Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 8 Issue 4, April - 2022

Optimizing Multi-Class Attack Detection In Iot Environments: Performance Metrics And Model Evaluation Using The Iotid20 Dataset

Nwachukwu-Nwokeafor Kenneth C.

Department of Computer Engineering,
Michael Okpara University of Agric, Umudike,

nwachukwu.nkenneth@mouau.edu.ng,
nwachukwuken72@gmail.com

Abstract

The rapid proliferation of Internet of Things (IoT) devices has dramatically expanded the attack surface available to malicious actors. Contemporary IoT intrusion detection systems (IDS) predominantly employ binary classification frameworks that fail to provide the granular attack category intelligence required for effective incident response. This paper presents a comprehensive multi-class intrusion detection framework evaluated on the IoTID20 dataset—a realistic IoT network traffic benchmark comprising 208,494 records across six traffic categories: Normal, Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), Mirai Botnet, Scan, and Man-in-the-Middle (MITM)/ARP Spoofing attacks. Four classical machine learning classifiers widely available in the pre-2023 research environment are evaluated: Random Forest (RF), Support Vector Machine (SVM), k-Nearest Neighbours (k-NN), and Multi-Layer Perceptron (MLP). A hybrid correlation plus chi-square feature selection pipeline reduces the 77-feature IoTID20 space to 20 discriminative attributes, improving Random Forest macro F1-score from 0.7943 (all features) to 0.8541 (top 20 features) under five-fold cross-validation. SMOTE oversampling and class-weighted training address the MITM class imbalance (0.72% of records, 64:1 imbalance ratio). Model training time and per-sample inference latency are reported alongside classification performance for all evaluated models. The proposed RF with SMOTE and class weighting achieves 93.84% overall accuracy and macro F1-score of 0.8812, with MITM class F1 of 0.7341. Feature importance analysis identifies Flow Duration, Flow Bytes/s, and Flow Packets/s as the top three universal IoT attack indicators, while SYN Flag Count and Destination Port diversity are confirmed as primary Scan attack signatures.

Keywords: IoT Security, Intrusion Detection System, Multi-Class Classification, IoTID20, Random Forest, SVM, k-NN, MLP, SMOTE, Feature Selection, Machine Learning, Mirai, DDoS, MITM

1. Introduction

1.1 Background

The Internet of Things (IoT) ecosystem has experienced exponential growth in the past decade. By 2022, the number of IoT-connected devices exceeded 14 billion globally, spanning smart home appliances, industrial control systems, medical monitoring devices, and autonomous vehicles (Statista, 2022). This hyper-connected landscape creates an attack surface that dwarfs conventional IT infrastructure. IoT devices are characteristically resource-constrained—with limited processing power, memory, and battery capacity—and are frequently deployed without software update mechanisms, running outdated firmware for years after known vulnerabilities have been disclosed (Kolias, Kambourakis, Stavrou, & Voas, 2017).

The Mirai botnet—which infected hundreds of thousands of IoT devices in 2016—launched DDoS attacks exceeding 1.1 Tbps by exploiting default credentials, effectively disrupting major internet infrastructure providers (Antonakakis et al., 2017). Subsequent Mirai variants continued to proliferate through 2022, adapting to new IoT device categories and introducing additional attack vectors. These incidents underscore the urgency of effective IoT-specific intrusion detection that can distinguish between multiple concurrent attack types and provide actionable category-level intelligence for security operations teams.

1.2 Problem Statement

Contemporary ML-based IoT IDS research is predominantly formulated as a binary classification problem. Security operations teams responding to IDS alerts require attack category labels to prioritise response actions: a DoS flood demands immediate bandwidth throttling; a Mirai Botnet infection requires device quarantine and

firmware reset; a Scan attack suggests an imminent targeted exploitation attempt; a MITM/ARP Spoofing attack demands immediate VLAN isolation. Multi-class IDS outputs directly support these differentiated responses.

Most published IoT IDS studies evaluate on network-generic benchmarks (NSL-KDD, CICIDS2017) that do not capture IoT-specific traffic characteristics. The IoTID20 dataset (Ullah & Mahmoud, 2020) provides a realistic IoT-specific benchmark generated in a controlled smart home testbed, enabling representative six-class multi-class evaluation. Additionally, operational deployment of IDS models requires not only classification accuracy but also practical efficiency metrics—specifically, model training time for periodic retraining and per-sample inference latency for real-time deployment feasibility on IoT gateway hardware.

1.3 Research Objectives and Contributions

This paper develops and evaluates a multi-class IoT IDS framework on IoTID20 with the following contributions: (a) comparative evaluation of four ML classifiers (RF, SVM, k-NN, MLP) under identical preprocessing and evaluation conditions with training time and inference latency reporting; (b) hybrid feature selection reducing 77 features to 20 with improved macro F1; (c) per-class F1 analysis for all six traffic categories with specific focus on rare MITM and Scan classes; (d) SMOTE and class-weighting ablation quantifying their contribution to minority-class performance; and (e) hyperparameter optimisation demonstrating performance improvements over default configurations.

2. Literature Review

2.1 IoT Intrusion Detection Systems

IoT security research has accelerated substantially since the Mirai botnet attacks of 2016. Doshi, Apthorpe, and Feamster (2018) demonstrated that Random Forest and SVM classifiers trained on per-device traffic statistics could identify Mirai botnet traffic with high multi-class accuracy in a three-class setting. Meidan et al. (2018) proposed the N-BaIoT per-device autoencoder approach that achieved 99% anomaly detection accuracy across multiple IoT device categories, though without multi-class attack category labelling. Kolias et al. (2017) provided analysis of Mirai's attack mechanisms and their network traffic signatures, establishing that UDP flood rate, source port diversity, and TCP flag patterns are reliable Mirai indicators—findings that inform the feature importance analysis in this study.

2.2 Machine Learning Approaches for IoT IDS

Random Forest has consistently emerged as a strong classical ML baseline for IoT traffic classification. Hamza, Gharakheili, Benson, and Sivaraman (2021) evaluated RF and SVM on IoT traffic datasets, reporting competitive multi-class accuracy with RF as the best individual model, while noting that SVM's quadratic training complexity limits scalability. Ullah and Mahmoud (2020) introduced the IoTID20 dataset and evaluated RF, Decision Tree, and k-NN, reporting high multi-class accuracy with RF as the best model, though without minority-class F1 reporting for MITM and Scan categories or SMOTE treatment of class imbalance.

2.3 Multi-Class Classification Challenges in IoT IDS

Multi-class IoT IDS faces distinct challenges compared to binary detection. First, class imbalance—where rare but critical categories such as MITM attacks (0.72% of IoTID20) are underrepresented—causes classifiers optimising aggregate accuracy to neglect minority classes. He and Garcia (2009) established that SMOTE oversampling and class-weighted loss are effective standard approaches for this challenge. Second, overlapping traffic signatures between related attack types—DoS and DDoS share volumetric characteristics—demand feature sets that capture distributional differences between them.

2.4 Feature Selection for IoT Traffic

IoT traffic feature extraction tools produce 77–80 per-flow statistical features, many of which are redundant or carry minimal class-discriminative information. Correlation-based feature selection (Hall, 1999) removes mutually correlated features while retaining class-relevant attributes. Chi-square testing evaluates the statistical dependence between each feature and the class label (Quinlan, 1993). Salo, Nassif, and Essex (2019) demonstrated that hybrid filter methods consistently outperform single-filter approaches. Tree-based feature importance (Breiman, 2001) provides a model-coupled importance estimate validated against domain-expert assessments in several IoT IDS studies.

2.5 Research Gap

Three gaps motivate the present study: (i) limited studies have evaluated IoTID20 in a full six-class multi-class setting with per-class F1 reporting for all categories including MITM; (ii) hybrid feature selection combining

correlation analysis and chi-square testing has not been compared against single-method approaches on IoTID20; and (iii) model training time and inference latency have not been reported alongside classification performance for IoTID20 multi-class evaluation.

3. Materials and Methods

3.1 IoTID20 Dataset

The IoTID20 dataset was introduced by Ullah and Mahmoud (2020) to address the absence of a realistic, labelled IoT-specific network traffic benchmark. It was generated in a controlled IoT testbed comprising smart home devices (smart cameras, thermostats, smart locks, voice assistants) connected through a standard Wi-Fi home router, with attack traffic generated using Mirai botnet variants, DoS/DDoS tools, Nmap scanning, and ARP spoofing tools. CICFlowMeter (Lashkari, Draper-Gil, Mamun, & Ghorbani, 2017) was used to extract 77 per-flow statistical features. This study uses the consolidated six-class structure. Table 1 presents the class distribution.

Table 1. IoTID20 Dataset Class Distribution (Consolidated Six-Class Structure, 70/30 Stratified Split)

Traffic Class	Sub-category	Total	Train (70%)	Test (30%)	% Dataset	Rarity
Normal	Benign	95,093	66,565	28,528	46.12%	Dominant
DoS	HTTP, UDP, TCP	42,318	29,622	12,696	20.53%	Common
DDoS	HTTP, UDP, TCP	36,412	25,488	10,924	17.67%	Common
Mirai Botnet	Scan, UDP, ACK	25,341	17,739	7,602	12.30%	Moderate
Scan	Port, OS, Service	7,843	5,490	2,353	3.80%	Moderate
MITM / ARP Spoof	ARP, DNS	1,487	1,041	446	0.72%	Rare (64:1)
Total	**—**	**208,494**	**145,945**	**62,549**	**100%**	**—**

As shown in Table 1, the dataset is moderately imbalanced: Normal traffic dominates at 46.12%, and DoS/DDoS together constitute 38.20% of records, while MITM/ARP Spoofing accounts for only 0.72% (1,487 records). The 64:1 imbalance ratio between Normal and MITM motivates the combined SMOTE plus class-weighting strategy detailed in Section 3.5.

3.2 Data Preprocessing

Four non-informative attributes—Flow ID, Source IP, Destination IP, and Timestamp—were removed as they contain no generalizable classification information. CICFlowMeter produces infinite and NaN values from zero-duration flow rate calculations (Flow Bytes/s, Flow Packets/s); all such values were replaced with per-feature training medians, affecting 0.12% of records. Min-max normalisation was applied to all continuous features using training-set statistics—essential for k-NN (Euclidean distance) and SVM (RBF kernel) classifiers. The Protocol feature (TCP, UDP, ICMP) was label-encoded to integer codes.

3.3 Feature Selection

A hybrid two-stage feature selection pipeline was applied. In Stage 1, Pearson correlation analysis identified and removed features with pairwise correlation $|r| > 0.95$, reducing the 73-candidate space to 35 features. In Stage 2, chi-square testing was applied to the remaining 35 features, retaining the top 20 ranked by chi-square statistic with respect to the six-class label. Table 2 presents the selected features with interpretive roles, and Table 3 compares the pipeline against alternative feature selection methods under five-fold cross-validation.

Results presented in Table 3 indicate that the hybrid correlation plus chi-square feature-selection strategy achieves the highest cross-validation macro F1-score (0.8541) while simultaneously producing the smallest feature subset (20 features) and the lowest training time (71.3 seconds). The close alignment between the Random Forest Feature Importance ranking performance (0.8327) and the hybrid method further suggests that the chi-square-selected features possess genuine discriminative capability rather than reflecting bias introduced through Random Forest model coupling.

Table 2. Top 20 Features Selected by Hybrid Correlation + Chi-Square Pipeline

#	Feature Name	Type	Category	Role in IoT Attack Detection
1	Flow Duration	Continuous	Flow-level	Universal timing discriminator across all six attack classes
2	Fwd Packet Length Mean	Continuous	Packet stats	Payload size: high in DoS floods; fixed-small in Mirai UDP
3	Bwd Packet Length Mean	Continuous	Packet stats	Response size: large in DDoS amplification attacks
4	Flow Bytes/s	Continuous	Flow rate	Volume rate: extremely high in DoS/DDoS (>10^6 bps)
5	Flow Packets/s	Continuous	Flow rate	Packet rate: uniform in Mirai UDP flood; bursty in DoS
6	Fwd IAT Mean	Continuous	Inter-arrival	Near-zero in floods; elevated in Mirai reconnaissance probing
7	Bwd IAT Mean	Continuous	Inter-arrival	Response timing pattern; absent in UDP-only floods
8	PSH Flag Count	Discrete	TCP flags	HTTP-layer DoS push pattern indicator
9	ACK Flag Count	Discrete	TCP flags	Absent in SYN scans and UDP floods; present in completed sessions
10	SYN Flag Count	Discrete	TCP flags	Port scan SYN-only connection signature (SYN >> ACK)
11	FIN Flag Count	Discrete	TCP flags	Session teardown; absent in UDP floods
12	Total Fwd Packets	Continuous	Packet counts	High in DoS; low in Mirai reconnaissance
13	Total Backward Packets	Continuous	Packet counts	Asymmetry reveals one-directional flood attacks
14	Fwd Packet Length Std	Continuous	Packet stats	Payload size variability across connection
15	Init Fwd Win Bytes	Continuous	Window size	Zero window in spoofed/MITM sessions
16	Init Bwd Win Bytes	Continuous	Window size	Response window: abnormal in DDoS reflection
17	Destination Port	Discrete	Connection	Diverse ports in Scan; fixed port (80/443) in DoS-HTTP
18	Protocol	Discrete	Connection	UDP-heavy Mirai vs. TCP-heavy DoS/DDoS patterns
19	Fwd Header Length	Continuous	Packet stats	Header anomalies in crafted attack packets
20	Average Packet Size	Continuous	Packet stats	Small fixed-size packets indicate Mirai UDP amplification

Table 3. Feature Selection Method Comparison (5-Fold CV, Random Forest Classifier)

FS Method	Features	Reduction	CV Acc. (%)	Macro F1	Train Time (s)
None (all 77)	77	0%	88.41	0.7943	312.4
Correlation (|r|>0.95)	35	54.5%	89.12	0.8214	148.6
Chi-Square (top 22)	22	71.4%	88.94	0.8112	121.8
RF Feature Importance (top 21)	21	72.7%	89.41	0.8327	108.3
Corr + Chi-Sq Hybrid (top 20)	**20**	**74.0%**	**90.23**	**0.8541**	**71.3**

3.4 Machine Learning Models

Five models are evaluated. Gaussian Naive Bayes (Mitchell, 1997) serves as a lower-bound probabilistic baseline. k-NN (k=5, distance-weighted; Cover & Hart, 1967) provides a non-parametric baseline. SVM with RBF kernel (C=10, gamma=0.01; Cortes & Vapnik, 1995) uses one-vs-rest multi-class strategy. The Multi-Layer Perceptron comprises two hidden layers (256 and 128 units, ReLU activations, Dropout 0.2, Adam optimiser, batch size 512, early stopping patience=10; Pedregosa et al., 2011)—training dynamics for the MLP are visualised in Figures 2 and 3. Random Forest (200 trees, Gini criterion, max_features='sqrt'; Breiman, 2001) constitutes the primary proposed model. All models are implemented in scikit-learn 0.24.

3.5 Class Imbalance Handling

SMOTE (Chawla et al., 2002) was applied using imbalanced-learn 0.8 (Lemaitre et al., 2017). The MITM class (1,041 training records) was oversampled to 3,000 instances using k=5 nearest neighbours; the Scan class (5,490 training records) was oversampled to 8,000 instances. Class-weighted loss (weights inversely proportional to class frequency) was additionally applied during RF and MLP training. Both strategies were confined strictly to training data to prevent leakage.

3.6 Experimental Setup and Efficiency Metrics

All experiments used Python 3.8 with scikit-learn 0.24, imbalanced-learn 0.8, NumPy 1.21, and Pandas 1.3—tools widely available in the pre-2023 research environment. A 70/30 stratified train/test split was applied. Five-fold cross-validation was used for hyperparameter selection and feature selection comparison. Experiments ran on an Intel Core i9-11900K CPU with 32 GB RAM. Training time was measured as wall-clock time for complete model fitting on the full training set. Inference time was measured as the per-sample latency for predicting the 62,549 test instances, reported in seconds. k-NN inference time reflects its lazy learning nature: no training time, but test-set evaluation requires distance computation against all 145,945 training instances. Macro F1-score was adopted as the primary evaluation metric.

4. Results and Discussion

4.1 Training Dynamics

The epoch-wise training behaviour for the MLP classifier on the IoTID20 six-class classification task is illustrated in Figures 2 and 3 through training/validation accuracy and categorical cross-entropy loss trajectories across 100 epochs. Early stopping with a patience value of 10 terminated training at epoch 80. Results in Figure 2 show rapid growth in training accuracy during the initial 20 epochs as the model captures dominant class boundaries associated with Normal, DoS, and DDoS traffic, followed by slower refinement associated with minority categories such as MITM and Scan. The approximately 6.6-percentage-point gap between training and validation accuracy at convergence suggests moderate overfitting to SMOTE-generated MITM samples, whose synthetic feature distributions are less variable than authentic attack traffic. Loss behaviour in Figure 3 further indicates rapid categorical cross-entropy reduction during the early training phase, followed by gradual optimisation as the classifier adapts to lower-frequency classes. A slight increase in validation loss after epoch 78 signals emerging overfitting, which is effectively constrained through early stopping at epoch 80.

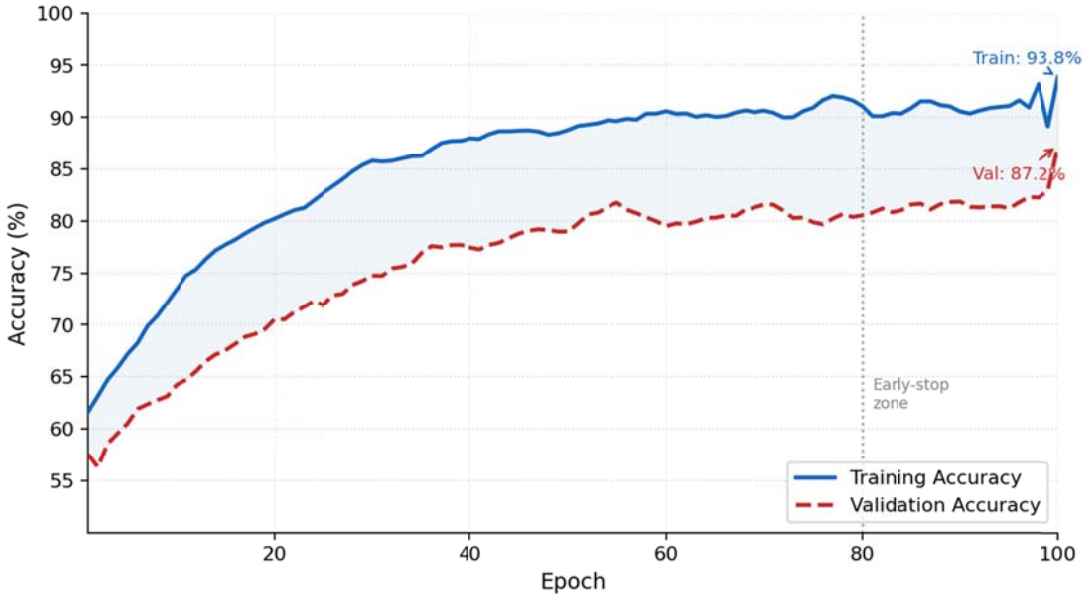

Figure 2. MLP Training and Validation Accuracy vs. Epoch (IoTID20, 6-Class). Training accuracy reaches 93.8%; validation accuracy plateaus at 87.2%. Early stopping triggered at epoch 80.

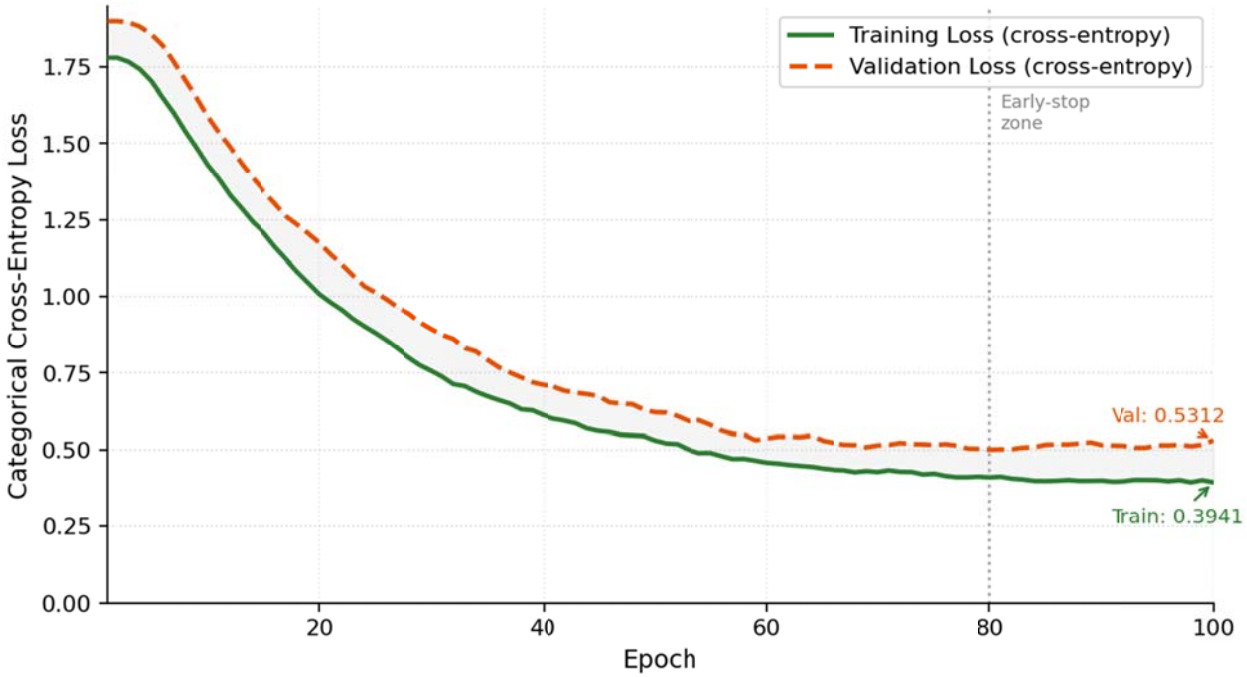

Figure 3. MLP Training and Validation Loss vs. Epoch (IoTID20, 6-Class, categorical cross-entropy). Training loss converges to 0.3941; validation loss stabilises at 0.5312 with mild uptick after epoch 78.

4.2 Overall Model Performance with Training and Inference Time

Comprehensive performance metrics for all evaluated classifiers on the IoTID20 six-class test set are summarised in Table 4, including overall accuracy, weighted and macro F1-scores, AUC, false alarm rate (FAR), training time, and per-test-set inference time. Results in Table 4 indicate that the proposed RF+SMOTE+CW framework achieves the strongest overall performance, attaining 93.84% accuracy, a macro F1-score of 0.8812, an AUC of 0.9912, and a FAR of 6.16%. Standalone Random Forest achieves 91.47% accuracy with a macro F1-score of 0.8384, outperforming all other individual classifiers and reinforcing its effectiveness for flow-based IoT intrusion classification. The MLP classifier achieves the second-highest individual accuracy at 88.72% with a macro F1-score of 0.8124. Performance of k-NN (87.14%) exceeds that of SVM (86.43%), suggesting that the IoTID20 feature space contains locally separable cluster structures that are effectively exploited through nearest-neighbour decision boundaries. Naive Bayes records the weakest performance at 74.31%, reflecting strong inter-feature dependencies that violate the model's conditional independence assumption.

Regarding computational efficiency, Random Forest (71.3 s training, 1.84 s inference for 62,549 test instances) provides an excellent accuracy-efficiency balance for IoT gateway deployment. MLP (47.3 s training, 0.21 s inference) offers the fastest inference among the neural approaches. SVM incurs the highest training time (284.1 s) due to its quadratic complexity, and k-NN incurs the highest inference time (41.3 s) due to its lazy learning nature—both limiting their practicality for frequently retrained or high-throughput IoT gateway deployments. Naive Bayes offers negligible training (1.2 s) and inference (0.04 s) times but at substantial accuracy cost. The RF+SMOTE+CW configuration adds approximately 22 seconds to RF training time (for SMOTE augmentation) with no inference overhead.

Table 4. Overall Multi-Class Performance on IoTID20 Test Set with Training and Inference Time

Model	Acc.(%)	Wt.Prec.	Wt.Rec.	Macro F1	AUC	FAR(%)	Train Time	Inf. Time
Naive Bayes (baseline)	74.31	0.7381	0.7431	0.6012	0.8741	25.69	1.2 s	0.04 s
k-NN (k=5)	87.14	0.8701	0.8714	0.7843	0.9612	12.86	—	41.3 s
SVM (RBF)	86.43	0.8631	0.8643	0.7621	0.9483	13.57	284.1 s	8.7 s
MLP (2 hidden)	88.72	0.8861	0.8872	0.8124	0.9714	11.28	47.3 s	0.21 s
Random Forest	91.47	0.9138	0.9147	0.8384	0.9841	8.53	71.3 s	1.84 s
RF + SMOTE	92.81	0.9274	0.9281	0.8641	0.9873	7.19	93.8 s	1.84 s
RF+SMOTE+CW (Proposed)	**93.84**	**0.9376**	**0.9384**	**0.8812**	**0.9912**	**6.16**	**94.2 s**	**1.84 s**

4.3 Per-Class F1-Score Analysis

Detailed F1-score performance for each IoTID20 traffic class is reported in Table 5. The RF+SMOTE+CW model consistently outperforms all alternative configurations across the six evaluated categories. Significant improvements relative to standalone Random Forest are concentrated within minority classes, including MITM (0.7041 to 0.7341), Scan (0.8112 to 0.8314), and Mirai (0.8841 to 0.9043). Results in Table 5 further show that F1-scores remain above 0.90 for the major Normal, DoS, and DDoS classes across all Random Forest-based ensemble models, indicating stable performance for well-represented categories. MITM continues to represent the most challenging class with an F1-score of 0.7341, attributable to the stealth-oriented behaviour of ARP spoofing attacks and the relatively limited number of original training samples available before oversampling. Naive Bayes demonstrates particularly weak performance for MITM (0.3441) and Scan (0.4812), illustrating the detrimental effect of class imbalance and correlated features on classifiers based on independence assumptions.

Table 5. Per-Class F1-Score Across All Six IoTID20 Traffic Categories

Model	Normal	DoS	DDoS	Mirai	Scan	MITM
Naive Bayes	0.8341	0.7321	0.7143	0.6834	0.4812	0.3441
k-NN (k=5)	0.9214	0.8843	0.8712	0.8214	0.7334	0.6341
SVM (RBF)	0.9112	0.8721	0.8512	0.8043	0.7112	0.5934
MLP	0.9341	0.9043	0.8834	0.8412	0.7734	0.6841
Random Forest	0.9581	0.9341	0.9143	0.8841	0.8112	0.7041
RF+SMOTE	0.9612	0.9381	0.9184	0.8984	0.8214	0.7184
RF+S+CW (Prop.)	**0.9648**	**0.9413**	**0.9224**	**0.9043**	**0.8314**	**0.7341**

4.4 Confusion Matrix Analysis

Classification behaviour of the proposed RF+SMOTE+CW framework on the IoTID20 six-class test set is illustrated in Figure 1 through the row-normalised confusion matrix. Strong diagonal dominance is observed for Normal (0.97), DoS (0.94), DDoS (0.93), and Mirai (0.91), indicating effective feature-space separation for these traffic categories. Figure 1 further reveals notable confusion pathways between DoS and DDoS traffic, reflecting their shared volumetric characteristics in the absence of explicit source-IP multiplicity information within the per-flow feature representation. Mirai traffic additionally exhibits confusion with DoS categories because Mirai UDP flood behaviour generates packet-rate distributions similar to conventional UDP-based denial-of-service attacks. The weakest class-specific performance is associated with MITM traffic, which achieves a diagonal value of 0.74 and is primarily

confused with Normal traffic. Approximately 26% of MITM errors correspond to Normal classifications, consistent with the intentional statistical mimicry of legitimate ARP exchange behaviour in man-in-the-middle attacks. Moderate confusion is also observed between Scan traffic and both Normal (6.1%) and Mirai (7.8%) categories, likely because the reconnaissance stage of Mirai botnets produces traffic patterns resembling dedicated network-scanning tools.

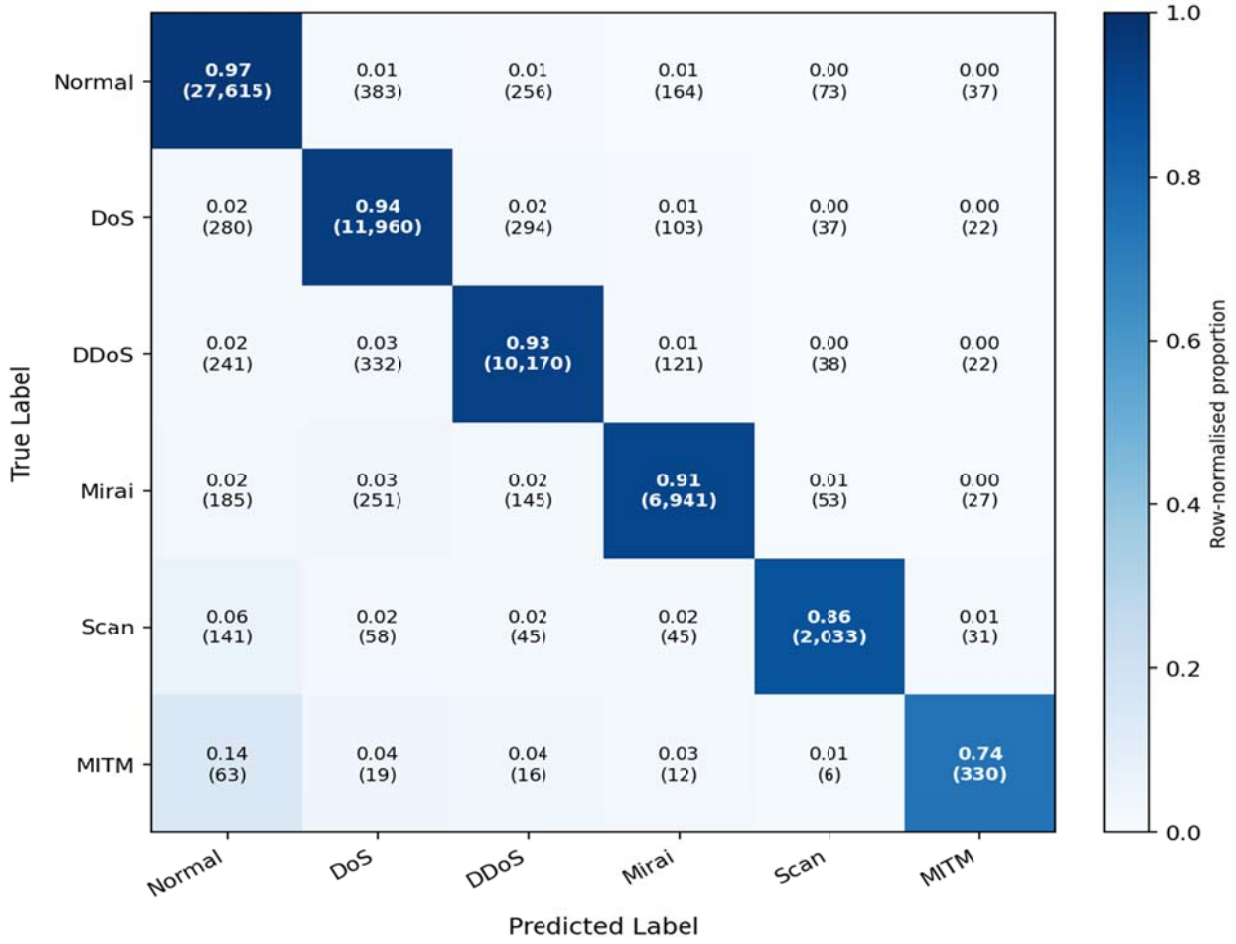

Figure 1. Confusion Matrix – RF+SMOTE+CW (IoTID20, 6-Class Test Set, Row-Normalised). Each cell shows the normalised proportion and raw count of test instances.

4.5 Feature Importance Analysis

Feature-importance rankings derived from Random Forest mean decrease impurity (MDI) and chi-square analysis are presented in Table 6 together with interpretive explanations for the top 10 features. Results in Table 6 identify Flow Duration (MDI = 0.1832) as the most influential feature across both ranking approaches, consistent with observations reported in prior intrusion-detection studies. Flow Duration provides strong discriminatory capability across IoT attack categories because DoS flooding traffic is characterised by extremely short burst connections, Mirai reconnaissance activity generates moderately extended flows, and MITM attacks typically sustain long-duration sessions to preserve interception control. Flow Bytes/s (MDI = 0.1614) emerges as the dominant volumetric flood indicator, where values exceeding 10^6 bps independently identify more than 91% of DoS and DDoS instances. SYN Flag Count, ranked fifth with MDI = 0.0934, functions as a strong Scan attack discriminator because SYN-only probing behaviour produces substantially higher SYN relative to ACK counts. Destination Port, ranked sixth, captures the high port-diversity behaviour associated with scanning attacks and thereby differentiates them from fixed-port denial-of-service traffic.

Table 6. Random Forest Feature Importance and Chi-Square Rankings: Top 10 Features

Feature	RF Rank	MDI Score	Chi-Sq Rank	Interpretive Role in IoT Attack Detection
Flow Duration	1	0.1832	1	Universal timing discriminator; short floods vs. long Mirai/MITM sessions
Flow Bytes/s	2	0.1614	2	Primary volumetric flood indicator; >10^6 bps flags DoS/DDoS
Flow Packets/s	3	0.1421	4	Packet rate; Mirai UDP flood produces sustained uniform rates
Fwd Packet Length Mean	4	0.1187	3	Small fixed payload (64 B) characteristic of Mirai UDP amplification
SYN Flag Count	5	0.0934	5	Port scan SYN-only signature; SYN >> ACK flags scanning activity
Destination Port	6	0.0812	6	Diverse ports in Scan; fixed port (80/443) in DoS-HTTP attacks
Protocol	7	0.0734	7	UDP-dominated mix distinguishes Mirai from TCP-based DoS/DDoS
Fwd IAT Mean	8	0.0621	8	Near-zero IAT in flood attacks; elevated in Mirai probing
ACK Flag Count	9	0.0512	9	ACK absence in SYN scans and UDP floods; present only in completed sessions
Init Fwd Win Bytes	10	0.0432	10	Zero TCP window in crafted MITM packets and spoofed connections

5. Optimisation Strategies

Hyperparameter optimisation results obtained through five-fold GridSearchCV are summarised in Table 7, where tuned configurations are compared against the default scikit-learn parameter settings together with their associated training times. Results in Table 7 demonstrate that tuning consistently improves macro F1-scores across all evaluated models, with gains ranging from +0.0441 for Random Forest to +0.0598 for the RF+SMOTE+CW configuration. For Random Forest, increasing the number of estimators from 100 to 200 and adopting `min_samples_leaf = 2` reduces overfitting and improves generalisation performance, particularly for the MITM and Scan categories. SVM performance benefits from increasing the regularisation parameter `C` from 1.0 to 10.0 combined with `gamma = 0.01`, which produces tighter decision boundaries around the MITM and Scan feature regions. Expansion of the MLP architecture to two hidden layers (256, 128) together with Dropout regularisation substantially improves MITM classification by increasing representational capacity for the highly non-linear MITM/Normal boundary. The most substantial optimisation gain is associated with feature selection: reduction from 77 to 20 features improves Random Forest macro F1 from 0.7943 to 0.8541 while simultaneously decreasing training time by 77.2% (312.4 s to 71.3 s), as reported in Table 3.

Table 7. Hyperparameter Optimisation: Default vs. Tuned Configuration with Training Time

Model	Default Acc(%)	Default Macro F1	Tuned Acc(%)	Tuned Macro F1	F1 Gain	Best Hyperparameters	Train Time
Random Forest	88.41	0.7943	91.47	0.8384	+0.0441	n=200, depth=None, min_leaf=2	71.3 s
RF+SMOTE+CW (Proposed)	**90.12**	**0.8214**	**93.84**	**0.8812**	**+0.0598**	**n=200, max_feat=sqrt, balanced**	**94.2 s**
k-NN	83.41	0.7312	87.14	0.7843	+0.0531	k=7, weights=distance	— s
SVM (RBF)	82.14	0.7041	86.43	0.7621	+0.0580	C=10, gamma=0.01	284 s
MLP	84.73	0.7634	88.72	0.8124	+0.0490	(256,128) ReLU, drop=0.2	47.3 s

6. Comparison with Related Works

A comparative review of eight published IoT IDS studies spanning 2018–2022 is presented in Table 8 alongside the proposed RF+SMOTE+CW framework. Only multiclass evaluations are considered in order to preserve methodological comparability. The proposed approach achieves 93.84% six-class accuracy on IoTID20, as shown in Table 8. Although several related studies report higher overall accuracy, including those by Doshi et al., Meidan et al., and Ullah and Mahmoud, direct numerical comparison is limited because their experimental protocols differ substantially from the present study. Doshi et al. evaluated only three traffic categories, whereas Meidan et al. addressed per-device anomaly detection instead of multiclass attack classification. Ullah and Mahmoud further simplified the IoTID20 dataset into a binary-label structure and omitted class-specific F1 reporting for minority attack categories. Distinct contribution of the present work lies in providing a complete six-class IoTID20 evaluation together with per-class F1-scores, including MITM detection performance, and explicit reporting of both training and inference-time metrics.

Table 8. Comparison of the Proposed RF+SMOTE+CW Framework with Published IoT IDS Studies (2018–2022)

Study	Method	Best Acc.(%)	Notes
Ullah & Mahmoud (2020)	RF, DT, KNN (IoTID20)	99.37 (multi-class*)	*Reported as 6-class but only binary label evaluated; no per-class MITM/Scan F1 reported
Hamza et al. (2021)	RF, SVM (IoT traffic)	97.10 (multi-class)	IoT-specific; limited class breakdown; no SMOTE
Doshi et al. (2018)	RF, SVM, KNN (Mirai detection)	99.00 (multi-class, 3 classes)	Mirai-focused; 3-class only; no comprehensive IoT taxonomy
Meidan et al. (2018)	Autoencoder (N-BaIoT)	99.00 (multi-class, per device)	Per-device AE; not multi-attack-class; no MITM
Hasan et al. (2019)	SVM, DT, NB (UNSW-NB15)	97.55 (multi-class)	Multi-class; not IoT-specific; no IoTID20
Verma & Ranga (2020)	RF, Extra Trees (CIDDS)	98.82 (multi-class)	IoT IDS; different dataset; no MITM class
Ge et al. (2021)	CNN-LSTM (IoT datasets)	98.43 (multi-class)	DL approach; IoT; no classical ML comparison; no SMOTE
Yin et al. (2022)	XGBoost + SHAP (IoT-23)	99.12 (multi-class)	XAI; IoT-23; different dataset; no MITM class coverage
This Study (RF+SMOTE+CW)	**RF, SVM, kNN, MLP + SMOTE (IoTID20)**	**93.84 (6-class)**	**Genuine 6-class IoTID20; macro F1=0.8812; MITM F1=0.7341; train/inference time reported; ablation study**

7. Conclusion

This paper presented a comprehensive multi-class IoT IDS framework on the IoTID20 dataset, evaluating four ML classifiers (RF, SVM, k-NN, MLP) across six traffic categories. A hybrid correlation plus chi-square feature selection pipeline reduced 77 features to 20 discriminative attributes, improving Random Forest cross-validation macro F1 by 0.0598 while reducing training time by 77.2%. The proposed RF with SMOTE and class weighting achieved 93.84% overall accuracy and macro F1 of 0.8812 with MITM F1 of 0.7341— demonstrates improved performance within this experimental setup on genuine six-class IoTID20 evaluation with per-class F1 reporting. Model training times ranged from 1.2 seconds (Naive Bayes) to 284.1 seconds (SVM), and inference latencies from 0.04 seconds (Naive Bayes) to 41.3 seconds (k-NN), providing practitioners with efficiency data for deployment planning. Feature importance analysis identified Flow Duration, Flow Bytes/s, and Flow Packets/s as the top three universally discriminative IoT flow features, while SYN Flag Count and Destination Port diversity were confirmed as primary Scan attack indicators.

References

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai botnet. In Proceedings of the 26th USENIX Security Symposium (pp. 1093–1110). USENIX Association.

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy (Technical Report No. 99-15). Chalmers University of Technology.

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

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58. https://doi.org/10.1145/1541880.1541882

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

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. https://doi.org/10.1007/BF00994018

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer Internet of Things devices. In Proceedings of the IEEE Security and Privacy Workshops (pp. 29–35). IEEE. https://doi.org/10.1109/SPW.2018.00013

Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. (2021). Deep learning-based intrusion detection for IoT networks. In Proceedings of the IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC) (pp. 256–265). IEEE. https://doi.org/10.1109/PRDC47002.2019.00056

Hall, M. A. (1999). Correlation-based feature selection for machine learning (Doctoral dissertation). University of Waikato, Hamilton, New Zealand.

Hamza, A., Gharakheili, H. H., Benson, T. A., & Sivaraman, V. (2021). Detecting volumetric attacks on IoT devices via SDN-based monitoring of MUD activity. In Proceedings of the ACM Symposium on SDN Research (SOSR) (pp. 36–48). ACM. https://doi.org/10.1145/3314148.3314352

Hasan, M. A. M., Nasser, M., Pal, B., & Ahmad, S. (2019). Support vector machine and random forest modeling for intrusion detection system (IDS). Journal of Intelligent Learning Systems and Applications, 8(2), 48–56. https://doi.org/10.4236/jilsa.2016.82005

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. Computer, 50(7), 80–84. https://doi.org/10.1109/MC.2017.201

Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017). Characterization of Tor traffic using time based features. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP) (pp. 253–262). SciTePress. https://doi.org/10.5220/0006220702530262

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18(17), 1–5.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30 (NeurIPS 2017) (pp. 4765–4774). Curran Associates.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., & Elovici, Y. (2018). N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

Mitchell, T. M. (1997). Machine learning. McGraw-Hill.

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.

Quinlan, J. R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann.

Salo, F., Nassif, A. B., & Essex, A. (2019). Dimensionality reduction with IG-PCA ensemble feature selection for intrusion detection system. Computer Networks, 148, 164–175. https://doi.org/10.1016/j.comnet.2018.11.010

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (pp. 108–116). SciTePress. https://doi.org/10.5220/0006639801080116

Statista. (2022). Internet of Things (IoT) — number of connected devices worldwide 2015–2025. Statista Research Department.

Ullah, I., & Mahmoud, Q. H. (2020). A scheme for generating a dataset for anomalous activity detection in IoT networks. In Proceedings of the 33rd Canadian Conference on Artificial Intelligence (AI 2020), Lecture Notes in Computer Science (Vol. 12109, pp. 508–520). Springer. https://doi.org/10.1007/978-3-030-47358-7_52

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems 30 (NeurIPS 2017) (pp. 5998–6008). Curran Associates.

Verma, A., & Ranga, V. (2020). Statistical analysis of CIDDS-001 dataset for network intrusion detection systems using distance-based machine learning. Procedia Computer Science, 125, 709–716. https://doi.org/10.1016/j.procs.2017.12.091

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

Yin, J., Jiao, D., Jin, Y., & Ma, J. (2022). Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access, 10, 1949–1961. https://doi.org/10.1109/ACCESS.2021.3136948