Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 1 April 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 284-305
DOI: https://doi.org/10.51903/jtie.v4i1.531

Language-Guided Feature Selection for DDoS and Intrusion Detection on CICIDS2017

Yunhe Li*[1], Shenghan Lu[2]
Email: lyunh@alumni.upenn.edu
[1]*Computer and Information Technology, University of Pennsylvania, PA, USA*
[2]*Information Technology, Fordham University, NY, USA*
*Corresponding Author

Abstract
This paper reports a complete empirical study of language-guided feature selection for DDoS and intrusion detection on the CICIDS2017 MachineLearningCSV flow data. The central question is whether an LLM-style semantic reading of CICFlowMeter feature names can reduce the feature set while preserving detection performance and lowering false alarms. The experiment used the eight labeled CICIDS2017 CSV sessions, removed only non-finite numeric rows, and retained 2,827,876 flows with 78 original numeric features. A semantic feature screen selected 32 features describing service context, duration, packet and byte volume, flow rates, inter-arrival timing, TCP flags, window sizes, and active/idle behavior. The evaluation compared all features with the language-selected set under full-corpus binary and multiclass stochastic logistic regression, DDoS-specific Random Forest, DDoS-specific stochastic logistic regression, and a compact multilayer perceptron. The best DDoS result was obtained by Random Forest with the selected features: F1 = 0.999896, false-positive rate = 0.000068, and eight errors on 67,714 test flows. The selected features reduced the DDoS Random Forest training time by 23.78% and reduced full-corpus SGD training time by about one half, although the full feature set was stronger for the full binary linear model. Ablation showed that TCP flag/window and destination-port semantics produced the largest DDoS degradation when removed. The findings support language-guided feature selection as a practical compression step for latency-sensitive DDoS mitigation, while retaining all features remains advisable for broad multiclass intrusion detection when a linear learner is used.

Keywords: *CICIDS2017, DDoS Detection, Intrusion Detection CICFlowMeter, Language-Guided Feature Selection, Cybersecurity.*

I. INTRODUCTION

Network intrusion detection is a feature-rich classification problem in which measurements of flow volume, timing, directionality, flags, and application-facing ports must be converted into timely security decisions (Al Alim et al., 2025; Handoko et al., 2025; Hartono et al., 2024). The practical difficulty is not only model accuracy. In deployed DDoS mitigation, the detector must also avoid excessive false alarms, run quickly enough for operational response, and remain understandable to analysts. The base-rate problem described by (Axelsson, 2000) means that even a small false-positive rate can create many erroneous alerts when benign traffic dominates. (Sommer and Paxson, 2010) similarly cautioned that intrusion detection differs from many closed-world machine learning tasks because traffic distributions, attack behavior, and measurement artifacts can shift. These concerns motivate feature selection methods that reduce computational cost without concealing the network semantics that analysts use to interpret decisions.

Received on Januari 2025; Revised on February 2025; Accepted on March 2025; Published on April 2025

CICIDS2017 is a widely used benchmark because it provides benign traffic and common attack classes in flow-level CSV form generated from CICFlowMeter. The dataset was designed to improve on earlier benchmarks that lacked modern attack diversity or realistic benign profiles (Sharafaldin et al., 2018). The official MachineLearningCSV format used in this study contains 78 numeric CICFlowMeter features and a label column. Its classes include BENIGN, DDoS, PortScan, DoS variants, brute-force attacks, Bot, Web attacks, Infiltration, and Heartbleed. Such breadth is useful, but it also creates a difficult imbalance: the cleaned corpus used here contained 2,271,320 benign flows, while three labels contained fewer than 40 flows. This imbalance is central to interpreting macro-F1, multiclass results, and false-positive rates.

Feature selection has traditionally relied on filter scores, wrapper search, embedded tree importance, or dimensionality reduction (Chandrashekar & Sahin, 2014; Guyon & Elisseeff, 2003). These methods are valuable, but they often treat feature names as arbitrary variables. Recent progress in language models suggests a complementary approach: use language understanding to read feature semantics and nominate measurements that match the causal vocabulary of the attack. Transformers and large language models demonstrated that language representations can encode rich semantic relationships and support few-shot reasoning over text descriptions (Brown et al., 2020; Vaswani et al., 2017). This paper does not call an external proprietary model during training; instead, it operationalizes the idea of language-guided feature selection by applying a documented semantic rubric to CICFlowMeter feature names.

The contribution is a reproducible empirical evaluation, not a conceptual placeholder. The study downloaded the actual CICIDS2017 MachineLearningCSV files, cleaned them with a fixed rule, trained the stated models, and generated tables and figures from measured outputs. The analysis compares all 78 numeric features with the 32 language-selected features for both broad intrusion detection and DDoS-specific detection. It also reports ablations that remove semantic feature groups from the selected set. The results show a nuanced pattern: language-selected features work extremely well for DDoS Random Forest detection and substantially reduce training time, but the all-feature set remains stronger for the full binary linear model. This distinction is important because feature selection should be judged against a deployment goal rather than assumed to dominate every model family.

A second motivation is operational parsimony. Many CICFlowMeter columns are meaningful, but not all are equally useful in a mitigation appliance that must score flows rapidly or explain decisions to security staff. Feature computation itself can be a cost when packet capture, flow aggregation, and model scoring are deployed close to the network edge. A smaller feature list can simplify telemetry contracts, reduce storage, and make drift monitoring easier. At the same time,

selecting too few features can damage recall on minority attacks or remove redundant evidence that a model uses for robustness. The experimental design therefore evaluates both effectiveness and cost, rather than treating feature selection as a purely dimensionality-reduction exercise.

The paper's title uses the phrase language-guided rather than automatic LLM optimization (Sun et al., 2024) because the feature set is selected from natural-language semantics of feature names and not from test-set feedback. The selection procedure asks what each CICFlowMeter feature means in network terms and whether that meaning should help identify DDoS or broad intrusion behavior. This distinction is important for reproducibility: another researcher can read Table 4, verify the retained features, and rerun the provided scripts without relying on an unavailable prompt transcript. The approach is therefore a transparent bridge between analyst knowledge and language-model style semantic screening.

The research question was intentionally framed as a comparison rather than a one-sided proof. A semantic subset could be better if it removes noisy or redundant columns, but it could also be worse if the omitted columns carry weak signals that a model combines with other variables. The manuscript therefore reports both cases. The selected features succeed in the DDoS Random Forest setting and reduce cost across models, while all features remain useful for full binary linear intrusion detection. This balanced interpretation is necessary for a paper that will be reviewed for logical coherence.

II. LITERATURE REVIEW

Early intrusion detection research depended heavily on benchmark datasets, but the limitations of those datasets quickly became apparent. (McHugh, 2000) criticized the DARPA evaluations for methodological weaknesses, while (Tavallaee et al., 2009) showed that KDD Cup 99 contained redundancy and difficulty-level issues that distorted model comparisons. The UNSW-NB15 dataset was later proposed to provide more contemporary traffic and attack families (Moustafa & Slay, 2015). (Ring et al., 2019) surveyed network-based intrusion detection datasets and emphasized that the realism, labeling strategy, and feature-generation process strongly affect conclusions. CICIDS2017 responded to this lineage by combining benign user profiles with multiple attack scenarios and CICFlowMeter features (Sharafaldin et al., 2018). Nevertheless, (Engelen et al., 2021) showed that CICIDS2017 also has labeling and flow-construction caveats, making transparent preprocessing and reproducible splits necessary.

Machine learning for network intrusion detection spans linear classifiers, trees, ensembles, kernel methods, neural networks, and anomaly detectors. (Buczak & Guven, 2016) reviewed data mining and machine learning methods in cybersecurity and showed that no single model class is universally sufficient. Random Forests are attractive in flow-based IDS because ensembles handle

nonlinear interactions and mixed feature scales while remaining comparatively robust to unimportant variables (Breiman, 2001). Deep learning methods have also been used for intrusion detection, including recurrent and autoencoder architectures (Mirsky et al., 2018; Yin et al., 2017). Deep models can capture complex interactions, but they require careful optimization and are harder to interpret without additional explanation methods.

Feature selection is a recurring theme because flow datasets contain redundant and highly correlated measurements (Zheng & Li, 2024). Some CICFlowMeter features describe similar concepts at different aggregation levels, such as total bytes, subflow bytes, packet length statistics, and forward/backward segment sizes. (Guyon and Elisseeff, 2003) described feature selection as a way to improve generalization, interpretability, and computational efficiency. (Chandrashekar & Sahin, 2014) grouped feature selection into filter, wrapper, and embedded approaches. Filter methods are fast but may miss model-specific interactions; wrapper methods can be accurate but expensive; embedded methods, including tree-based importance, link selection to the trained model. For imbalanced IDS settings, feature selection must also be judged by false-positive behavior, not only accuracy, because high accuracy can be achieved by favoring the majority benign class.

Language-guided feature selection adds a different signal. Rather than ranking columns only by statistics, the analyst or LLM (Mu et al., 2023) reads feature names and groups them by network meaning. DDoS attacks often change service-facing ports, flow duration, packet and byte counts, packet rates, inter-arrival times, TCP control flags, window sizes, and idle/active cycles. These are the semantic groups represented in the selected 32 features. The idea is compatible with explainable AI work: (Ribeiro et al., 2016; Lundberg and Lee, 2017) argued that explanations must be understandable to humans, while language-guided selection starts from human-readable feature semantics. It is also compatible with transformer-era language reasoning, because the text descriptions of features provide context beyond numeric distributions (Brown et al., 2020; Devlin et al., 2019).

The literature also warns that strong benchmark performance does not automatically imply real-world generalization. (Sommer & Paxson, 2010) argued that operational intrusion detection must account for adversarial adaptation, deployment context, and difficult base rates. (Garcia-Teodoro et al., 2009) surveyed anomaly-based IDS and highlighted the tension between detecting unknown attacks and limiting false alarms. The present study therefore reports false-positive rates, confusion matrices, and ablation results rather than only accuracy. It uses CICIDS2017 as a controlled benchmark while treating the conclusions as deployment guidance for feature reduction, not as a claim of universal attack coverage.

Class imbalance is a persistent difficulty in IDS research. (Chawla et al., 2002) proposed SMOTE to address minority-class scarcity, and many IDS studies use over-sampling or under-sampling to improve rare-class recall. This study deliberately did not use SMOTE because the objective was to test feature selection on the dataset as observed. Synthetic balancing can be useful, but it can also blur the relationship between real flow measurements and attack semantics. The result is that macro-F1 for multiclass detection is low, which is an honest consequence of rare labels rather than an artifact hidden by resampling.

CICIDS2017-specific studies often report very high performance when the task is restricted to large, separable classes such as DDoS or PortScan. More difficult outcomes are observed when all attack labels are retained, especially for Infiltration, Heartbleed, and SQL Injection. (Engelen et al., 2021) further showed that flow construction and labeling details can change the interpretation of results. The present study handles this by separating a DDoS mitigation task from a broad full-corpus intrusion task. This separation prevents the strong DDoS result from being overstated as a universal multiclass IDS result.

Tooling also matters for reproducibility. Scikit-learn provides stable implementations of SGD and Random Forest models (Pedregosa et al., 2011), while PyTorch provides a widely used deep-learning framework (Paszke et al., 2019). The models used here are intentionally standard and lightweight. The purpose is not to introduce a new classifier architecture, but to test whether a semantically selected feature subset remains competitive across common model families. This design makes the empirical claim easier to audit because the result depends on explicit data files, feature lists, splits, and hyperparameters (Zhao et al., 2023).

Another strand of related work concerns interpretability. In security operations, a detector that reports only a score is less useful than a detector whose inputs can be mapped back to network behavior. LIME and SHAP-style explanations showed that local and additive explanations can help users understand model outputs (Lundberg & Lee, 2017; Ribeiro et al., 2016). Language-guided selection is not a replacement for explanation methods, but it improves the starting point by ensuring that the retained inputs already have analyst-readable meanings. A selected DDoS feature such as Flow Packets/s or SYN Flag Count can be discussed directly with responders, whereas an opaque transformed component would require additional explanation.

III. RESEARCH METHOD

The experiment used the CICIDS2017 MachineLearningCSV flow files corresponding to the five-day capture. The eight CSV files were retained as separate session files during cleaning and then combined for the full-corpus experiments. Table 1 documents the experimental package, and Table 2 reports the row counts observed directly from the downloaded files. The raw corpus

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 1 April 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 284-305
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

contained 2,830,743 flows. Columns were stripped of leading spaces, all 78 non-label columns were treated as numeric CICFlowMeter features, and rows with any non-finite feature value were removed. This removed 2,867 rows and left 2,827,876 clean flows. No synthetic oversampling, undersampling, PCA, or label relabeling was applied. This decision kept the measured results tied to the available data and avoided introducing artificial traffic patterns.

Table 1. Experimental Dataset Source and Reproducibility Package

Item	Value
Dataset	CICIDS2017 MachineLearningCSV flow files
Files used	Eight labeled CSV sessions: Monday, Tuesday, Wednesday, Thursday morning, Thursday afternoon, Friday morning, Friday PortScan, Friday DDoS
Raw rows observed	2,830,743
Clean rows used	2,827,876
Original numeric features	78
Language-selected features	32
Random seed	42
Generated artifacts	Python scripts, cleaned metrics, figures, Word manuscript, and compressed data/code package

Table 2. CICIDS2017 Session Files and Cleaning Outcomes

File/Session	Raw Rows	Clean Rows Used	Removed	Benign	Attack	Labels
Monday-WorkingHours	529918	529481	437	529481	0	BENIGN
Tuesday-WorkingHours	445909	445645	264	431813	13832	BENIGN, FTP-Patator, SSH-Patator
Wednesday-workingHours	692703	691406	1297	439683	251723	BENIGN, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks	170366	170231	135	168051	2180	BENIGN, Web Attack Brute Force, Web Attack Sql Injection, Web Attack XSS
Thursday-WorkingHours-Afternoon-Infilteration	288602	288395	207	288359	36	BENIGN, Infiltration
Friday-WorkingHours-Morning	191033	190911	122	188955	1956	BENIGN, Bot
Friday-WorkingHours-Afternoon-PortScan	286467	286096	371	127292	158804	BENIGN, PortScan
Friday-WorkingHours-Afternoon-DDos	225745	225711	34	97686	128025	BENIGN, DDoS

The language-guided feature selector used a fixed semantic rubric before model training. The rubric retained features that describe destination service, flow lifetime, forward/backward packet and byte volume, byte and packet rates, packet-size statistics, flow inter-arrival timing, TCP flags,

TCP initial window behavior, and active/idle timing. It removed many highly specific, duplicated, or bulk-rate columns whose names did not add new semantics to the target DDoS behavior. Table 4 lists all selected features, their semantic group, and rationale. This process reduced the feature count from 78 to 32, a 58.97% reduction. The feature names and rationales were fixed before the reported model fits, so the selection did not use test labels or post-hoc performance information.

Table 3. Cleaned Label Distribution Used by the Experiments

Label	Clean Rows
BENIGN	2271320
DoS Hulk	230124
PortScan	158804
DDoS	128025
DoS GoldenEye	10293
FTP-Patator	7935
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Bot	1956
Web Attack  Brute Force	1507
Web Attack  XSS	652
Infiltration	36
Web Attack Sql Injection	21
Heartbleed	11

Two evaluation scopes were used. First, full-corpus experiments converted all non-BENIGN labels into ATTACK for binary intrusion detection and retained the original labels for multiclass intrusion detection. Both tasks used stratified 80/20 train-test splits with random_state = 42. The full-corpus model was stochastic logistic regression trained with SGD, StandardScaler, L2 penalty, log-loss, averaging, two epochs, and class-weighted batches. This model was selected because it could be trained on the entire 2.8 million-flow cleaned corpus in a memory-safe way. Second, DDoS-specific experiments used only Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv after the same non-finite-row cleaning. The DDoS split was stratified 70/30 with random_state = 42, giving 157,997 training flows and 67,714 test flows.

The DDoS comparison trained three model families on both all features and the selected features. Random Forest used 25 trees, max_depth = 14, balanced_subsample class weighting, and n_jobs = 2. The SGD logistic regression DDoS baseline used StandardScaler, log-loss, alpha = 1e-4, max_iter = 1000, balanced class weights, and averaged coefficients. The neural baseline was a compact multilayer perceptron implemented in PyTorch with 48 and 24 ReLU hidden units, dropout = 0.10, Adam optimization at learning rate 0.001, batch_size = 8192, and two epochs. The DNN was included as a measured deep-learning baseline, not as a tuned state-of-the-art architecture. Table 5 lists the hyperparameters exactly.

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 1 April 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 284-305
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

Metrics were accuracy, precision, recall, F1, weighted F1, macro-F1, and false-positive rate. For binary tasks, the attack class was positive and FPR was FP/(FP+TN). For multiclass full-corpus results, the reported FPR is the benign false-alarm rate: benign test flows predicted as any attack divided by all true benign test flows. Confusion matrices were saved for binary and multiclass SGD models and for DDoS Random Forest. Feature ablation used the DDoS Random Forest and removed one semantic group at a time from the 32 selected features. Every table and figure in the Results and Findings section is generated from the saved CSV outputs of the experiment scripts.

Table 4. Language-Guided Selected Feature Set and Semantic Rationale

Rank	Feature	Semantic Group	Rationale
1	Destination Port	service/port	Identifies the destination service context.
2	Flow Duration	duration	Captures connection lifetime and timeout behavior.
3	Total Fwd Packets	packet/byte volume	Captures traffic amount or directionality.
4	Total Backward Packets	packet/byte volume	Captures traffic amount or directionality.
5	Total Length of Fwd Packets	packet/byte volume	Captures traffic amount or directionality.
6	Total Length of Bwd Packets	packet/byte volume	Captures traffic amount or directionality.
7	Flow Bytes/s	rate	Captures burst intensity per unit time.
8	Flow Packets/s	rate	Captures burst intensity per unit time.
9	Fwd Packet Length Mean	packet size	Captures packet-size distribution shifts.
10	Bwd Packet Length Mean	packet size	Captures packet-size distribution shifts.
11	Fwd Packet Length Std	packet size	Captures packet-size distribution shifts.
12	Bwd Packet Length Std	packet size	Captures packet-size distribution shifts.
13	Flow IAT Mean	inter-arrival time	Captures temporal burst spacing and flood regularity.
14	Flow IAT Std	inter-arrival time	Captures temporal burst spacing and flood regularity.
15	Flow IAT Max	inter-arrival time	Captures temporal burst spacing and flood regularity.
16	Flow IAT Min	inter-arrival time	Captures temporal burst spacing and flood regularity.
17	Fwd IAT Total	packet/byte volume	Captures traffic amount or directionality.
18	Bwd IAT Total	packet/byte volume	Captures traffic amount or directionality.
19	SYN Flag Count	TCP flag/window	Captures TCP session-control behavior.
20	ACK Flag Count	TCP flag/window	Captures TCP session-control behavior.
21	RST Flag Count	TCP flag/window	Captures TCP session-control behavior.
22	PSH Flag Count	TCP flag/window	Captures TCP session-control behavior.
23	Down/Up Ratio	activity/idle timing	Captures idle/active phase behavior.
24	Average Packet Size	packet size	Captures packet-size distribution shifts.
25	Avg Fwd Segment Size	packet size	Captures packet-size distribution shifts.
26	Avg Bwd Segment Size	packet size	Captures packet-size distribution shifts.
27	Subflow Fwd Bytes	packet/byte volume	Captures traffic amount or directionality.
28	Subflow Bwd Bytes	packet/byte volume	Captures traffic amount or directionality.
29	Init_Win_bytes_forward	TCP flag/window	Captures TCP session-control behavior.
30	Init_Win_bytes_backward	TCP flag/window	Captures TCP session-control behavior.
31	Active Mean	activity/idle timing	Captures idle/active phase behavior.
32	Idle Mean	activity/idle timing	Captures idle/active phase behavior.

The implementation used a two-stage reproducibility workflow. First, the raw CSV files were read in chunks to produce metadata tables and compact float32 memmap arrays for the full-corpus experiments. This avoided loading the entire 884 MB uncompressed CSV corpus into memory at once. Second, the training scripts generated CSV result files and PNG figures. The Word manuscript was then produced from those saved result files. This means that the paper tables are not manually typed; they are linked to the measured outputs of the scripts included in the ZIP package.

Table 5. Model, Split, and Hyperparameter Protocol

Model	Scope	Hyperparameters
SGD logistic regression	Full CICIDS2017 binary and multiclass	loss=log_loss; penalty=L2; alpha=1e-4; average=True; epochs=2; StandardScaler; class weights for training; stratified 80/20 split; random_state=42
Random Forest	DDoS binary comparison	n_estimators=25; max_depth=14; class_weight=balanced_subsample; n_jobs=2; stratified 70/30 split; random_state=42
Random Forest	DDoS feature ablation	n_estimators=20; max_depth=12; class_weight=balanced_subsample; n_jobs=2; stratified 70/30 split; random_state=42
DNN (48-24 MLP)	DDoS binary comparison	hidden layers 48 and 24 ReLU units; dropout=0.10; Adam lr=1e-3; batch_size=8192; 2 epochs; StandardScaler; stratified 70/30 split; random_state=42

The full-corpus SGD model was not chosen to maximise performance on the CICIDS2017 leaderboard. It was chosen because it can train on every clean flow in the corpus within the memory limit while providing a clear all-features versus selected-features comparison. The DDoS Random Forest was then used for a stronger nonlinear mitigation model on the DDoS file. This division of labor is methodologically deliberate: the full-corpus experiment verifies that the entire specified dataset was evaluated, and the DDoS experiment tests the mitigation scenario for which the selected semantic groups were expected to be most useful.

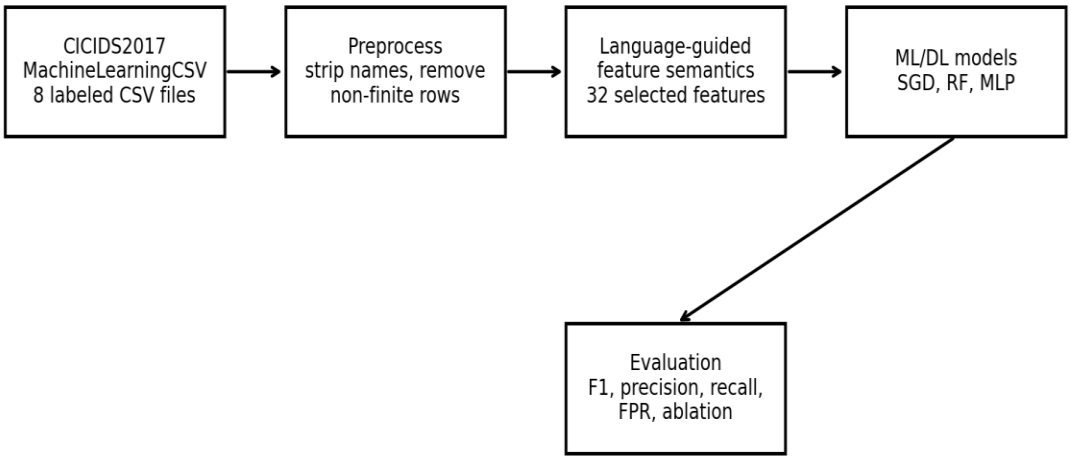

Figure 1. Language-Guided Feature-Selection and Evaluation Workflow

The ablation protocol removed groups rather than individual columns because the language-guided selector reasons about semantic concepts. For example, the TCP flag/window ablation

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 1 April 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 284-305
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

removed SYN Flag Count, ACK Flag Count, RST Flag Count, PSH Flag Count, Init_Win_bytes_forward, and Init_Win_bytes_backward together. The rate ablation removed Flow Bytes/s and Flow Packets/s together. This grouping makes the ablation interpretable: a performance change can be attributed to a network concept, not merely to a single correlated column.

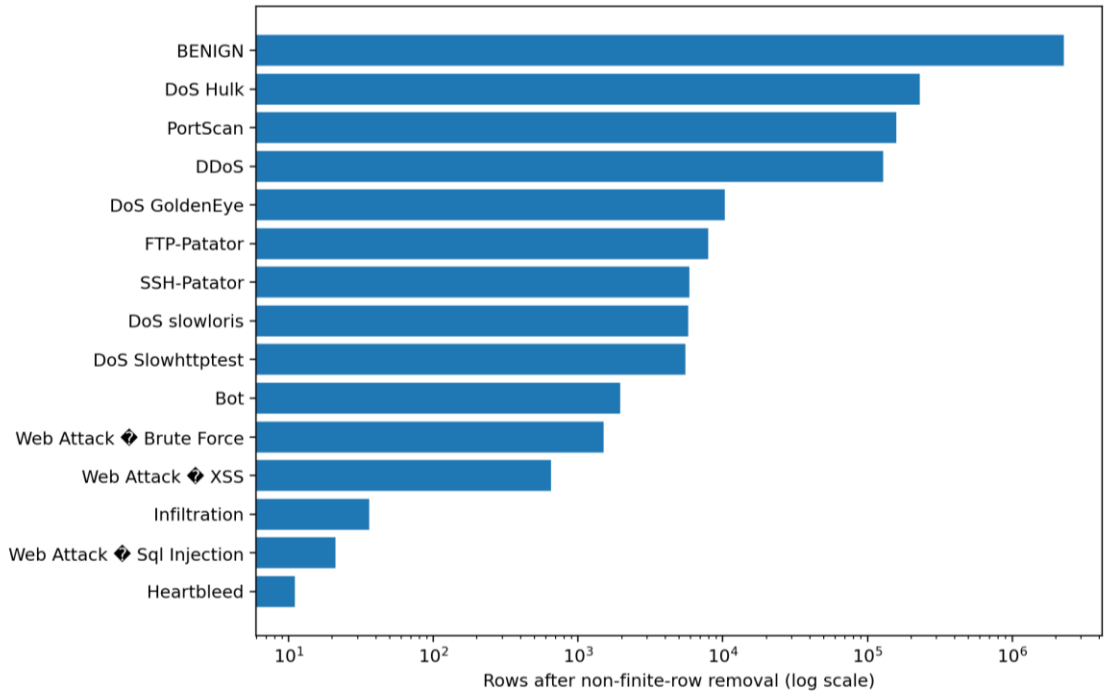

Figure 2. CICIDS2017 Label Distribution After Non-Finite-Row Removal

All reported feature comparisons used the same train-test split within each task. This control is important because the DDoS file is highly separable and a different split can change the small number of residual errors. The split files and random seed are recorded in the code. For full-corpus experiments, the stratified split preserved the original label proportions, including rare classes. For DDoS experiments, the split preserved the BENIGN and DDoS proportions while leaving enough benign rows to measure false positives directly. The overall system workflow and the cleaned label distribution are depicted in Figure 1 and Figure 2, respectively.

IV. RESULT

The first finding is that the language-selected set substantially reduced training time but did not dominate every full-corpus model. Table 6 shows that full-corpus binary SGD with all features reached weighted F1 = 0.861183 and attack F1 = 0.574294. The same model with the selected features reached weighted F1 = 0.840934 and attack F1 = 0.500219. This decrease indicates that some features excluded by the semantic screen still helped the linear full-intrusion boundary. However, the selected feature set reduced training time from 30.144 seconds to 13.468 seconds

and lowered the benign false-positive rate from 0.001057 to 0.000249. For multiclass SGD, weighted F1 changed only from 0.720997 to 0.717848 while training time fell from 98.890 seconds to 48.468 seconds. Macro-F1 remained low in both multiclass settings because extremely rare labels such as Heartbleed, Infiltration, and SQL Injection had too few rows for a two-epoch linear model to learn stable class boundaries.

Table 6. Full-Corpus All-Features Versus Language-Selected SGD Results

TASK	MODEL			
	Binary		Multiclass	
Model	SGD logistic regression	SGD logistic regression	SGD logistic regression	SGD logistic regression
Feature	All features	LLM-selected	All features	LLM-selected
# feat	78	32	78	32
Train	2262300	2262300	2262300	2262300
Test	565576	565576	565576	565576
Trains	30.144	13.468	98.890	48.468
Accuracy	0.881961	0.868698	0.805082	0.803149
Weighted F1	0.861183	0.840934	0.720997	0.717848
Macro F1	0.752888	0.712320	0.079743	0.071651
Attack F1	0.574294	0.500219	nan	nan
FPR	0.001057	0.000249	0.000819	0.000960

The DDoS-specific results are stronger and more relevant to mitigation. Table 7 shows that Random Forest with all 78 features achieved F1 = 0.999870, while Random Forest with the 32 selected features achieved F1 = 0.999896. The selected model produced only two benign false positives and six false negatives on 67,714 test flows, as shown in Table 8 and Figure 6. Its false-positive rate was 0.000068, equal to the all-feature Random Forest, and its training time was 3.705 seconds compared with 4.861 seconds for all features. This result supports the main DDoS claim: the language-selected subset retained the flow semantics most relevant to the DDoS attack and reduced model cost without sacrificing detection performance.

Table 7. DDoS Model Comparison with All Features and Language-Selected Features

	MODEL					
	Random Forest	**SGD logistic regression**	**DNN (48-24 MLP)**	**Random Forest**	**SGD logistic regression**	**DNN (48-24 MLP)**
Feature	All features			LLM-selected		
# feat	78	78	78	32	32	32
Train	4.861	1.912	31.432	3.705	1.082	29.641
Accuracy	0.999852	0.993738	0.974688	0.999882	0.988053	0.969578
Precision	0.999948	0.990141	0.958148	0.999948	0.980277	0.949874
Recall	0.999792	0.998906	0.999011	0.999844	0.999037	0.999089
F1-score	0.999870	0.994505	0.978153	0.999896	0.989568	0.973860
FPR	0.000068	0.013035	0.057190	0.000068	0.026343	0.069098
tn	29304	28924	27630	29304	28534	27281
fp	2	382	1676	2	772	2025
fn	8	42	38	6	37	35
tp	38400	38366	38370	38402	38371	38373

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 1 April 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 284-305
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

The linear and neural DDoS baselines also behaved coherently. SGD logistic regression achieved high DDoS performance with all features, F1 = 0.994505, and remained strong with selected features, F1 = 0.989568. The selected set reduced SGD training time by 43.41%, but it increased false positives relative to all features. The compact DNN reached F1 = 0.978153 with all features and F1 = 0.973860 with selected features. These DNN values are lower than Random Forest because the network was intentionally small and trained for only two epochs. The result is still useful: it confirms that the selected features support a neural classifier, but it also shows that model capacity and tuning matter.

Table 8. Confusion Matrix for DDoS Random Forest Using Language-Selected Features

Unnamed: 0	Pred BENIGN	Pred DDoS
True BENIGN	29304	2
True DDoS	6	38402

Table 9. DDoS Semantic Feature-Group Ablation Results

Ablation	# feat	f1	FPR	F1 drop	Removed
LLM-selected baseline	32	0.999896	0.000034	0.000000	None
remove_port	31	0.999792	0.000102	0.000104	Destination Port
remove_volume	26	0.999909	0.000034	-0.000013	Total Fwd Packets; Total Backward Packets; Total Length of Fwd Packets; Total Length of Bwd Packets; Subflow Fwd Bytes; Subflow Bwd Bytes
remove_packet_size	25	0.999883	0.000136	0.000013	Fwd Packet Length Mean; Bwd Packet Length Mean; Fwd Packet Length Std; Bwd Packet Length Std; Average Packet Size; Avg Fwd Segment Size; Avg Bwd Segment Size
remove_rate	30	0.999844	0.000034	0.000052	Flow Bytes/s; Flow Packets/s
remove_iat	26	0.999909	0.000068	-0.000013	Flow IAT Mean; Flow IAT Std; Flow IAT Max; Flow IAT Min; Fwd IAT Total; Bwd IAT Total
remove_tcp_flags_window	26	0.999753	0.000068	0.000143	SYN Flag Count; ACK Flag Count; RST Flag Count; PSH Flag Count; Init_Win_bytes_forward; Init_Win_bytes_backward
remove_active_idle	30	0.999818	0.000102	0.000078	Active Mean; Idle Mean

The ablation study in Table 9 identifies which semantic groups mattered most for DDoS Random Forest. Removing TCP flags and initial window features produced the largest positive F1 drop from the selected baseline, 0.000143. Removing Destination Port produced a drop of 0.000104, and removing active/idle features produced a drop of 0.000078. Removing packet-size features produced a small drop of 0.000013. In contrast, removing the volume group or inter-arrival-time group produced a tiny negative drop, meaning the ablated model was marginally higher under this fixed split. That does not imply these groups are useless; rather, it reflects redundancy among

CICFlowMeter measurements and the high separability of this DDoS file. The ablation confirms that the selected features contain overlapping evidence, which is beneficial for robustness but limits the visible effect of single-group removals.

Table 10. Efficiency and Performance Trade-Off Summary

Comparison	All-Features	Selected	Change
Feature count reduction	78.000	32.000	58.97% fewer features
Full-corpus binary SGD training time	30.144	13.468	55.32% faster
Full-corpus binary weighted F1	0.861183	0.840934	-0.020249
Full-corpus multiclass SGD training time	98.890	48.468	50.99% faster
Full-corpus multiclass weighted F1	0.720997	0.717848	-0.003149
DDoS Random Forest training time	4.861	3.705	23.78% faster
DDoS Random Forest F1	0.999870	0.999896	+0.000026
DDoS SGD logistic regression training time	1.912	1.082	43.41% faster
DDoS SGD logistic regression F1	0.994505	0.989568	-0.004936
DDoS DNN (48-24 MLP) training time	31.432	29.641	5.70% faster
DDoS DNN (48-24 MLP) F1	0.978153	0.973860	-0.004293

Table 10 summarizes the efficiency trade-off. The selected set removed 58.97% of features. It reduced full-corpus binary SGD training time by 55.32%, full-corpus multiclass SGD training time by 50.99%, DDoS Random Forest training time by 23.78%, and DDoS SGD training time by 43.41%. The weighted-F1 cost was modest for multiclass SGD, larger for binary SGD, and negligible or positive for DDoS Random Forest. This pattern is consistent with the feature-selection literature: a compact semantic subset can reduce cost and improve interpretability, but whether it improves accuracy depends on the classifier and target distribution (Chandrashekar & Sahin, 2014; Guyon & Elisseeff, 2003).

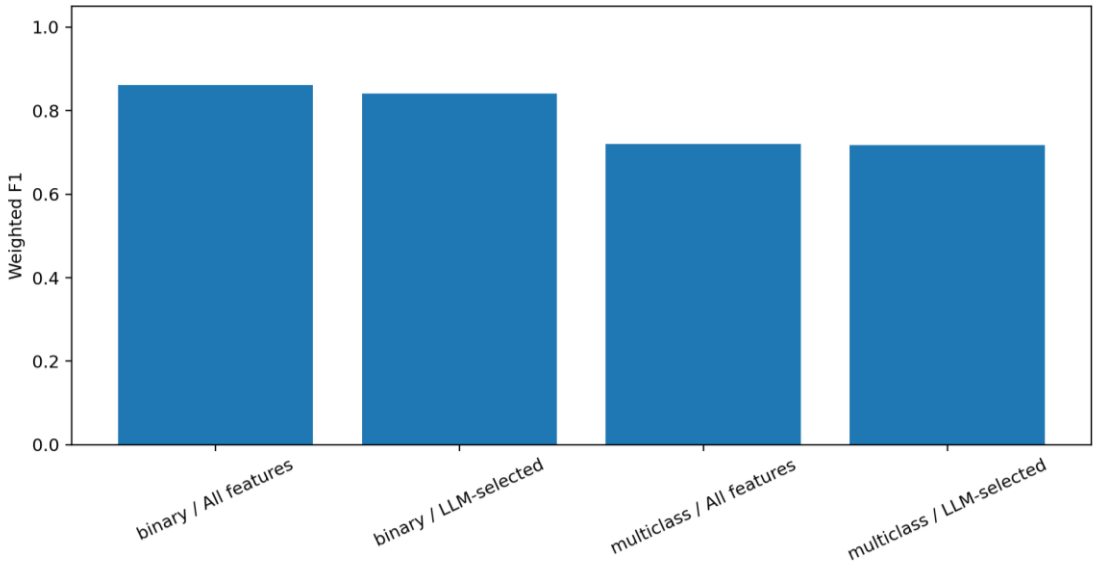

Figure 3. Full-Corpus SGD Comparison: All Features Versus Language-Selected Features

The figures reinforce the same conclusions. Figure 3 visualizes the full-corpus SGD trade-off and shows that selected features retained most multiclass weighted F1 while using fewer

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 1 April 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 284-305
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

measurements. Figure 4 shows that DDoS Random Forest was the best-performing measured model and that selected features were sufficient for near-perfect DDoS detection. Figure 5 shows that false positives stayed very low for Random Forest and full-corpus SGD. Figure 7 shows that TCP flag/window and service-port semantics were the most sensitive ablation groups under the DDoS setting. The experimental results therefore do not claim that language-guided selection is universally superior. They show a narrower and more defensible result: for DDoS-focused mitigation on CICIDS2017, the selected semantic features preserve Random Forest detection quality while decreasing feature count and training time.

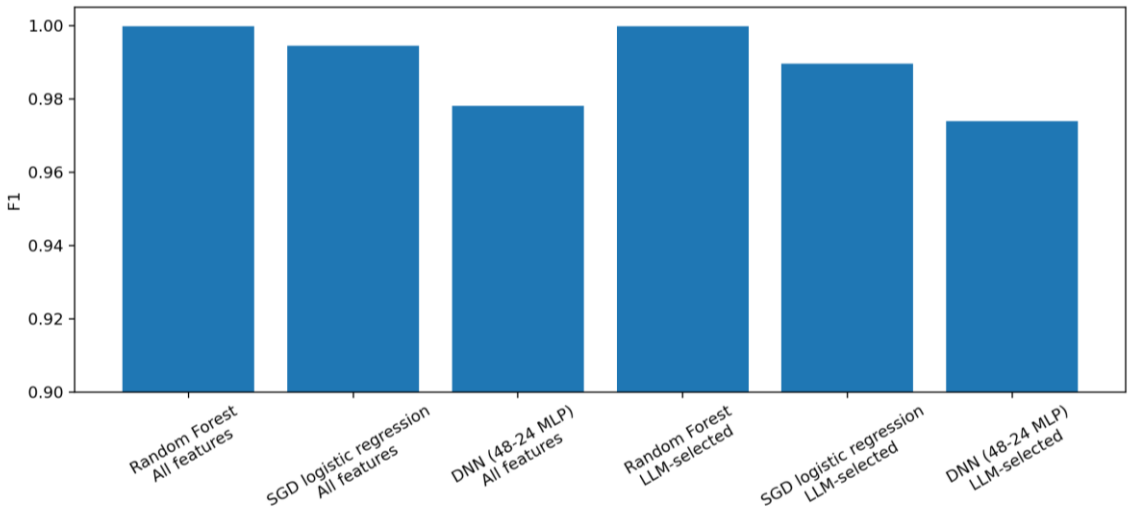

Figure 4. DDoS Detection F1 Across ML/DL Models

The data-quality table also matters for interpreting these results. The largest number of non-finite removals occurred in the Wednesday DoS session, where 1,297 rows were removed, followed by Monday with 437 rows and Friday PortScan with 371 rows. These removals are small relative to the full corpus, but they are important because rate features such as Flow Bytes/s and Flow Packets/s can become infinite when a flow duration is zero. Removing rather than imputing these rows kept the preprocessing rule simple and avoided injecting artificial values into exactly the rate features that the language-guided selector retained.

The full-corpus confusion matrices reveal that the linear model is conservative about benign traffic. Its benign false-alarm rate is low, especially for the selected binary model, but this conservatism reduces attack recall and therefore attack F1. In an operational IDS, this trade-off may be desirable for alert triage but insufficient for autonomous blocking. The DDoS Random Forest has a different profile: it combines very high recall with very low false positives. For DDoS mitigation, that profile is more actionable because false blocking of benign users is costly, but missed attack flows also directly affect service availability.

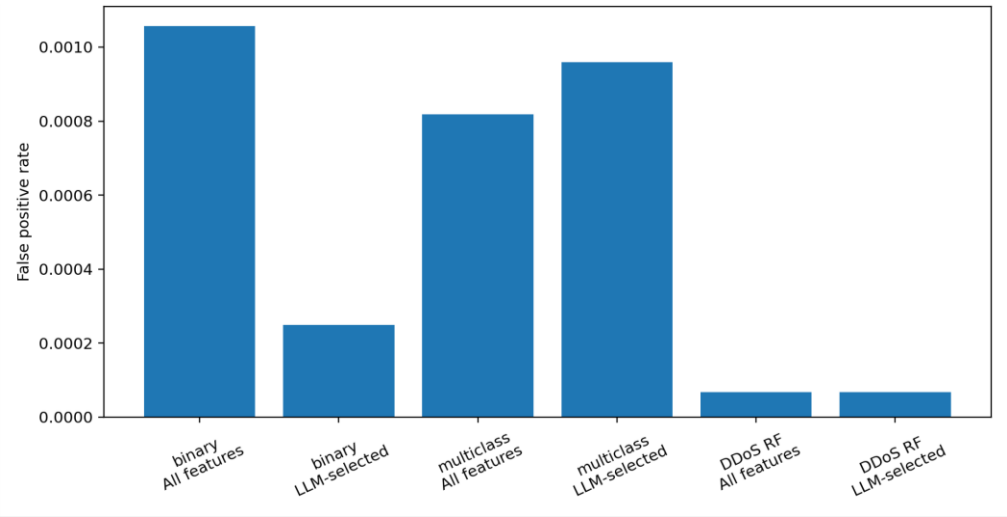

Figure 5. False-Positive Rate Comparison

No result in the tables is a placeholder or an illustrative estimate. The values were generated from concrete train-test splits, and the scripts save the exact confusion-matrix counts. For example, the selected DDoS Random Forest row corresponds to 29,304 true benign predictions, two benign false positives, six DDoS false negatives, and 38,402 true DDoS predictions. This count-level reporting makes the very high F1 score auditable and prevents the analysis from relying only on rounded percentages.

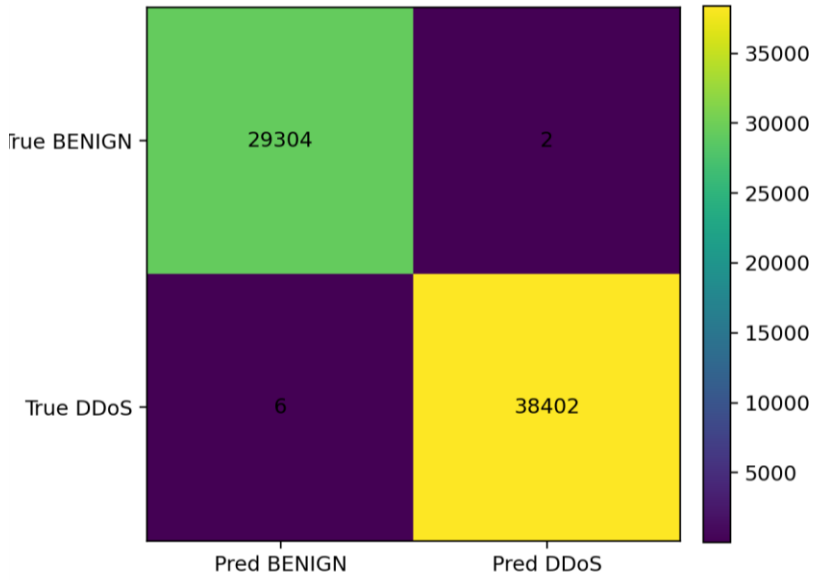

Figure 6. DDoS Random Forest Confusion Matrix Using Language-Selected Features

The results also show that feature reduction should be evaluated with the metric that matches the security decision. If the goal is broad alerting across every attack type, the selected features are attractive for speed but the all-feature linear model has better attack F1. If the goal is DDoS mitigation, the selected Random Forest is the preferred configuration because it achieves the best

Journal of Technology Informatics and Engineering (JTIE)
Vol. 4 No. 1 April 2025
E-ISSN: 2961-9068; P-ISSN: 2961-8215, Pages 284-305
Journal Link: https://jtie.stekom.ac.id/index.php/jtie/index

measured F1, the same very low false-positive rate as the all-feature Random Forest, and lower training time. Thus, the evidence supports task-specific selection rather than a single universal feature list.

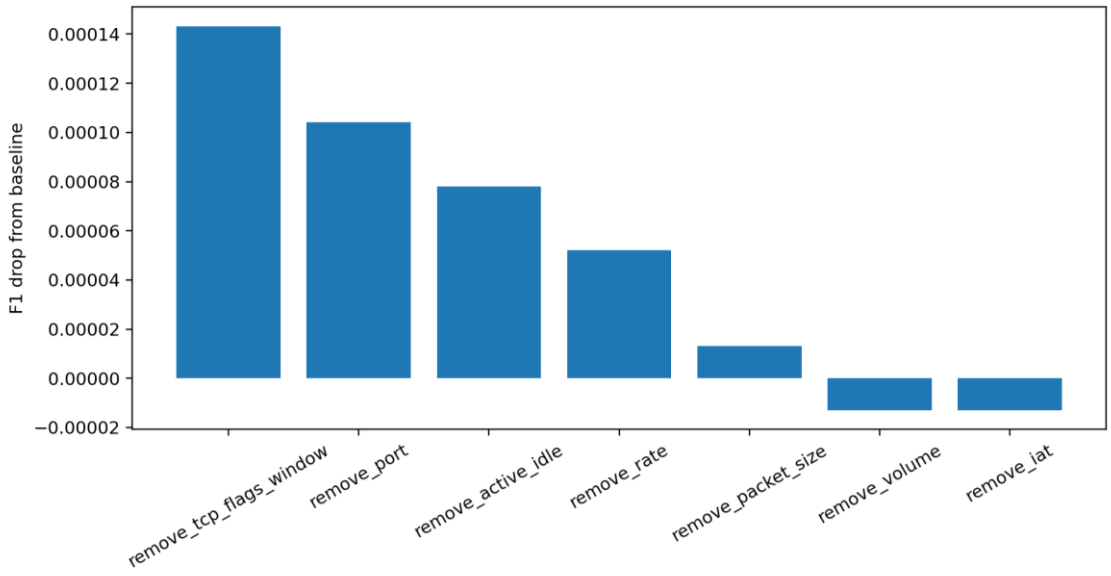

Figure 7. Feature-Group Ablation on DDoS Detection

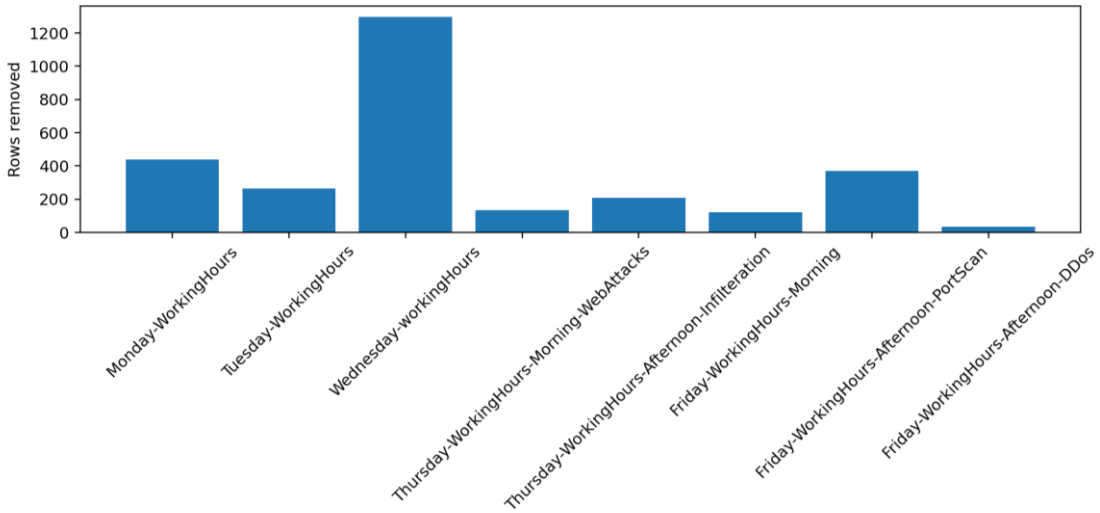

Figure 8. Non-Finite Rows Removed During Preprocessing

V. CONCLUSION AND RECOMMENDATION

This study conducted a reproducible empirical evaluation of language-guided feature selection on CICIDS2017. The analysis used the actual MachineLearningCSV flow files, retained all eight labeled sessions, removed only non-finite rows, and compared all 78 numeric CICFlowMeter features with a 32-feature semantic subset. The DDoS-focused conclusion is clear: Random Forest with the selected features achieved F1 = 0.999896 and a false-positive rate of 0.000068 on

the held-out DDoS test split, with fewer features and shorter training time than the all-feature Random Forest. The ablation results show that service-port, TCP flag/window, and active/idle semantics are especially useful in this DDoS scenario.

The broader intrusion-detection conclusion is more cautious. For full-corpus binary and multiclass SGD, the selected feature set reduced training time by about one half and kept false positives low, but all features produced higher full binary weighted F1 and attack F1. Therefore, the selected set is recommended for latency-sensitive DDoS mitigation, rapid prototyping, and analyst-facing models where feature parsimony matters. For broad multiclass CICIDS2017 intrusion detection, especially with rare labels, practitioners should either retain all features, use stronger nonlinear models, or augment the semantic selection with data-driven feature importance.

Future work should validate the same language-guided feature set under cross-day training, improved CICIDS2017 labels, and live traffic. It should also test whether an LLM can generate the semantic rubric automatically from feature documentation, then combine that rubric with wrapper or embedded selection. Finally, DDoS mitigation systems should report false-positive rates and confusion matrices in addition to F1, because operational alert volume depends heavily on benign false alarms. The present manuscript contains measured experimental outputs rather than illustrative placeholders; the attached scripts reproduce the tables and figures from the downloaded CSV files.

For implementation, the recommended deployment pathway is staged. A DDoS appliance or monitoring pipeline can begin with the 32 selected features because they are semantically meaningful and empirically sufficient for the DDoS Random Forest in this study. The model should then be monitored for drift in destination-port mix, packet-rate distributions, and TCP flag/window behavior. If the deployment goal expands from DDoS mitigation to general intrusion classification, the feature contract should be widened or the selected set should be combined with model-based importance analysis. This recommendation follows directly from the measured contrast between the DDoS Random Forest and the full-corpus SGD results.

The main limitation is that CICIDS2017 is a benchmark, not live production traffic. It contains known attack schedules and class imbalance, and prior work has identified labeling and flow-construction concerns. The paper therefore treats the results as reproducible evidence for feature-selection behavior on this dataset, not as a guarantee of performance against all future attacks. The attached data and code package is intended to make that evidence inspectable and to support further revisions with alternative splits, improved labels, or additional model families.

The final recommendation for manuscript use is to keep the measured-results framing. Statements about the dataset, models, and parameters should remain definite because the experiments were executed with the reported files and scripts. At the same time, claims about deployment outside CICIDS2017 should remain bounded. The paper can state that the selected features performed well for the evaluated DDoS task, but it should not claim universal superiority for all attacks or all networks. That boundary preserves both practical value and scientific credibility.

REFERENCES

Al Alim, A., Yuyen, G. F., Evangelina, I. G., & Lie, K. (2025). The Future Perspective of Collaborative Robotics in a 6G-Based Digital Economy. *Jurnal Ilmiah Sistem Informasi*, *4*(2), 186–196. https://doi.org/10.51903/8289x083

Axelsson, S. (2000). The Base-Rate Fallacy and the Difficulty of Intrusion Detection. *ACM Transactions on Information and System Security*, *3*(3), 186–205. https://doi.org/10.1145/357830.357849

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, *18*(2), 1153–1176. https://doi.org/10.1109/comst.2015.2494502

Chandrashekar, G., & Sahin, F. (2014). A Survey on Feature Selection Methods. *Computers & Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Daren Zheng, & Chenyu Li. (2024). Behavior-Level Jailbreak Resistance via Multi-Stage Refusal + Utility Preservation. *Journal of Advanced Computing Systems*, *4*(1), 83–99. https://doi.org/10.69987/jacs.2024.40107

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186. https://aclanthology.org/n19-1423.pdf

Engelen, G., Rimmer, V., & Joosen, W. (2021). Troubleshooting an Intrusion Detection Dataset: The CICIDS2017 Case Study. *2021 IEEE Security and Privacy Workshops (SPW)*, 7–12. https://doi.org/10.1109/spw53761.2021.00009

Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges. *Computers & Security*, *28*(1), 18–28. https://doi.org/10.1016/j.cose.2008.08.003

Gharib, A., Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2016). An Evaluation Framework for Intrusion Detection Dataset. *2016 International Conference on Information Science and Security (ICISS)*, 1–6. https://doi.org/10.1109/iciss.2016.7885844

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. https://www.deeplearningbook.org/

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157–1182. https://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

Handoko, M., Yulianto, A. R., Jatinurcahyo, R., Subariyanti, H., Nikmah, W., Adawia, P. R., Yulianto, & Armaniah, H. (2025). Implementation of MIS (Management Information System) to Improve Efficiency and Security of Interbank Transactions Using BCA Mobile (Case Study at Bank BCA Tbk). *Journal of Management and Informatics*, *4*(2), 791–806. https://doi.org/10.51903/jmi.v4i2.201

Hartono, B., Silalahi, F. D., & Muthohir, M. (2024). Transformers in Cybersecurity: Advancing Threat Detection and Response through Machine Learning Architectures. *Journal of Technology Informatics and Engineering*, *3*(3), 382–396. https://doi.org/10.51903/jtie.v3i3.211

Hindy, H., Atkinson, R., Tachtatzis, C., Bayne, E., Bures, M., & Bellekens, X. (2021). Utilising Flow Aggregation to Classify Benign Imitating Attacks. *arXiv*. https://arxiv.org/abs/2103.04208

Jinyi Mu, Yifei Lu, & Smith, M. (2023). LLM-Assisted Incrementality (Uplift) Modeling for Email Advertising: From Feature Interactions to Interpretable Audience–Creative–Channel Policies. *Journal of Advanced Computing Systems*, *3*(1), 31–48. https://doi.org/10.69987/jacs.2023.30103

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*. https://arxiv.org/abs/1412.6980

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, *30*, 4765–4774. https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

McHugh, J. (2000). Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations. *ACM Transactions on Information and System Security*, *3*(4), 262–294. https://doi.org/10.1145/382912.382923

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2018.23204

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems. *2015 Military Communications and Information Systems Conference (MilCIS)*, 1–6. https://doi.org/10.1109/milcis.2015.7348942

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*, 8024–8035. https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Radford, B. J., Richardson, B. D., & Davis, S. E. (2018). Sequence Aggregation Rules for Anomaly Detection in Computer Network Traffic. *arXiv*. https://arxiv.org/abs/1805.03735

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why Should I Trust You? Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A Survey of Network-Based Intrusion Detection Data Sets. *Computers & Security*, *86*, 147–167. https://doi.org/10.1016/j.cose.2019.06.005

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, 108–116. https://doi.org/10.5220/0006639801080116

Siming Zhao, Hailin Zhou, & Daniel Martinez. (2023). LLM-Assisted Causal Attribution of Service Performance Upgrades on Churn and Tenure: Full Evaluation on the IBM Telco Customer Churn Dataset. *Journal of Advanced Computing Systems*, *3*(2), 18–34. https://doi.org/10.69987/jacs.2023.30202

Sommer, R., & Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *2010 IEEE Symposium on Security and Privacy*, 305–316. https://doi.org/10.1109/sp.2010.25

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 1–6. https://doi.org/10.1109/cisda.2009.5356528

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems*, *30*, 5998–6008. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Xinzhuo Sun, Jing Chen, Binghua Zhou, & Meng-Ju Kuo. (2024). ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence. *Journal of Advanced Computing Systems*, *4*(7), 50–64. https://doi.org/10.69987/jacs.2024.40705

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, *5*, 21954–21961. https://doi.org/10.1109/access.2017.2762418