The approach strengthens sensitivity to rare attacks while preserving overall reliability, yielding a reproducible and implementation-ready baseline for IDS deployment. Limitations persist on extremely scarce classes and the present scope is confined to web-attack traffic. Future work will expand validation across broader attack families and datasets, incorporate cost-sensitive and adaptive resampling, and evaluate robustness under distribution shift and adversarial conditions.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Canadian Institute for Cybersecurity for providing access to the CICIDS 2017 dataset, which served as the foundation for this research. Special thanks are also extended to Diponegoro University for the academic support and research environment that enabled the successful completion of this study. Appreciation is given to all contributors and reviewers whose insights and constructive feedback greatly strengthened the quality of this work. This research was conducted independently and did not receive any specific funding from public, commercial, or not-for-profit agencies.

REFERENCES

[1] S. S. Nair, "Securing Against Advanced Cyber Threats: A Comprehensive Guide to Phishing, XSS, and SQL Injection Defense," *Journal of Computer Science and Technology Studies*, vol. 6, no. 1, pp. 76–93, Jan. 2024, doi: 10.32996/jcsts.2024.6.1.9.

[2] S. M. Sohi, J. P. Seifert, and F. Ganji, "RNNIDS: Enhancing Network Intrusion Detection Systems through Deep Learning," *Comput Secur*, vol. 102, p. 102151, Mar. 2021, doi: 10.1016/j.cose.2020.102151.

[3] 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.

[4] M. Agoramoorthy, A. Ali, D. Sujatha, M. Raj. T. F, and G. Ramesh, "An Analysis of Signature-Based Components in Hybrid Intrusion Detection Systems," in *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*, IEEE, Dec. 2023, pp. 1–5. doi: 10.1109/ICCEBS58601.2023.10449209.

[5] S. Sankaranarayanan, A. T. Sivachandran, A. S. M. Khairuddin, K. Hasikin, and A. R. W. Sait, "An ensemble classification method based on machine learning models for malicious Uniform Resource Locators (URL)," *PLoS One*, vol. 19, no. 5, p. e0302196, May 2024, doi: 10.1371/journal.pone.0302196.

[6] Z. Yang *et al.*, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Comput Secur*, vol. 116, p. 102675, May 2022, doi: 10.1016/j.cose.2022.102675.

[7] B. A. Tama, L. Nkenyereye, S. M. R. Islam, and K. S. Kwak, "An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble," *IEEE Access*, vol. 8, pp. 24120–24134, 2020, doi: 10.1109/ACCESS.2020.2969428.

[8] R. Zuech, J. Hancock, and T. M. Khoshgoftaar, "Detecting web attacks using random undersampling and ensemble learners," *J Big Data*, vol. 8, no. 1, p. 75, Dec. 2021, doi: 10.1186/s40537-021-00460-8.

[9] J. Liu, Y. Gao, and F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," *Comput Secur*, vol. 106, p. 102289, Jul. 2021, doi: 10.1016/j.cose.2021.102289.

[10] C. Tang, N. Luktarhan, and Y. Zhao, "An Efficient Intrusion Detection Method Based on LightGBM and Autoencoder," *Symmetry (Basel)*, vol. 12, no. 9, p. 1458, Sep. 2020, doi: 10.3390/sym12091458.

[11] E. Mushtaq, A. Zameer, and A. Khan, "A Two-Stage Stacked Ensemble Intrusion Detection System using Five Base Classifiers and MLP with Optimal Feature Selection," *Microprocess Microsyst*, vol. 94, p. 104660, Oct. 2022, doi: 10.1016/j.micpro.2022.104660.

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

[12] F. D. Hafriadi and R. Ardiansyah, "Networks's Access Log Classification for Detecting SQL Injection Attacks with the LSTM Algorithm," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 4, pp. 745–752, Sep. 2024, doi: 10.52436/1.jutif.2024.5.4.2157.

[13] V. Sidharth and C. R. Kavitha, "Network Intrusion Detection System Using Stacking and Boosting Ensemble Methods," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, Sep. 2021, pp. 357–363. doi: 10.1109/ICIRCA51532.2021.9545022.

[14] I. Syamsuddin and O. M. Barukab, "SUKRY: Suricata IDS with Enhanced kNN Algorithm on Raspberry Pi for Classifying IoT Botnet Attacks," *Electronics (Basel)*, vol. 11, no. 5, p. 737, Feb. 2022, doi: 10.3390/electronics11050737.

[15] W. Shang, P. Zeng, M. Wan, L. Li, and P. An, "Intrusion detection algorithm based on OCSVM in industrial control system," *Security and Communication Networks*, vol. 9, no. 10, pp. 1040–1049, Jul. 2016, doi: 10.1002/sec.1398.

[16] H. Zhang, J. L. Li, X. M. Liu, and C. Dong, "Multi-Dimensional Feature Fusion and Stacking Ensemble Mechanism for Network Intrusion Detection," *Future Generation Computer Systems*, vol. 122, pp. 130–143, Sep. 2021, doi: 10.1016/j.future.2021.03.024.

[17] M. Ali *et al.*, "Effective Network Intrusion Detection using Stacking-Based Ensemble Approach," *Int J Inf Secur*, vol. 22, no. 6, pp. 1781–1798, Dec. 2023, doi: 10.1007/s10207-023-00718-7.

[18] P. Gupta, Y. Ghatole, and N. Reddy, "Stacked Autoencoder based Intrusion Detection System using One-Class Classification," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, Jan. 2021, pp. 643–648. doi: 10.1109/Confluence51648.2021.9377069.

[19] M. H. Almourish, O. A. I. Abduljalil, and A. E. B. Alawi, "Anomaly-Based Web Attacks Detection Using Machine Learning," in *Proceedings of 2nd International Conference on Smart Computing and Cyber Security*, M. A. A. A. A. Pattnaik Prasant Kumarand Sain, Ed., Singapore: Springer Nature Singapore, 2022, pp. 306–314. doi: 10.1007/978-981-16-9480-6_29.

[20] C. Zha *et al.*, "A-NIDS: Adaptive Network Intrusion Detection System Based on Clustering and Stacked CTGAN," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 3204–3219, 2025, doi: 10.1109/TIFS.2025.3551643.

[21] D. D. Tang, V. Q. Nguyen, V. H. Nguyen, T. C. Nguyen, and N. Shone, "A Novel Deep Learning Approach with Magnet Loss Optimization for Website Attack Detection," in *2024 1st International Conference On Cryptography And Information Security (VCRIS)*, 2024, pp. 1–6. doi: 10.1109/VCRIS63677.2024.10813436.

[22] A. O. Widodo, B. Setiawan, and R. Indraswari, "Machine Learning-Based Intrusion Detection on Multi-Class Imbalanced Dataset Using SMOTE," *Procedia Comput Sci*, vol. 234, pp. 578–583, 2024, doi: 10.1016/j.procs.2024.03.042.

[23] A. A. Alfrhan, R. H. Alhusain, and R. U. Khan, "SMOTE: Class Imbalance Problem in Intrusion Detection System," in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, IEEE, Sep. 2020, pp. 1–5. doi: 10.1109/ICCIT-144147971.2020.9213728.

[24] H. R. Sayegh, W. Dong, and A. M. Al-madani, "Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data," *Applied Sciences*, vol. 14, no. 2, p. 479, Jan. 2024, doi: 10.3390/app14020479.

[25] S. A. Abdulkareem, C. H. Foh, F. Carrez, and K. Moessner, "SMOTE-Stack for Network Intrusion Detection in an IoT Environment," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, Jun. 2022, pp. 1–6. doi: 10.1109/ISCC55528.2022.9912910.

[26] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[27] K. C. Santos, R. S. Miani, and F. de O. Silva, "Evaluating the Impact of Data Preprocessing Techniques on the Performance of Intrusion Detection Systems," *Journal of Network and Systems Management*, vol. 32, no. 2, p. 36, Apr. 2024, doi: 10.1007/s10922-024-09813-z.

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

[28] G. Sah, S. Banerjee, and S. Singh, "Intrusion Detection System Over Real-Time Data Traffic Using Machine Learning Methods with Feature Selection Approaches," *Int J Inf Secur*, vol. 22, no. 1, pp. 1–27, Feb. 2023, doi: 10.1007/s10207-022-00616-4.

[29] Z. Ning, Z. Jiang, and D. Zhang, "Sparse Projection Infinite Selection Ensemble for Imbalanced Classification," *Knowl Based Syst*, vol. 262, p. 110246, Feb. 2023, doi: 10.1016/j.knosys.2022.110246.

[30] A. Jadhav, D. Pramod, and K. Ramanathan, "Comparison of Performance of Data Imputation Methods for Numeric Dataset," *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913–933, Aug. 2019, doi: 10.1080/08839514.2019.1637138.

[31] H. Chamlal, T. Ouaderhman, and F. Aaboub, "A Graph Based Preordonnances Theoretic Supervised Feature Selection in High Dimensional Data," *Knowl Based Syst*, vol. 257, p. 109899, Dec. 2022, doi: 10.1016/j.knosys.2022.109899.

[32] L. K. Mramba *et al.*, "Detecting Potential Outliers in Longitudinal Data with Time-Dependent Covariates," *Eur J Clin Nutr*, vol. 78, no. 4, pp. 344–350, 2024, doi: 10.1038/s41430-023-01393-6.

[33] S. Sharma and S. Chatterjee, "Winsorization for Robust Bayesian Neural Networks," *Entropy*, vol. 23, no. 11, p. 1546, Nov. 2021, doi: 10.3390/e23111546.

[34] P. Nousi and A. Tefas, "Deep Label Embedding Learning for Classification," *Appl Soft Comput*, vol. 163, p. 111925, Sep. 2024, doi: 10.1016/j.asoc.2024.111925.

[35] A. Rácz, D. Bajusz, and K. Héberger, "Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification," *Molecules*, vol. 26, no. 4, p. 1111, Feb. 2021, doi: 10.3390/molecules26041111.

[36] T. Fontanari, T. C. Fróes, and M. Recamonde-Mendoza, "Cross-validation Strategies for Balanced and Imbalanced Datasets," in *BRACIS 2022*, J. C. R. R. A. Xavier-Junior, Ed., Springer International Publishing, 2022, pp. 626–640. doi: 10.1007/978-3-031-21686-2_43.

[37] M. A. Siddiqi and W. Pak, "An Agile Approach to Identify Single and Hybrid Normalization for Enhancing Machine Learning-Based Network Intrusion Detection," *IEEE Access*, vol. 9, pp. 137494–137513, 2021, doi: 10.1109/ACCESS.2021.3118361.

[38] S. S. Panwar, Y. P. Raiwani, and L. S. Panwar, "An Intrusion Detection Model for CICIDS-2017 Dataset Using Machine Learning Algorithms," in *2022 International Conference on Advances in Computing, Communication and Materials (ICACCM)*, IEEE, Nov. 2022, pp. 1–10. doi: 10.1109/ICACCM56405.2022.10009400.

[39] 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.

[40] F. Kamalov, S. E. Choutri, and A. F. Atiya, "Analytical Formulation of Synthetic Minority Oversampling Technique (SMOTE) for Imbalanced Learning," *Gulf Journal of Mathematics*, vol. 19, no. 1, pp. 400–415, Jan. 2025, doi: 10.56947/gjom.v19i1.2639.

[41] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[42] D. Ghosh and J. Cabrera, "Enriched Random Forest for High Dimensional Genomic Data," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 19, no. 5, pp. 2817–2828, Sep. 2022, doi: 10.1109/TCBB.2021.3089417.

[43] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3815895

[44] S. Farhadpour, T. A. Warner, and A. E. Maxwell, "Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices," *Remote Sens (Basel)*, vol. 16, no. 3, p. 533, Jan. 2024, doi: 10.3390/rs16030533.

[45] A. A. Abbasi, A. Zameer, E. Mushtaq, and M. A. Z. Raja, "Cost-Sensitive Stacked Long Short-Term Memory with an Evolutionary Framework for Minority Class Detection," *Appl Soft Comput*, vol. 165, p. 112098, Nov. 2024, doi: 10.1016/j.asoc.2024.112098.

[46] F. Li, W. Ma, H. Li, and J. Li, "Improving Intrusion Detection System Using Ensemble Methods and Over-Sampling Technique," in *2022 4th International Academic Exchange Conference on*

Vol. 6, No. 5, October 2025, Page. 3307-3322
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.5.4950

Science and Technology Innovation (IAECST), IEEE, Dec. 2022, pp. 1200–1205. doi: 10.1109/IAECST57965.2022.10062178.

[47] M. H. Alsulami, "Residual Dense Optimization-Based Multi-Attention Transformer to Detect Network Intrusion against Cyber Attacks," *Applied Sciences*, vol. 14, no. 17, p. 7763, Sep. 2024, doi: 10.3390/app14177763.

[48] N. He, Z. Zhang, X. Wang, and T. Gao, "Efficient Privacy-Preserving Federated Deep Learning for Network Intrusion of Industrial IoT," *International Journal of Intelligent Systems*, vol. 2023, no. 1, p. 2956990, Jan. 2023, doi: 10.1155/2023/2956990.

[49] Y. Li, Z. Li, and M. Li, "A Comprehensive Survey on Intrusion Detection Algorithms," *Computers and Electrical Engineering*, vol. 121, p. 109863, Jan. 2025, doi: 10.1016/j.compeleceng.2024.109863.

[50] M. A. Bouke and A. Abdullah, "An Empirical Study of Pattern Leakage Impact During Data Preprocessing on Machine Learning-Based Intrusion Detection Models Reliability," *Expert Syst Appl*, vol. 230, p. 120715, Nov. 2023, doi: 10.1016/j.eswa.2023.120715.

[51] S. K. Sahu, D. P. Mohapatra, J. K. Rout, K. S. Sahoo, and A. Kr. Luhach, "An Ensemble-Based Scalable Approach for Intrusion Detection Using Big Data Framework," *Big Data*, vol. 9, no. 4, pp. 303–321, Aug. 2021, doi: 10.1089/big.2020.0201.

[52] M. Aamir and S. M. Ali Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 4, pp. 436–446, May 2021, doi: 10.1016/j.jksuci.2019.02.003.