Zahra: Bulletin Big data, Data Science and Artificial Intelligence
Vol. 3, No. 1, [June 2024], pp. [1–13]
E-ISSN 2830-7984

Research Article Open Access (CC–BY-SA)

A Hybrid Genetic Algorithm–Support Vector Machine Framework for Feature Selection and Multiclass Network Intrusion Detection

Tengku Mohd Diansyah[1*], H.R Merdu Wira Jasa[2], Ronal Watrianthos[3], Mazlina Abdul Majid[4]
[1] *Computer Science Program, Faculty of Engineering and Computer Science, Universitas Harapan, Medan, Indonesia*
[2] *Informatics, STIKOM Tunas Bangsa, Pematangsiantar, North Sumatra, Indonesia*
[3] *Department of Information Technology, Politeknik Negeri Padang, Padang, Indonesia*
[4] *Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, Pahang, Malaysia*

[1] *dian.10.22@gmail.com;* [2] *hamdanhrmwj@gmail.com;* [3] *ronalwatrianthos@pnp.ac.id;* [4] *mazlina@umpsa.edu.my*
** Corresponding author*

Abstract

Multiclass network intrusion detection must balance feature complexity against recognition of minority attacks. This study evaluates whether a hybrid genetic algorithm and support vector machine (GA-SVM) improve class-balanced intrusion classification through joint feature and parameter selection. The executed experiment uses 257,673 UNSW-NB15 records, including a training-side development split and 175,341 observations in the test-side file. Five configurations were compared: an all-feature SVM baseline, SVM parameter tuning, random joint feature-and-parameter search, GA feature selection alone, and the proposed joint GA-SVM. The original 42 input columns were reduced to 27 by the feature-selection branches. On the recorded test-side evaluation, the baseline achieved macro-F1 0.381508, whereas joint GA-SVM achieved 0.359207, a decrease of 0.022301. The hybrid lowered accuracy from 0.718098 to 0.710302 and balanced accuracy from 0.384733 to 0.368182. Its macro one-versus-rest ROC-AUC increased slightly, from 0.914785 to 0.916670, and measured inference time fell from 20.223 to 16.573 milliseconds per 1,000 predictions. The main objective did not improve despite a 35.71% reduction in original predictor columns. Notably, the record-based test audit identified 10,279 hashed rows matching training-side records and 74,301 repeated test rows, limiting independent generalization claims. At least one attack class had zero reported recall, and the imported optimizer source was unavailable for complete verification. The evidence supports feature compression but does not establish a stronger multiclass intrusion detector under the recorded protocol.

Keywords: intrusion detection; UNSW-NB15; genetic algorithm; support vector machine; multiclass feature selection

1. Introduction

Intrusion detection increasingly relies on multiclass traffic analysis to distinguish benign connections from heterogeneous attack categories [1]. Attack classes in UNSW-NB15 exhibit strong frequency imbalance and overlapping feature patterns [2]. A nominally strong overall accuracy can therefore coexist with weak detection of uncommon threats [3]. Feature selection seeks to remove uninformative flow variables while preserving class-specific discrimination [4]. Genetic search has been used to construct compact intrusion-detection representations [5]. Support Vector Machines provide a useful structured-data classifier whose performance depends on scaling and model configuration [6]. Reliable evaluation requires separating feature discovery from the observations used to assess the resulting classifier [7].

Recent studies have combined feature filtering, wrapper selection, and ensemble models to improve intrusion classification [8]. Evolutionary and hybrid searches can compress predictors but may also discard features important to rare attacks [9]. Imbalance-aware ensembles address a different source of error from the reduction of feature dimensionality [10]. SVM-based cost-sensitive learning illustrates why minority-class outcomes should be evaluated separately [11]. Empirical analyses of UNSW-NB15 show that transformations and selection choices affect measured detection performance [12]. Data overlap complicates the interpretation of apparently independent

classification scores [2]. These findings motivate a comparison that includes default SVM, isolated feature search, isolated hyperparameter search, and a joint intervention.

This work examines whether genetic feature-subset selection combined with SVM configuration search improves multiclass macro-F1 on UNSW-NB15. The experiment uses 257,673 labeled records with a fixed development split within the designated training partition. The comparison includes five fitted alternatives evaluated on one recorded final-test partition. Macro-F1 is the primary endpoint because it averages class-wise F1 rather than allowing frequent classes to dominate the result. Accuracy, balanced accuracy, class recall, false-alarm rate, feature count, and inference cost provide complementary evidence. A random joint feature-and-parameter search supplies a conventional optimization comparator [7]. The analysis tests for incremental improvement without assuming that a hybrid search is necessarily superior.

2. Method

2.1. Related Work

Wrapper-based feature selection has been combined with genetic optimization and SVM classification on UNSW-NB15 [5]. Joint genetic tuning and hybrid feature selection have also been examined for intrusion-detection models on other benchmark traffic [6]. A more recent GA-SVM hybrid reports dimensionality reduction on NSL-KDD rather than this exact multiclass partition [13]. Genetic optimization of temporal models has been tested on several IoT intrusion corpora [14]. Fusion of feature filters and genetic search offers another approach to constructing compact inputs [15]. Enhanced genetic metaheuristics have additionally been evaluated with neural classifiers on UNSW-NB15 [16]. These studies establish methodological relevance but do not substitute for a matched comparison of the particular SVM and data partition examined here.

Selection and extraction approaches alter the number and structure of traffic variables in different ways [4]. IGRF-RFE combines feature-ranking and recursive elimination for UNSW-NB15 intrusion detection [9]. Hybrid information-gain and sparse selection methods provide a distinct dimension-reduction strategy [17]. Minimum-feature studies indicate that representation size must be balanced against attack-class information [18]. Multivariate filter selection has also been integrated with stacked intrusion classifiers [19]. Feature aggregation and ensemble approaches can change predictive behavior without using an SVM kernel [20]. These directions motivate reporting both feature-count reduction and class-balanced detection instead of equating compactness with improved classification.

UNSW-NB15 experiments have documented the effects of train-test overlap and imbalanced classes [2]. Oversampling comparisons illustrate how minority detection can change when training distributions are adjusted [21]. Multi-class evaluations show that average scores conceal substantial differences among individual attack types [3]. Imbalance handling has been investigated with whale optimization and ensemble classifiers [22]. Anomaly-oriented GAN and nearest-neighbor balancing address another dimension of representation quality [23]. GA-based augmentation of intrusion examples similarly targets minority-class learning rather than purely feature selection [24]. The present experiment does not infer transferability from any of these distinct preprocessing or sampling protocols.

Intrusion detection models increasingly combine explainability, feature selection, and computational constraints [25]. Hybrid stacking approaches illustrate gains that depend on ensemble integration rather than SVM configuration [8]. Lightweight models demonstrate the importance of measuring latency after selecting a reduced input space [26]. Hierarchical detection systems provide alternative trade-offs between detection cost and complexity [27]. Forest-based attack-specific comparisons expose substantial class-dependent effects of resampling [28]. Graph-based detection models exploit information unavailable to independent-record classifiers [29]. Attention-based architectures offer further comparisons but use a different learning family [30]. These differences discourage cross-paper performance rankings without matched splits and labels.

Hyperparameter selection may overfit the development metric when many candidate combinations are examined [7]. Probability calibration is conceptually distinct from multiclass discrimination and should not be assumed from a classifier score [31]. Class-specific feature discrimination has been addressed through multiscale subspace models [32]. Hybrid metaheuristic feature selection is also assessed using neural classifiers and several benchmark datasets [33]. Large imbalanced intrusion datasets motivate learning pipelines that attend explicitly to feature compression [10]. Earlier dependable hybrid learning work illustrates the use of imbalance processing with feature selection [34]. A hybrid model for ransomware classification further demonstrates that optimization behavior depends on the

underlying threat and label task [35]. The unresolved question in this study is the incremental contribution of GA-SVM feature-and-parameter optimization under the executed evaluation.

Table 1 locates the present investigation among recent studies of intrusion classification and feature optimization. Genetic feature search is represented by wrapper-based experiments [5]. Combined parameter tuning is represented by GA-based intrusion detection research [6]. Benchmark imbalance and data overlap are emphasized by an UNSW-NB15 study [2]. Independent feature ranking and selection are represented by IGRF-RFE [9]. Hyperparameter evaluation is interpreted using established search methodology [7]. The remaining research question concerns the joint method's benefit relative to strong matched controls.

Table 1. Comparative synthesis of selected related studies

Author / year	Research contribution	Specific relevance / limitation
Maseno and Wang (2024)	GA wrapper and SVM-oriented intrusion feature selection	Different search and training protocol
Bakir and Ceviz (2024)	Joint tuning and hybrid feature selection	Different benchmark and model combinations
Zoghi and Serpen (2024)	UNSW-NB15 imbalance and overlap challenges	Protocol-specific data-quality dependence
Yin et al. (2023)	IGRF-RFE feature selection on UNSW-NB15	Uses an MLP rather than the present SVM
Bischl et al. (2023)	Fair hyperparameter-search design and evaluation	General methodology, not a task-specific score
Saputra et al. (2026)	Leakage-aware information and sparse feature selection	Different target subset and classifiers

The methods summarized in Table 1 target overlapping but nonidentical experimental questions. Some optimize feature subsets and others change the learning architecture. Genetic wrapper methods can reduce input dimensions without guaranteeing superior minority recall [5]. Conventional hyperparameter search is essential when judging incremental optimizer value [7]. Imbalanced-class evaluation should include class-level recall and false-alarm effects [2]. The present analysis therefore treats the five executed comparison branches as the relevant evidence for this dataset and split.

2.2. Dataset

The analysis uses the publicly distributed UNSW-NB15 multiclass traffic benchmark. The source data comprise designated training-side and test-side CSV files. The recorded training-side file contains 82,332 rows, divided into 65,865 fitting and 16,467 development observations. The recorded test-side file contains 175,341 rows, yielding 257,673 total records. The experiment identifies 42 original flow predictors and includes normal traffic alongside nine named attack families. Feature preprocessing and optimization followed the recorded training-development procedure, although the imported implementation routines could not be independently inspected. Table 2 reports the observed file counts, evaluation limits, and row-hash overlap diagnostics.

Table 2. Dataset characteristics, executed partitions, and overlap audit

Dataset attribute	Executed observation
Public data source	UNSW-NB15 via Kaggle; train-side and test-side CSV records
Endpoint and predictors	Multiclass attack category including normal traffic; 42 original columns
Recorded partition	Train 65,865 / development 16,467 / test-side 175,341; seed 42
Source data audit	257,673 total records; 10,279 hashed train/test matches; 74,301 repeats within test
Validation limitation	No verified independent flows or chronological groups; external modules not uploaded
Interpretation scope	Fixed-file multiclass comparison, not unseen-attack deployment validation

The source audit in Table 2 distinguishes the actual experiment from an idealized unseen-attack protocol. The 175,341-row test file was reserved for assessment after model selection. Row hashing nevertheless identified 10,279 test records matching training-side records and 74,301 repeated records within the test file. These matches complicate independence and can influence apparently strong class-level statistics. Independent reproduction of the hash audit remains constrained by the unavailability of the underlying implementation modules. Published analyses of UNSW-NB15 likewise highlight overlap and imbalance risks [2]. The present inference is therefore bounded by the recorded source files and hash diagnostics.

The evaluation sequence separates recorded source profiling, fitting, optimization, and final scoring. A development split is drawn only from the training-side input. The test-side input is designated for the final comparison. Five alternatives share the same class mapping and evaluation output. Search and feature selection do not use recorded final-test metrics for model selection. An overlap audit qualifies the assumption of an independent test sample. Table 3 records these boundaries and the principal controls.

Table 3. Executed source, selection, and evaluation sequence

Stage	Recorded sequence
Source audit	Two original CSV files; total 257,673 records, 42 predictors; duplicate hashes recorded
Split	65,865 training / 16,467 development inside train-side file; 175,341 test-side
Fitting	Five SVM-oriented alternatives under the approximate backend declared by notebook
Model selection	Development macro-F1; 20 hyperparameter-only and 48 per feature-search branch
Assessment	All five fitted models scored on the same test-side file; macro-F1 primary, class/efficiency metrics secondary

The recorded procedure in Table 3 summarizes the experiment's recorded split and model-selection sequence. Each alternative uses 65,865 fitting rows and 16,467 development rows. The reported 175,341-row test partition is scored after the model selection step. Row-hash overlap was detected despite the nominal separation between the source partitions. The score comparison is consequently a held-out-file comparison rather than a verified independent-flow evaluation. Contemporary intrusion benchmark studies emphasize the need to distinguish these notions [1]. This limitation is considered when interpreting the numerical differences.

2.3. Research Framework

Figure 1 illustrates the baseline and hybrid feature-selection framework. The schematic baseline uses the complete recorded predictor set. The proposed arm joins feature selection with classifier configuration search. The SVM backend was configured as approximate rather than independently verified as an exact RBF solver. Three additional controls isolate SVM tuning, GA-only selection, and random joint search. The five executed branches therefore provide a broader comparison than the two schematic branches. Class-balanced performance and input dimensionality are assessed on the same reported test file.

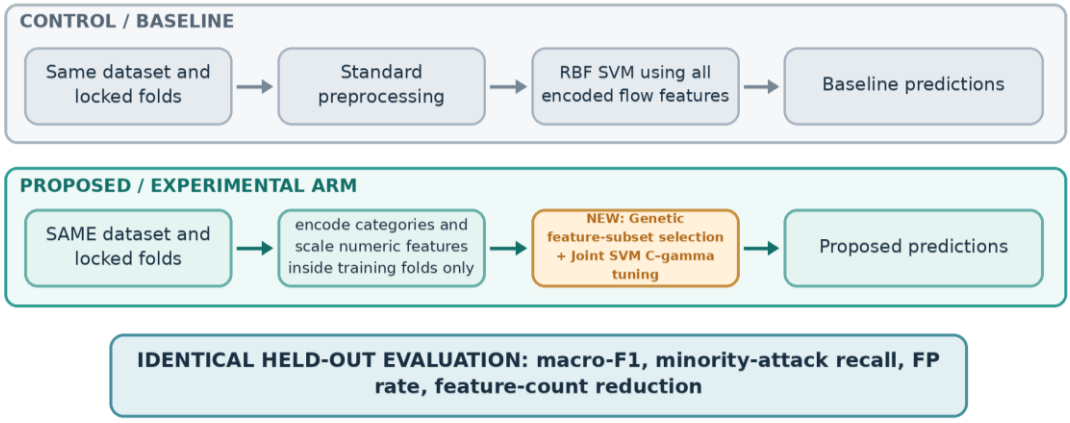

Figure 1. Controlled baseline-versus-proposed research framework

The research framework in Figure 1 separates common preprocessing from model-specific optimization. The baseline retains 42 original features. Each joint or feature-selection branch reports 27 selected columns. The diagram describes the intended architecture rather than independently proving which kernel approximation was computed. The executed SVM backend uses an approximate implementation. Stronger comparisons include the isolated-tuning and random-joint controls [7]. These distinctions prevent the visual from being mistaken for evidence of a complete exact-kernel search.

Table 4 identifies the components isolated in the five evaluated branches. The algorithm baseline uses all 42 predictors. The SVM-tuning branch retains all predictors while searching classifier settings. GA-only selection and the two joint methods use 27 columns. The random joint method provides a competing feature-and-parameter search. The combined GA-SVM branch is the paper's proposed configuration. Equal recorded evaluation counts are

interpreted separately from training time. Results are assessed against both the baseline and the strongest single-component control.

Table 4. Baseline, component ablations, and joint-search controls

Component	Executed configuration
Algorithm baseline	Approximate SVM backend; all 42 original predictors
SVM hyperparameters only	42 predictors; 20 objective evaluations
GA feature selection only	27 retained predictors; 48 objective evaluations
Random joint search	27 predictors; 48 feature/parameter evaluations
Proposed Joint GA-SVM	27 predictors; 48 joint GA feature/parameter evaluations

The ablation comparison in Table 4 shows that the feature reductions are held constant across three reduced-input alternatives. The baseline and hyperparameter-only branches retain 42 original predictors. A joint method cannot be credited with better feature reduction than its matched 27-feature control. The search objective is macro-F1 on development data. Random search and GA search are evaluated through the same outcome metrics. A component-specific improvement must therefore be demonstrated rather than inferred from the architecture diagram [7]. The final scores reported later determine the status of this claim.

2.4. Proposed Method

Figure 2 represents a hybrid genetic feature-subset and SVM-configuration workflow. The design jointly considers predictor inclusion and classifier settings. The combined experimental configuration is denoted Joint GA-SVM. It uses the approximate SVM backend specified in its configuration. The unavailable optimizer implementation prevents independent inspection of the genetic operators. The proposed search is therefore described according to recorded outputs rather than reconstructed internals. Figure 2 represents the conceptual architecture evaluated in the study.

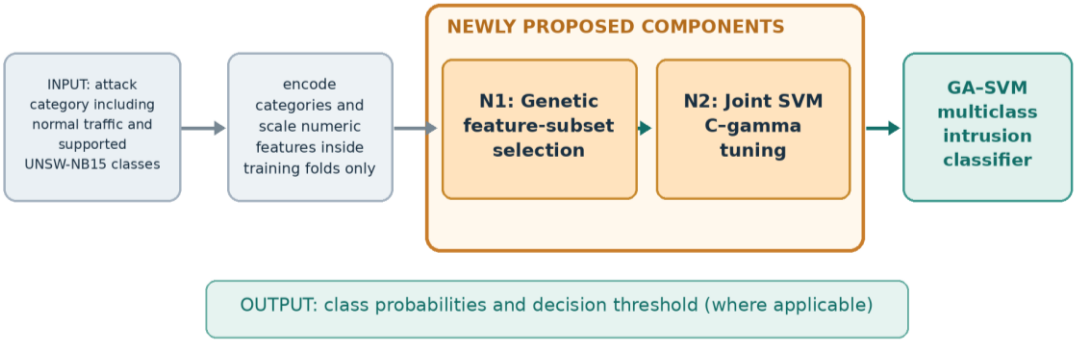

Figure 2. Proposed method architecture with explicitly highlighted novel components

The method architecture in Figure 2 illustrates feature-mask search followed by configuration of the approximate SVM backend. The executed classifier uses random Fourier features with a linear SVM instead of an exact RBF-SVC implementation. The selected hybrid representation contains 27 original predictors, compared with 42 for the baseline. This reduction is relevant to feature dimensionality but does not guarantee improved multiclass discrimination. The conceptual diagram refers to a probability output, although the available notebook does not establish a separately calibrated probability stage. The joint method is therefore evaluated using recorded test predictions and macro-F1, not an assumed calibration improvement. The results determine whether compression offsets the observed loss in minority-class detection.

Table 5 specifies the optimization representation and the measured endpoints. A binary feature mask encodes which original predictors are retained. Classifier search operates within the development partition. Macro-F1 is the main criterion used to compare candidate configurations. Feature count expresses the compression objective. Secondary endpoints include minority recall, overall accuracy, MCC, false-alarm rate, and inference time. The internal GA search operators could not be independently reconstructed from the available implementation evidence. These definitions distinguish the conceptual formulation from the available execution evidence.

Table 5. Feature-mask objective, classifier controls, and evaluated endpoints

Design element	Methodological definition
Input branch	Original flow predictors; coded categories and scaled numeric inputs
Component N1	Genetic feature-mask search; 27 of 42 original columns retained
Component N2	Joint SVM parameter search under approximate SVM backend
Output model	Multiclass attack label from the combined classifier
Conceptual fitness	Development macro-F1 is documented; any feature-count penalty in the GA internals is unverified
Ablation comparison	Baseline, hyperparameter-only, GA-only, random joint, joint GA-SVM
Primary and secondary endpoints	Macro-F1, accuracy, macro recall, AUC-OVR, benign false-alarm rate, features, runtime

The methodological definitions in Table 5 distinguish model representation from the performance criterion used for evaluation. The feature mask identifies selected input columns, while the approximate SVM backend uses a Fourier mapping and a linear classification stage. Development macro-F1 provides the directly documented model-selection measure. A feature-count penalty appears in the conceptual template but its coefficient and exact use cannot be independently verified from the uploaded optimizer artifacts. The combined GA–SVM branch selects 27 of the 42 original predictor columns. Reducing input size does not demonstrate better discrimination of minority attacks. Its performance must be interpreted against the original-feature baseline and the three alternative optimization controls.

The executed protocol covers the full original dataset with seed 42 and an approximate SVM backend. The source audit records 82,332 training-side rows and 175,341 test-side rows. The former is split into 65,865 fitting rows and 16,467 development rows. The baseline uses 42 original columns; three selection-based alternatives use 27. The development record includes 20 SVM-only parameter evaluations and 48 evaluations for each of the three feature-selection search branches, totaling 164. Search-related fit times are 1,466.984 seconds for tuning only, 6,077.891 seconds for random joint search, 3,576.109 seconds for GA-only selection, and 3,604.453 seconds for joint GA-SVM. The final reported metrics are calculated on the same 175,341-row test-side input. Hash-matching and duplicate diagnostics prevent the test file from being characterized as wholly independent of the training data.

Equations (1)–(8) define the hybrid classifier, its approximate kernel representation, and the principal evaluation measures. Equation (1) represents a subset of the d input predictors with a binary feature mask. Equation (2) defines the random Fourier feature map used to approximate the radial basis function kernel. Its projection vectors follow a normal distribution with covariance $2\gamma I$, and the phase offsets are uniformly distributed between zero and 2π. Equation (3) assigns the traffic class using linear multiclass scores calculated from the transformed inputs. Equation (4) gives equal weight to the F1 score of each of the K classes. Equation (5) expresses the development macro-F1 criterion used to compare search outcomes; it is not a reconstruction of undisclosed genetic-operator internals. Equations (6)–(8) quantify benign false alarms, input reduction, and the observed change relative to the baseline.

$$\mathbf{m} \in \{0,1\}^d, \qquad \mathbf{x}_i^{(\mathbf{m})} = \mathbf{x}_i \odot \mathbf{m} \tag{1}$$

$$\mathbf{z}_\gamma(\mathbf{x}) = \sqrt{\frac{2}{D}}\left[\cos\left(\boldsymbol{\omega}_j^\top \mathbf{x} + \beta_j\right)\right]_{j=1}^{D} \tag{2}$$

$$\hat{y}_i = \arg\max_{c \in \mathcal{C}} \left(\mathbf{w}_c^\top \mathbf{z}_\gamma\left(\mathbf{x}_i^{(\mathbf{m})}\right) + b_c\right) \tag{3}$$

$$F_{1,\text{macro}} = \frac{1}{K}\sum_{c=1}^{K} \frac{2\,TP_c}{2\,TP_c + FP_c + FN_c} \tag{4}$$

$$\left(\hat{\mathbf{m}}, \hat{C}, \hat{\gamma}\right) \in \arg\max_{\mathbf{m}, C, \gamma} F_{1,\text{macro}}^{\text{dev}}(\mathbf{m}, C, \gamma) \tag{5}$$

$$\text{FAR}_{\text{normal}} = \frac{\text{FP}_{\text{normal}}}{\text{FP}_{\text{normal}} + \text{TN}_{\text{normal}}} \tag{6}$$

$$\text{Reduction}(\%) = 100\,\frac{d - \|\mathbf{m}\|_0}{d} = 100\,\frac{42-27}{42} = 35.71\% \tag{7}$$

$$\Delta F_{1,\text{macro}} = F_{1,\text{GA-SVM}} - F_{1,\text{baseline}} = 0.359207 - 0.381508 = -0.022301 \tag{8}$$

The feature mask in Equation (1) retains a predictor when its corresponding binary entry equals one. The approximation in Equation (2) is appropriate for random Fourier features, rather than an exact nonlinear RBF-SVC fit. The transformed features are classified by the linear scores in Equation (3), with regularization controlled by C and the approximate RBF bandwidth controlled by γ. Equation (4) reports macro-F1 using one-versus-rest confusion counts for each observed class. The idealized maximization in Equation (5) states the development criterion without claiming an unverified feature-penalty coefficient in the unavailable optimizer source. Equation (6) defines the normal-traffic false-alarm rate as the proportion of benign observations assigned to attack classes. Equation (7) gives the observed 35.71% reduction from 42 original predictors to 27 selected predictors. Equation (8) records the negative macro-F1 difference of −0.022301, indicating that the proposed hybrid did not surpass the baseline on the recorded test file.

3. Results and Discussion

3.1. Dataset Distribution and Search Outcomes

The complete source record comprises 257,673 UNSW-NB15 observations. It provides 42 original traffic predictors before feature selection. The training-side file contributes 82,332 records, with 65,865 fitted and 16,467 reserved for development. The test-side file contributes 175,341 observations for final scoring. Traffic labels cover normal connections and nine attack categories. Figure 3 summarizes the class distribution and missing-value audit. The figure provides the descriptive basis for interpreting macro-averaged detection.

The recorded dataset distribution is visualized in Figure 3, with missingness shown alongside class support. Each panel refers to the loaded source records. The plot distinguishes normal traffic from nine attack categories. Rare classes are visible on a smaller scale than frequent classes. These differences motivate macro-averaged reporting. The figure depicts the recorded distribution and missingness without changing any experimental values.

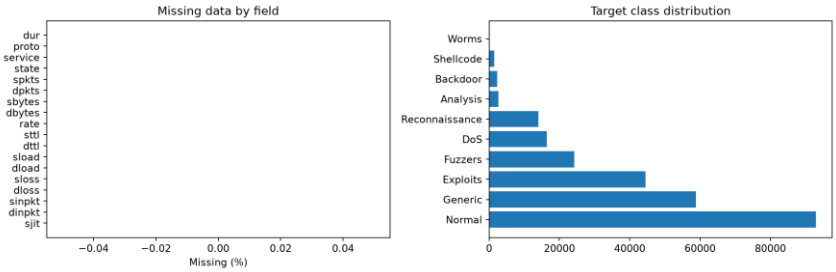

Figure 3. UNSW-NB15 class distribution and predictor missingness

The dataset visualization in Figure 3 shows a strongly uneven distribution across the ten traffic classes. Normal and Generic records are considerably more frequent than the rarest attacks. The displayed missingness assessment does not indicate missing values in the plotted predictor fields. Complete fields do not resolve class overlap or dependence between records. Hash diagnostics identify matching and repeated observations across the stored partitions. Class-balanced metrics remain necessary because majority classes can dominate overall accuracy. The figure describes the dataset rather than proving reliable detection of rare attacks.

Table 6 compares development-stage selection outcomes before test scoring. The algorithm baseline attains macro-F1 0.428678 on the development partition. SVM hyperparameter tuning alone reaches 0.438099, the highest recorded development objective. Joint GA-SVM reaches 0.427252 with 27 features. Random joint search reaches 0.425118 with the same feature count. GA-only selection obtains 0.409454 with 27 features. The recorded objective budgets are not identical between the 20-evaluation SVM tuning and 48-evaluation feature-search arms. These values must not be interpreted as final-test superiority.

Table 6. Development-stage macro-F1 and selected feature counts

Development model	Macro-F1 objective	Original predictor columns
Algorithm baseline	0.428678	42
SVM hyperparameters only	0.438099	42
Random joint search	0.425118	27
GA feature selection only	0.409454	27
Proposed Joint GA-SVM	0.427252	27

The development results in Table 6 show a small development benefit for SVM parameter tuning without feature reduction. The combined GA-SVM branch is slightly below the default on the same development objective. All reduced-input branches retain 27 of the original 42 predictors. The random joint search and GA-only search also underperform the development baseline. Their fitting times differ substantially despite a shared reported evaluation count. The final ranking therefore requires the test-side evaluation rather than development estimates alone. The result provides no preliminary basis for declaring the joint method superior.

3.2. Held-Out-File Baseline and Optimized Comparison

Table 7 presents the mandatory baseline-versus-proposed comparison on the 175,341-row test-side file. The default SVM achieves macro-F1 0.381508. The proposed joint GA-SVM achieves 0.359207. Both accuracy and balanced accuracy decline under the proposed model. The proposed model attains slightly higher macro one-versus-rest ROC-AUC, but lower weighted F1 and MCC. Benign false-alarm rate rises slightly while inference time per 1,000 predictions decreases. The table reports the relevant trade-offs without assigning an unsupported overall improvement.

Table 7. Held-out-file comparison: default SVM versus joint GA-SVM

Indicator	Algorithm baseline	Joint GA-SVM
Macro-F1 (primary)	0.381508	0.359207
Macro-F1 delta vs baseline	0.000000	-0.022301
Accuracy	0.718098	0.710302
Balanced accuracy	0.384733	0.368182
Macro precision	0.509109	0.466282
Macro recall	0.384733	0.368182
Weighted F1	0.680497	0.667596
Cohen kappa	0.632787	0.621686
MCC	0.646627	0.635619
Macro ROC-AUC (OVR)	0.914785	0.916670
Log loss	0.787270	0.821665
Benign false-alarm rate	0.021196	0.021375
Min. attacker-class recall	0.000000	0.000000
Original predictor columns	42	27
Inference ms / 1000 records	20.223	16.573

The held-out comparison in Table 7 establishes a macro-F1 reduction of 0.022301 for joint GA-SVM relative to the baseline. Accuracy decreases from 0.718098 to 0.710302. Balanced accuracy decreases from 0.384733 to 0.368182. Macro one-versus-rest ROC-AUC increases from 0.914785 to 0.916670, but this small ranking change does not reverse the primary result. Feature selection reduces the original columns from 42 to 27, or 35.71%. Benign false-alarm rate rises from 0.021196 to 0.021375. Inference latency decreases from 20.223 to 16.573 milliseconds per 1,000 predictions in the recorded environment. No superiority claim is supported on the predeclared macro-F1 criterion.

Table 8 provides the primary metric for all five fitted alternatives rather than only the two principal methods. The first row is the all-feature baseline. The tuning-only model retains 42 predictors while changing classifier settings. Both feature-search controls retain 27 predictors. The proposed Joint GA-SVM occupies the final row. Every value is a recorded score from the same test-side file. The table isolates the relevant negative result without omitting intermediate controls.

Table 8. Multiclass macro-F1 and input dimensionality across all controls

Model	Macro-F1	Original features
Algorithm baseline	0.381508	42
SVM hyperparameters only	0.377752	42
Random joint feature/parameter search	0.372305	27
GA feature selection only	0.360482	27
Joint GA-SVM	0.359207	27

The five-model evaluation in Table 8 confirms that none of the four optimized alternatives exceeds the baseline macro-F1. The strongest optimized control is SVM hyperparameter tuning, with macro-F1 0.377752. Random joint search follows with 0.372305. GA-only selection reaches 0.360482. Joint GA-SVM records the lowest macro-F1 at 0.359207. Three alternatives nevertheless reduce the original predictor count from 42 to 27. The results distinguish input compression from the study's primary classification criterion.

Figure 4 compares macro-F1, macro recall, and overall accuracy across all five evaluated models. The default configuration has the highest final macro-F1. Parameter tuning alone produces slightly higher accuracy but a lower macro-F1. Both joint search methods perform below the baseline on class-balanced metrics. Feature selection only also has lower macro-F1. The comparison is based on the same test-side records. These plots make the choice of evaluation metric consequential.

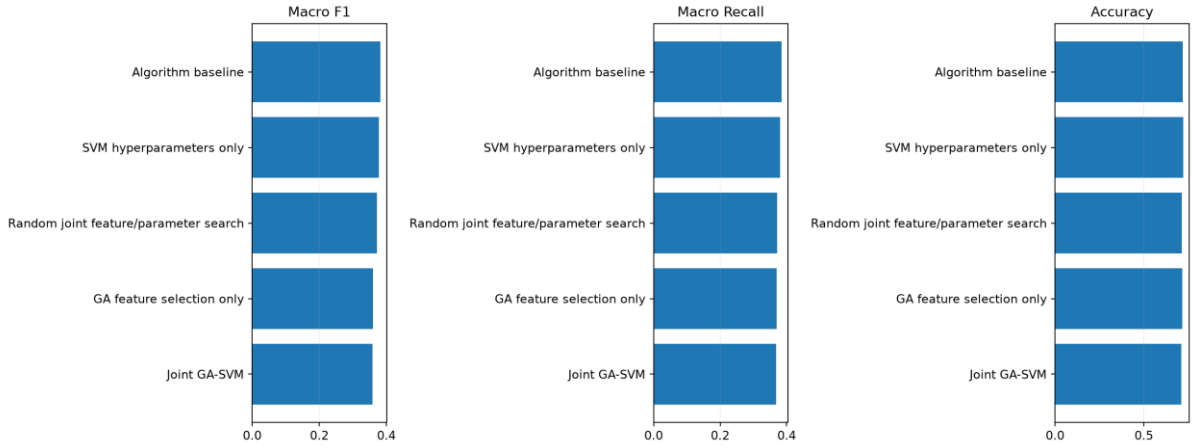

Figure 4. Recorded macro-F1, macro recall, and accuracy across five models

The metric comparison in Figure 4 illustrates that overall accuracy does not identify the best model under the macro-F1 criterion. SVM tuning yields accuracy 0.720562 but macro-F1 0.377752. The baseline yields accuracy 0.718098 and macro-F1 0.381508. The proposed joint model records accuracy 0.710302 and macro-F1 0.359207. Macro recall falls in the same direction as macro-F1. This pattern indicates that feature reduction did not preserve class-balanced classification performance in the recorded run. The result is a negative primary comparison despite the smaller input representation.

Figure 5 traces the recorded development search objective across 164 evaluations. Its best-so-far curve increases early and later stabilizes. Candidate values vary substantially below the best observed development score. A rising best-so-far curve is a property of an optimization trace, not evidence of test improvement. The search includes the recorded optimization branches but does not establish equal wall-clock budgets. The figure is interpreted only within the development boundary. The final-test macro-F1 comparison remains decisive.

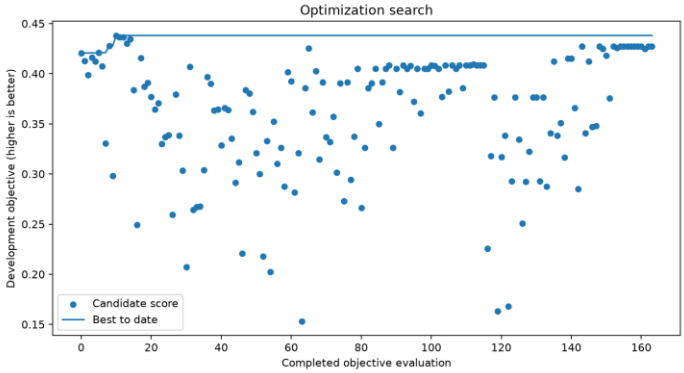

Figure 5. Development-stage optimization trace across completed objective evaluations

The recorded optimizer trace in Figure 5 shows the distribution of candidate development objectives rather than a validated learning curve. Some configurations achieve considerably lower scores than the selected candidate. The recorded maximum provides evidence that a search process was executed. It does not indicate which internal genetic operations generated each candidate without the imported source code. Random joint search and GA-based search

consume different measured fitting times. The strongest development model is not necessarily the strongest on the overlapping test-side file. The plot therefore informs search behavior but cannot establish generalization.

3.3. Per-Class Errors and Detection Trade-Offs

Figure 6 shows the baseline multiclass confusion matrix. The diagonal represents correct predictions for the ten traffic labels. The off-diagonal cells reveal confusions between attack categories and normal traffic. Several frequent categories dominate the displayed counts. The matrix helps explain why a single aggregate accuracy can obscure rare-class performance. Identical class labels are used for the proposed-model matrix in Figure 7. Both matrices describe the recorded test-side file.

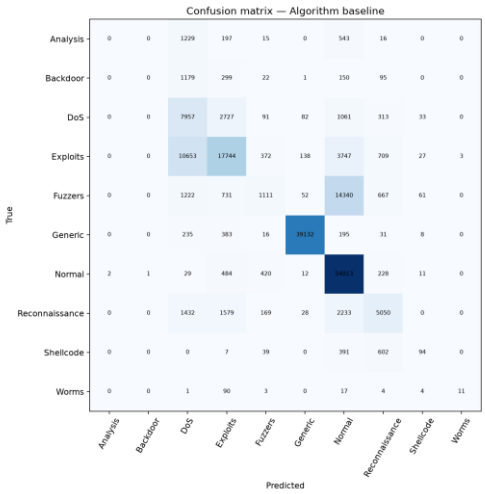

Figure 6. Confusion matrix for the original-feature SVM baseline

The baseline confusion matrix in Figure 6 demonstrates uneven recognition among individual attack categories. Some minority-class records are assigned to more frequent classes. The nonzero diagonal for large classes contributes strongly to aggregate accuracy. Macro averaging gives small attack classes equal weight in the primary score. The baseline nevertheless achieves the best macro-F1 among the compared models. No conclusion about unseen attack families can be drawn from the matrix. The recorded hash overlap further restricts independent generalization claims. Figure 7 presents the joint GA-SVM confusion matrix using the same attack labels. Its prediction profile differs from the baseline after reducing the selected predictors to 27. Several rare attack categories remain difficult to identify. Some off-diagonal counts are large relative to the diagonal of their class. These outcomes help explain the lower macro-F1 of the combined model. The figure is descriptive rather than a source for post-test model selection. It complements the numerical comparison in Table 7.

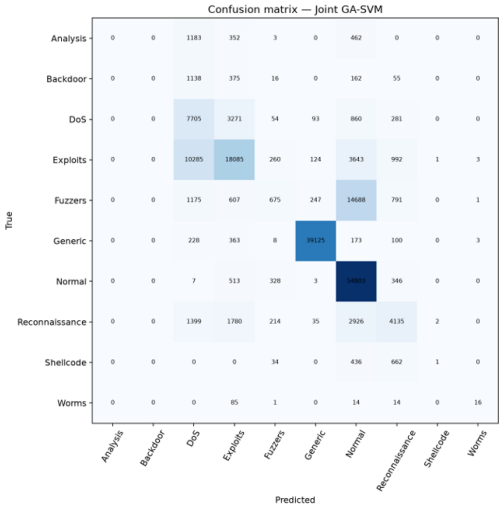

Figure 7. Confusion matrix for the proposed Joint GA-SVM

The hybrid confusion matrix in Figure 7 shows that reduced dimensionality does not ensure preservation of class-specific detection. At least one attacker class has zero recorded recall in the model-level minimum-recall

metric. The joint method does not achieve a higher macro-F1 than the all-feature baseline. The matrix contains substantial confusion among attack types with overlapping traffic signatures. False predictions remain relevant even where the number of input columns is smaller. No causal attribution to a particular deleted feature is possible without an additional controlled analysis. The observation is consistent with a representation-compression trade-off rather than a uniform classifier improvement.

Figure 8 reports class-wise recall for the joint GA-SVM configuration. The ten bars include normal traffic and nine attack categories. The recall values differ widely across labels. At least one rare attack category has no correctly recalled test observations. Larger classes have markedly higher recall than some minority classes. This imbalance is hidden when only overall accuracy is reported. The figure directly addresses the study's multiclass detection objective.

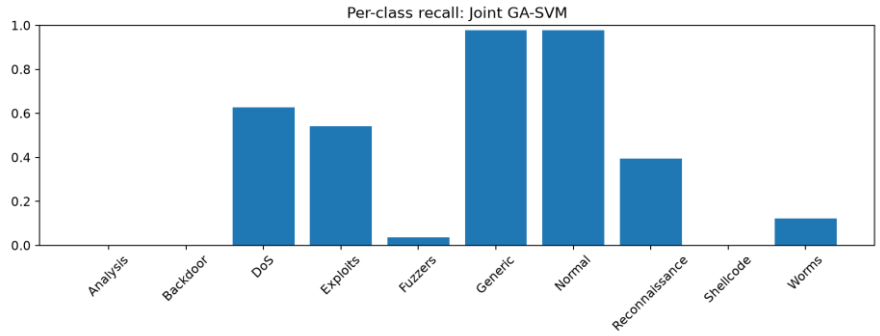

Figure 8. Recorded per-class recall for the proposed Joint GA-SVM

The per-class detection profile in Figure 8 highlights the gap between common-traffic and minority-attack detection. Some classes receive near-zero recall even when aggregate accuracy exceeds 0.70. The proposed method's macro recall is 0.368182, below the baseline value of 0.384733. The minimum attacker-class recall is zero for every recorded model. Feature compression therefore cannot be described as maintaining reliable detection across all attacks. Remediation would require attention to class imbalance and a verified, leakage-safe evaluation protocol. The current result supports further methodological investigation rather than deployment claims.

3.4. Interpretation and Validity Limitations

The proposed hybrid GA-SVM reduces the input representation by 15 of the original 42 columns. Its macro-F1 decreases from 0.381508 to 0.359207 relative to the algorithm baseline. The change is approximately a 5.85% relative decrease in the primary score. SVM tuning alone, GA-only selection, and random joint search also fail to exceed the baseline macro-F1. The hybrid yields slightly improved macro ROC-AUC and lower measured inference latency, which are secondary benefits. Those changes do not compensate for the reduction in class-balanced detection. The evidence does not support a superior intrusion classifier under the recorded primary objective.

The evaluation is constrained by nonindependent traffic records and the absence of independently available optimization source modules. A row-hash audit identified 10,279 test rows matching training-side records. The same audit identified 74,301 repeated rows within the test file. These counts are based on row hashing and should be confirmed with exact-row checks before a reproducibility claim is finalized. The data are not organized into verified time-separated or host-independent groups. Reported timing reflects a particular Windows computational environment rather than a deployed detection system. These limitations warrant caution in comparing results with studies that use other UNSW-NB15 splits. Additional paired trials on decontaminated flow groups are needed to assess robustness.

4. Conclusion

This study evaluated a hybrid genetic feature-selection and SVM configuration strategy on the recorded 257,673-row UNSW-NB15 benchmark. Five alternatives were compared using a fixed development partition and a common 175,341-row test-side file. The proposed GA-SVM retained 27 of 42 original predictors, a reduction of 35.71%. Its macro-F1 was 0.359207 versus 0.381508 for the all-feature SVM baseline. The hybrid also had lower balanced accuracy and a slightly higher benign false-alarm rate. A small improvement in macro ROC-AUC and reduced inference time did not change the unfavorable primary comparison. The results therefore support feature compression but not an incremental macro-F1 benefit from the combined search. Documented row overlap, test duplication, zero recall for at least one attack class, and unavailable optimizer internals limit broader inference.

Future work should evaluate an auditable hybrid implementation under exact-row decontamination, repeated paired seeds, and group- or time-aware testing.

References

[1] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis," *Algorithms*, vol. 17(2), p. 64, 2024, doi: 10.3390/a17020064.

[2] Z. Zoghi and G. Serpen, "Building an intrusion detection system on UNSW-NB15: Reducing the margin of error to deal with data overlap and imbalance," *Concurr. Comput.*, vol. 36(25), p. e8242, 2024, doi: 10.1002/cpe.8242.

[3] S. A. Ajagbe, J. B. Awotunde, and H. Florez, "Intrusion Detection: A Comparison Study of Machine Learning Models Using Unbalanced Dataset," *SN Comput. Sci.*, vol. 5, p. 1028, 2024, doi: 10.1007/s42979-024-03369-0.

[4] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *J. Big Data*, vol. 11, p. 36, 2024, doi: 10.1186/s40537-024-00892-y.

[5] E. M. Maseno and Z. Wang, "Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection," *J. Big Data*, vol. 11, p. 24, 2024, doi: 10.1186/s40537-024-00887-9.

[6] H. Bakir and O. Ceviz, "Empirical Enhancement of Intrusion Detection Systems: A Comprehensive Approach with Genetic Algorithm-based Hyperparameter Tuning and Hybrid Feature Selection," *Arab. J. Sci. Eng.*, vol. 49, pp. 13025–13043, 2024, doi: 10.1007/s13369-024-08949-z.

[7] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 13(2), p. e1484, 2023, doi: 10.1002/widm.1484.

[8] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning," *J. Big Data*, vol. 11, p. 133, 2024, doi: 10.1186/s40537-024-00994-7.

[9] Y. Yin *et al.*, "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *J. Big Data*, vol. 10, p. 15, 2023, doi: 10.1186/s40537-023-00694-8.

[10] M. A. Talukder *et al.*, "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *J. Big Data*, vol. 11, p. 33, 2024, doi: 10.1186/s40537-024-00886-w.

[11] A. Telikani *et al.*, "A Cost-Sensitive Machine Learning Model With Multitask Learning for Intrusion Detection in IoT," *IEEE Trans. Industr. Inform.*, vol. 20(3), pp. 3880–3890, 2024, doi: 10.1109/TII.2023.3314208.

[12] M. A. Umar, Z. Chen, K. Shuaib, and Y. Liu, "Effects of feature selection and normalization on network intrusion detection," *Data Science and Management*, vol. 8(1), pp. 23–39, 2025, doi: 10.1016/j.dsm.2024.08.001.

[13] I. B. Aldallal, A. A. Ibrahim, and S. R. Ahmed, "An Intelligent Multi-Stage GA-SVM Hybrid Optimization Framework for Feature Engineering and Intrusion Detection in Internet of Things Networks," *Computers, Materials & Continua*, vol. 87(1), p. 39, 2026, doi: 10.32604/cmc.2025.075212.

[14] Y. K. Saheed, O. H. Abdulganiyu, and T. Ait Tchakoucht, "Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities," *Appl. Soft Comput.*, vol. 155, p. 111434, 2024, doi: 10.1016/j.asoc.2024.111434.

[15] M. Mahiuddin and M. S. Islam, "A Fusion and Genetic Optimization-Based Feature Selection Method With GRU for Intrusion Detection System," *Applied Computational Intelligence and Soft Computing*, vol. 2026, p. 1937064, 2026, doi: 10.1155/acis/1937064.

[16] A. Kayyidavazhiyil, "Intrusion detection using enhanced genetic sine swarm algorithm based deep meta-heuristic ANN classifier on UNSW-NB15 and NSL-KDD dataset," *Journal of Intelligent & Fuzzy Systems*, vol. 45(6), pp. 10243–10265, 2023, doi: 10.3233/JIFS-224283.

[17] A. Saputra, K. Ramli, A. S. Nugroho, I. G. D. Nugraha, and B. Pranggono, "A Novel Hybrid IGL1 Feature Selection Method for High-Performance Intrusion Detection on the UNSW-NB15 Dataset Using Multiple Machine Learning Models," *Big Data and Cognitive Computing*, vol. 10(6), p. 182, 2026, doi: 10.3390/bdcc10060182.

[18] O. Ebrahem, S. Dowaji, and S. Alhammoud, "Towards a minimum universal features set for IoT DDoS attack detection," *J. Big Data*, vol. 12, p. 88, 2025, doi: 10.1186/s40537-025-01146-1.

[19] S. M. Hashemi Jouybari, A. Janbaz, and A. Esfandiari, "A multivariate filter feature selection and stacking-based ensemble learning algorithm for network intrusion detection," *J. Supercomput.*, vol. 81, p. 1129, 2025, doi: 10.1007/s11227-025-07609-9.

[20] Y. Huang, G. Chen, J. Gou, Z. Fan, and Y. Liao, "A hybrid feature selection and aggregation strategy-based stacking ensemble technique for network intrusion detection," *Applied Intelligence*, vol. 55, p. 28, 2025, doi: 10.1007/s10489-024-06015-7.

[21] H. A. Ahmed, A. Hameed, and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Comput. Sci.*, vol. 8, p. e820, 2022, doi: 10.7717/peerj-cs.820.

[22] T. B. Shana, N. Kumari, M. Agarwal, S. Mondal, and U. Rathnayake, "Anomaly-based intrusion detection system based on SMOTE-IPF, Whale Optimization Algorithm, and ensemble learning," *Intelligent Systems with Applications*, vol. 27, p. 200543, 2025, doi: 10.1016/j.iswa.2025.200543.

[23] H. Zouhri and A. Idri, "A novel CTGAN-ENN hybrid approach to enhance the performance and interpretability of machine learning black-box models in intrusion detection and IoT," *Future Generation Computer Systems*, vol. 173, p. 107882, 2025, doi: 10.1016/j.future.2025.107882.

[24] R. Jablaoui and N. Liouane, "GA-CNN-BiGRU-IDS: A robust framework for Intrusion Detection System based on GA for data augmentation and hybrid CNN-BiGRU model for spatiotemporal feature extraction," *Computers and Electrical Engineering*, vol. 130, p. 110900, 2026, doi: 10.1016/j.compeleceng.2025.110900.

[25] C. Rout, S. Sethi, J. C. Badajena, and R. K. Sahoo, "FSEGM: feature selection and ensemble generative model for adaptive cloud security," *Journal of Cloud Computing*, vol. 15, p. 6, 2026, doi: 10.1186/s13677-025-00818-w.

[26] H. Benaddi, M. Jouhari, and O. Elharrouss, "A lightweight hybrid approach for intrusion detection systems using a chi-square feature selection approach in IoT," *Internet of Things*, vol. 32, p. 101624, 2025, doi: 10.1016/j.iot.2025.101624.

[27] S. K. R. Mallidi and R. R. Ramisetty, "A multi-level intrusion detection system for industrial IoT using bowerbird courtship-inspired feature selection and hybrid data balancing," *Discover Computing*, vol. 28, p. 109, 2025, doi: 10.1007/s10791-025-09632-z.

[28] J. Pospichal, A. Augustin, L. Huraj, P. Strelec, and D. Gabriska, "Random Forest-Based Network Intrusion Detection with Feature Selection and Class Balancing on UNSW-NB15 Traffic," *Information*, vol. 17(9), p. 836, 2026, doi: 10.3390/info17090836.

[29] Z. Qin, L. Li, Q. Luo, X. Yu, and Q. Zhang, "Enhancing intrusion detection performance using GCN-LOF: A hybrid graph-based anomaly detection approach," *Computer Networks*, vol. 271, p. 111606, 2025, doi: 10.1016/j.comnet.2025.111606.

[30] G. P. Oise *et al.*, "Attention-based intrusion detection using transformer networks on UNSW-NB15," *Sci. Rep.*, vol. 16, p. 33862, 2026, doi: 10.1038/s41598-026-65710-2.

[31] T. de Menezes e Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: A survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112(9), pp. 3211–3260, 2023, doi: 10.1007/s10994-023-06336-7.

[32] T. Liu, Y. Fu, K. Wang, X. Duan, and Q. Wu, "A multiscale approach for network intrusion detection based on variance-covariance subspace distance and EQL v2," *Comput. Secur.*, vol. 148, p. 104173, 2025, doi: 10.1016/j.cose.2024.104173.

[33] S. Mansouri and S. Hosseini, "A Chaos-Enhanced Hybrid Metaheuristic Feature Selection for Intrusion Detection Using Artificial Neural Networks," *International Journal of Computational Intelligence Systems*, vol. 19, p. 307, 2026, doi: 10.1007/s44196-026-01417-6.

[34] M. A. Talukder *et al.*, "A dependable hybrid machine learning model for network intrusion detection," *Journal of Information Security and Applications*, vol. 72, p. 103405, 2023, doi: 10.1016/j.jisa.2022.103405.

[35] K. Zhang, Y. Wang, and U. A. Bhatti, "Enhanced ransomware attacks detection using feature selection, sensitivity analysis, and optimized hybrid model," *J. Big Data*, vol. 12, p. 245, 2025, doi: 10.1186/s40537-025-01289-1.