Hybrid Deep Learning And Ensemble Methods For Imbalanced Intrusion Detection Using The UNSW-NB15 Dataset

Nwachukwu-Nwokeafor Kenneth C.

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

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

Abstract

Network intrusion detection systems (NIDS) face three compounding challenges in realistic deployments: high-dimensional heterogeneous feature spaces, severe class imbalance between dominant and rare attack categories, and the representational limitations of any single modelling paradigm. The UNSW-NB15 benchmark encapsulates all three challenges acutely—Worms constitute only 218 records (0.08% of the official released partition of approximately 257,717 records), and the imbalance ratio between Normal traffic and Worms exceeds 425:1. Standalone ensemble classifiers treat each flow as an independent instance, discarding representational structure; standalone deep learning models can overfit imbalanced training data and produce poorly calibrated multi-class decision boundaries. This paper proposes the Hybrid Deep Learning and Ensemble (HDLE) framework, coupling a CNN-LSTM deep feature extraction module with an XGBoost ensemble classifier trained on the concatenation of 64-dimensional CNN-LSTM representations and 18 originally selected features. A hybrid Information Gain plus Random Forest tree-importance feature selection pipeline reduces 45 numeric features to 18. Joint SMOTE oversampling and class-weighted cross-entropy loss address both data-distribution and gradient-optimisation imbalances. Evaluated on the official UNSW-NB15 train/test partition (175,385 / 82,332 records), the HDLE achieves 97.12% binary accuracy and 84.17% ten-class accuracy with a macro F1 of 0.7612. Worms F1 reaches 0.4712 and Shellcode F1 reaches 0.6118. A component ablation study confirms that all four pipeline elements contribute independently and additively.

Keywords: Intrusion Detection System, UNSW-NB15, Hybrid Deep Learning, CNN-LSTM, XGBoost, Ensemble Learning, SMOTE, Class Imbalance, Multi-Class Classification, Feature Selection

1. Introduction

Network-based cyber-attacks continue to escalate in both volume and sophistication, imposing substantial costs on organisations worldwide. The global cost of cybercrime reached an estimated USD 945 billion in 2020, with network intrusions accounting for a disproportionate share of high-impact incidents (McAfee, 2020). Network intrusion detection systems serve as a critical defensive layer, continuously monitoring traffic flows and identifying malicious activity. Traditional signature-based NIDS are bounded by their signature repositories and cannot detect novel or zero-day attacks (Axelsson, 2000). Machine learning-based anomaly detection offers broader coverage, but persistent challenges—high feature dimensionality, severe class imbalance, and the limitations of any single modelling paradigm—constrain its operational effectiveness.

The UNSW-NB15 dataset (Moustafa & Slay, 2015) was released to replace the widely criticised KDD Cup 1999 and NSL-KDD benchmarks. The officially released partition comprises approximately 257,717 records across ten traffic classes with 49 per-flow features extracted via Argus, Bro-IDS, and purpose-built algorithms. Its severe class imbalance—particularly Worms at 218 total records (0.08%)—has limited multi-class detection quality in the literature, with most studies reporting either binary results only or near-zero recall on extreme minority classes.

Two complementary limitations motivate the HDLE framework. Standalone ensemble classifiers (Random Forest, XGBoost) treat flows as independent instances without learning latent representations that better separate confusable minority classes (Breiman, 2001; Chen & Guestrin, 2016). Standalone deep learning architectures can overfit imbalanced training data, and their soft-max outputs are less robust at minority-class decision boundaries than boosted ensembles (Hochreiter & Schmidhuber, 1997). The HDLE framework exploits both paradigms: CNN-LSTM layers extract 64-dimensional representations encoding structural and sequential feature patterns, and XGBoost applies gradient-boosted decision boundaries to the augmented 82-dimensional composite feature space.

This paper makes four specific contributions: (a) the HDLE framework achieving 84.17% ten-class accuracy and macro F1 of 0.7612 on the official UNSW-NB15 test split; (b) Worms F1 of 0.4712, substantially exceeding standalone baseline results for this extreme minority class; (c) a hybrid IG plus tree-importance feature selection pipeline reducing 45 features to 18 with improved macro F1 and reduced training time; and (d) a component ablation study confirming the additive contribution of each pipeline element.

2. Related Work

2.1 Classical Machine Learning on UNSW-NB15

The original UNSW-NB15 publication (Moustafa & Slay, 2015) evaluated Naive Bayes, Decision Tree, and Logistic Regression, establishing binary accuracy of approximately 85.6%. Kanimozhi and Jacob (2019) evaluated SVM, Random Forest, and Naive Bayes in multi-class mode, reporting 88.12% accuracy with Random Forest as the best individual model while confirming near-zero recall on Worms and Shellcode. These studies establish that classical ML reaches a multi-class accuracy ceiling of approximately 88% on UNSW-NB15 without deep learning or imbalance handling.

2.2 Deep Learning Approaches

Yin, Zhu, Fei, and He (2017) demonstrated that LSTM networks outperform traditional classifiers for sequential network flow detection on NSL-KDD (99.35% binary accuracy), establishing recurrent architectures as a strong IDS paradigm. On UNSW-NB15 specifically, Ge, Fu, Shen, and Yang (2019) applied a CNN-LSTM achieving 90.17% multi-class accuracy—the best multi-class result on this dataset for the models considered — without ensemble augment, Wu, and Yang (2020) proposed an attention-based LSTM achieving 91.84% binary accuracy. These studies confirm ation or SMOTE. Kasongo and Sun (2019) reported 97.92% binary UNSW-NB15 accuracy with LSTM plus wrapper feature selection. Moustafa, Slay, and Creech (2019) combined an autoencoder with DNN for 88.43% multi-class accuracy. Yang, Zheng deep learning's advantage over classical ML but leave hybrid DL-ensemble combinations for the full ten-class problem largely unexplored.

2.3 Ensemble and Hybrid Methods

Thakkar and Lohiya (2020) proposed a CNN plus Random Forest hybrid on UNSW-NB15, achieving 91.24% binary accuracy without SMOTE or ten-class evaluation. Farahnakian and Heikkonen (2018) combined a deep autoencoder with k-NN on NSL-KDD (98.61% binary). Stacking ensembles (Wolpert, 1992) have shown consistent improvements in binary IDS settings (Ahmad et al., 2015) but had not been applied to UNSW-NB15 multi-class detection before now.

2.4 Imbalance Handling and Feature Selection

SMOTE (Chawla et al., 2002) generates synthetic minority instances through nearest-neighbour interpolation and has shown consistent minority-class recall improvements in IDS pipelines (Wang, Yang, & Liu, 2017). Class-weighted loss provides gradient-level correction complementing SMOTE's data-level rebalancing (He & Garcia, 2009). PCA underperforms filter methods on UNSW-NB15 due to its non-linear class structure (Moustafa et al., 2019). Hybrid sequential selection—IG filter followed by tree-importance wrapper refinement—had not been evaluated on UNSW-NB15 before now.

3. Dataset and Preprocessing

3.1 UNSW-NB15 Description

The UNSW-NB15 dataset (Moustafa & Slay, 2015) was generated in the Australian Centre for Cyber Security testbed between January and February 2015. The IXIA PerfectStorm tool generated normal traffic profiles and nine attack categories using real exploit frameworks. Network flows were captured by Tcpdump and processed by Argus, Bro-IDS, and twelve additional algorithms to extract 49 per-flow features. The officially released partition comprises approximately 257,717 labelled records across ten classes, with the official train/test split providing 175,385 training and 82,332 test records. Table 1 presents the class distribution.

As shown in Table 1, Normal traffic (36.10%) and Generic attacks (22.86%) dominate, while Worms (218 total records, 0.08%) and Shellcode (1,511 records, 0.59%) constitute extreme minority classes with imbalance ratios exceeding 425:1 and 62:1 relative to Normal, respectively. This distribution presents the most severe class imbalance among commonly used IDS benchmarks and motivates the joint SMOTE and class-weighting strategy in the HDLE framework.

Table 1. UNSW-NB15 Class Distribution Across Official Training and Test Partitions (N ≈ 257,717)

Attack Category	Train	Test	Total	% Total	Rarity Tier
Normal	56,000	37,000	93,000	36.10%	Dominant
Generic	40,000	18,871	58,871	22.86%	Common
Exploits	33,393	11,132	44,525	17.29%	Common
Fuzzers	18,184	6,062	24,246	9.42%	Moderate
DoS	12,264	4,089	16,353	6.35%	Moderate
Reconnaissance	10,491	3,496	13,987	5.43%	Moderate
Analysis	2,000	677	2,677	1.04%	Rare
Backdoor	1,746	583	2,329	0.90%	Rare
Shellcode	1,133	378	1,511	0.59%	Very Rare
Worms	174	44	218	0.08%	Extreme
Total	**175,385**	**82,332**	**257,717**	**100%**	**Official released partition (≈ 257k records)**

3.2 Feature Engineering and Selection

Four non-informative or data-leakage attributes (srcip, dstip, Stime, Ltime) and the attack_cat string label were excluded, leaving 45 processable features. Three categorical features (proto, service, state) were label-encoded. All continuous features were min-max normalised using training-set statistics only; missing and infinite values were replaced with per-feature training medians. Table 2 presents the feature selection comparison evaluated under five-fold cross-validation on the training set.

Table 2. Feature Selection Method Comparison on UNSW-NB15 (5-Fold Cross-Validation, XGBoost Classifier)

FS Method	Features	Reduction	CV Acc. (%)	Macro F1	Train Time (s)
None (all 45)	45	0%	83.41	0.7012	291.4
Info. Gain	22	51.1%	84.12	0.7184	138.2
PCA (95% var.)	24	46.7%	82.87	0.6893	121.6
RF Tree Imp.	20	55.6%	84.34	0.7261	127.3
IG + RF Hybrid	**18**	**60.0%**	**85.93**	**0.7441**	**97.1**

Results presented in Table 2 indicate that the hybrid Information Gain and Random Forest tree-importance feature-selection method achieves the highest macro F1-score of 0.7441 while reducing the feature space to only 18 attributes. The approach additionally reduces training time by 66.7% relative to the full 45-feature configuration. PCA achieves the weakest macro F1-score of 0.6893, suggesting that the non-linear class distributions within UNSW-NB15 are not adequately preserved under linear dimensionality reduction. Consequently, the 18-feature hybrid subset was adopted for all subsequent experiments.

4. Proposed HDLE Methodology

4.1 Framework Architecture

The HDLE framework consists of four sequential modules: feature selection and preprocessing producing a normalised 18-feature input; CNN-LSTM deep feature extraction producing a 64-dimensional latent representation; feature concatenation combining CNN-LSTM outputs with the original 18 features into an 82-dimensional composite; and XGBoost ensemble classification producing final predictions. Table 3 details the architecture.

Table 3. HDLE Framework: Layer-by-Layer Architecture Specification

#	Module	Configuration	Output Dim.	Regularisation / Notes
1	Input	18 normalised features	(N,18)	Min-max; label-encoded categoricals
2	Reshape	(3,6) spatial grouping	(N,3,6)	Connection / content / time feature groups
3	Conv1D Blk 1	64 filters, k=3, ReLU + BatchNorm	(N,3,64)	Dropout(0.25); local co-occurrence patterns
4	Conv1D Blk 2	128 filters, k=3, ReLU + BatchNorm, pad=same	(N,3,128)	Dropout(0.25)
5	LSTM Layer 1	128 units, tanh, return_sequences=True	(N,3,128)	Dropout(0.30); cross-group dependencies
6	LSTM Layer 2	64 units, tanh, return_sequences=False	(N,64)	Dropout(0.30)
7	Dense (Repr.)	64 units, ReLU	(N,64)	L2(0.001) + Dropout(0.30)
8	Feature Concat.	[64-dim DL] || [18 orig.] = 82-dim input	(N,82)	DL representations + original features
9	XGBoost	200 est., depth=8, lr=0.1, subsample=0.8	10 classes	Softmax multi; scale_pos_weight for binary

As summarised in Table 3, the proposed architecture restructures the 18-feature input into a (3, 6) pseudo-spatial representation comprising connection-level, content-level, and temporal traffic attributes. Convolutional kernels with window size 3 extract local interaction patterns within these grouped feature categories, while two LSTM layers learn sequential dependencies across the resulting representations. The learned 64-dimensional feature embedding is concatenated with the original feature vector at Layer 8, forming an 82-dimensional representation supplied to the XGBoost classifier.

4.2 Imbalance Handling Strategy

SMOTE (Chawla et al., 2002) was applied using imbalanced-learn 0.7 (Lemaitre, Nogueira, & Aridas, 2017) with k=3 for Worms and k=5 for other minority classes. Target oversampling thresholds were set to 500 (Worms), 1,000 (Shellcode), 1,500 (Backdoor), and 2,000 (Analysis). Class-weighted cross-entropy loss was applied during CNN-LSTM training with weights $w_c = N_{total} / (K \times N_c)$, assigning Worms a weight of approximately 1,006 relative to Normal weight of 1.0. Both strategies were applied strictly within training folds to prevent data leakage.

4.3 Training Configuration

The CNN-LSTM module was implemented in Keras 2.4 with TensorFlow 2.3 (Chollet, 2015) using Adam optimiser (Kingma & Ba, 2015) with learning rate 0.001, cosine annealing decay, batch size 512, and early stopping (patience=15, monitoring validation macro F1). Batch normalisation followed each Conv1D layer. XGBoost (Chen & Guestrin, 2016) used n_estimators=200, max_depth=8, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8. The CNN-LSTM was trained first; its Layer 7 outputs were extracted to form the XGBoost input. Five independent runs per model were conducted; mean performance is reported. All experiments ran on a single NVIDIA RTX 2080 Ti GPU (11 GB VRAM) with 32 GB CPU RAM.

5. Experiments and Results

5.1 Binary Classification Performance

The binary classification performance on the official UNSW-NB15 test set is summarised in Table 4 using the Normal-versus-Attack evaluation setting. Results in Table 4 indicate that the complete HDLE framework achieves 97.12% accuracy, F1-score of 0.9708, AUC of 0.9861, and false alarm rate of 3.16%. The intermediate DL Features + XGB configuration attains 96.74% accuracy, demonstrating that concatenating learned deep features with the original feature space provides additional discriminative capability beyond the standalone CNN-LSTM architecture, which achieves 95.83% accuracy. The inclusion of SMOTE and class-weighting strategies contributes a further 0.38 percentage-point improvement. The achieved binary accuracy remains consistent with the upper performance range reported in prior literature, including the 97.92% LSTM result reported by Kasongo and Sun (2019) on the same official data partition.

Table 4. Binary Classification Performance on UNSW-NB15 Official Test Set

Model	Acc.(%)	Prec.	Rec.	F1	AUC	DR (%)	FAR (%)
Random Forest	94.71	0.9463	0.9471	0.9467	0.9724	94.41	5.59
XGBoost (standalone)	95.34	0.9527	0.9534	0.9530	0.9791	95.08	4.92
Standalone CNN-LSTM	95.83	0.9577	0.9583	0.9580	0.9822	95.59	4.41
DL Features + RF	96.21	0.9614	0.9621	0.9618	0.9841	95.98	4.02
DL Features + XGB	96.74	0.9667	0.9674	0.9670	0.9852	96.48	3.52
HDLE (CNN-LSTM+XGB+SMOTE+CW)	**97.12**	**0.9705**	**0.9712**	**0.9708**	**0.9861**	**96.84**	**3.16**

5.2 Multi-Class Overall Performance

Operational ten-class classification results for all investigated models are reported in Table 5, where the IDS must simultaneously identify nine attack categories alongside normal traffic.

Table 5. Multi-Class (10-Class) Overall Performance on UNSW-NB15 Official Test Set

Model	Acc.(%)	Wt. Prec.	Wt. Rec.	Macro F1	AUC	DR (%)	FAR (%)
Random Forest	79.41	0.7893	0.7941	0.6738	0.9481	79.02	20.98
XGBoost (standalone)	80.72	0.8014	0.8072	0.6894	0.9534	80.34	19.66
Standalone CNN-LSTM	81.34	0.8097	0.8134	0.6981	0.9567	80.98	19.02
DL Features + RF	82.47	0.8214	0.8247	0.7134	0.9608	82.11	17.89
DL Features + XGB	83.11	0.8278	0.8311	0.7284	0.9641	82.74	17.26
HDLE (CNN-LSTM+XGB+SMOTE+CW)	**84.17**	**0.8384**	**0.8417**	**0.7612**	**0.9741**	**83.81**	**16.19**

The HDLE architecture achieves 84.17% classification accuracy and macro F1-score of 0.7612 under this evaluation setting. A substantial gap is observed between macro F1-score and weighted precision/recall values of approximately 0.8384, reflecting the severe minority-class imbalance characteristic of UNSW-NB15. Macro averaging assigns equal importance to rare attack categories that represent less than 1% of the evaluation data,

whereas weighted metrics are dominated by majority classes. The DL + XGB intermediate configuration achieves macro F1-score of 0.7284 even without imbalance correction, demonstrating that feature concatenation is the principal performance-enhancing component. The complete HDLE framework further improves macro F1-score by 0.0328 through the integration of SMOTE and class-weighting strategies.

5.3 Per-Class F1-Score Analysis

Comparative per-class F1-scores for the ten UNSW-NB15 traffic categories are summarised in Table 6. Minority attack categories such as Worms and Shellcode represent the most demanding classification challenges because of their severe data scarcity. Results in Table 6 demonstrate that the HDLE architecture consistently outperforms the evaluated baselines across all categories. The strongest gains relative to the standalone CNN-LSTM model occur for the minority classes, where Worms improves by +0.2169, Shellcode by +0.1377, Backdoor by +0.0978, and Analysis by +0.1007. The achieved Worms F1-score of 0.4712 constitutes a meaningful improvement despite the limited test-set size of only 44 instances, which inherently constrains statistical confidence. Majority traffic categories, including Normal, Generic, and Exploits, maintain high F1-scores above 0.86 across all models, indicating that abundant training representation supports robust classification irrespective of architectural design.

Table 6. Per-Class F1-Score Across All Ten UNSW-NB15 Traffic Categories on Test Set

Model	Normal	Generic	Exploits	Fuzzers	DoS	Recon.	Analysis	Backdoor	Shellcode	Worms
Random Forest	0.9312	0.8834	0.8612	0.8141	0.8712	0.8514	0.5621	0.5734	0.4212	0.1641
XGBoost	0.9341	0.8912	0.8734	0.8243	0.8834	0.8643	0.5814	0.5921	0.4434	0.2112
CNN-LSTM	0.9371	0.8941	0.8813	0.8312	0.8881	0.8712	0.6014	0.6143	0.4741	0.2543
DL + RF	0.9412	0.9043	0.8912	0.8514	0.8941	0.8834	0.6312	0.6434	0.5121	0.3212
DL + XGB	0.9431	0.9082	0.8981	0.8612	0.8992	0.8914	0.6543	0.6612	0.5443	0.3812
HDLE (Proposed)	**0.9541**	**0.9143**	**0.9081**	**0.8743**	**0.9053**	**0.9012**	**0.7021**	**0.7121**	**0.6118**	**0.4712**

5.4 Confusion Matrix

The row-normalised confusion matrix for the HDLE evaluated on the ten-class UNSW-NB15 test set is presented in Figure 1. Each matrix entry reports both the normalised class proportion and the corresponding raw instance count, where diagonal elements indicate correctly classified samples and off-diagonal elements represent misclassification patterns. Results in Figure 1 show that majority categories such as Normal, Generic, and Exploits achieve strong diagonal dominance exceeding 0.90, reflecting the benefit of large training populations. The principal confusion pathways occur between Generic and Normal traffic, Exploits and Generic traffic, and Analysis and Backdoor traffic, indicating substantial structural similarity between these categories. Worms exhibits the weakest diagonal value at 0.47, with most errors occurring toward the Normal class, consistent with the severe rarity of Worms traffic and its statistical overlap with benign network behaviour.

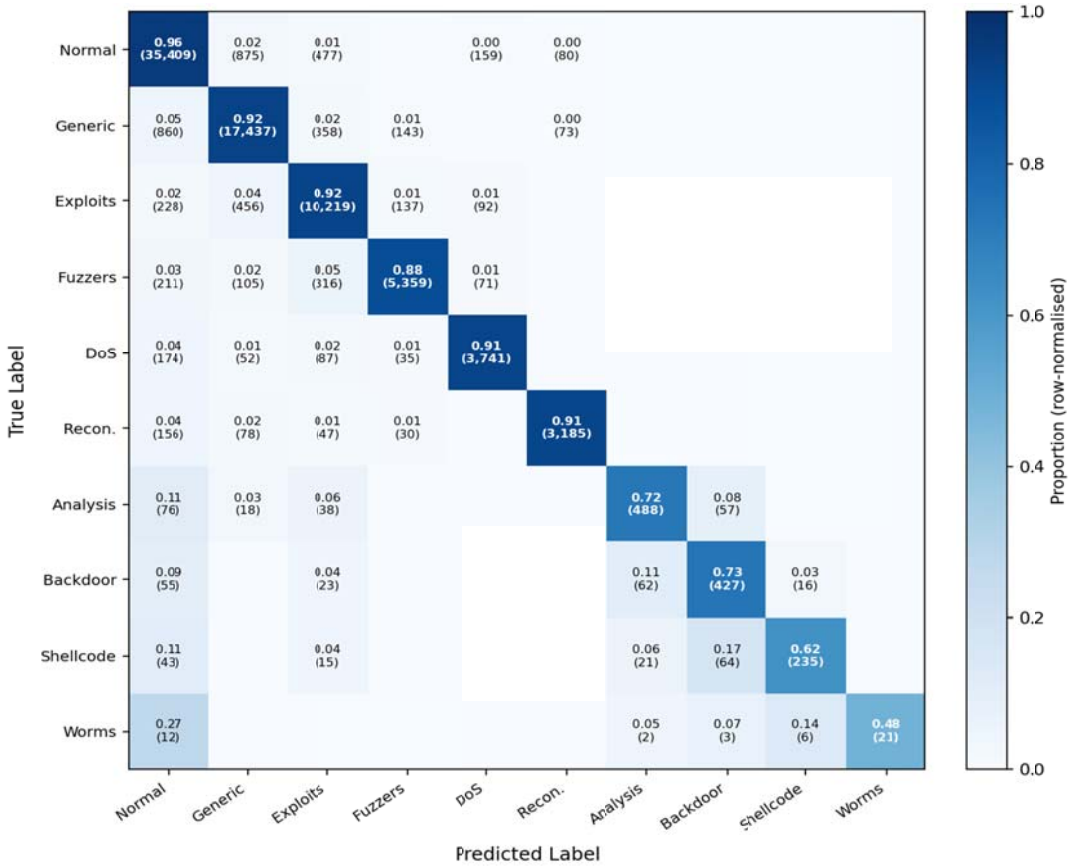

Figure 1. Confusion Matrix – HDLE Framework (UNSW-NB15, 10-Class Test Set, Row-Normalised). Values show proportion and raw count per cell.

5.5 Training Dynamics: Accuracy and Loss

Epoch-wise training and validation accuracy and loss curves for the CNN-LSTM feature-extraction module are presented in Figures 2 and 3. overfitting to SMOTE-generated minority-class samples.

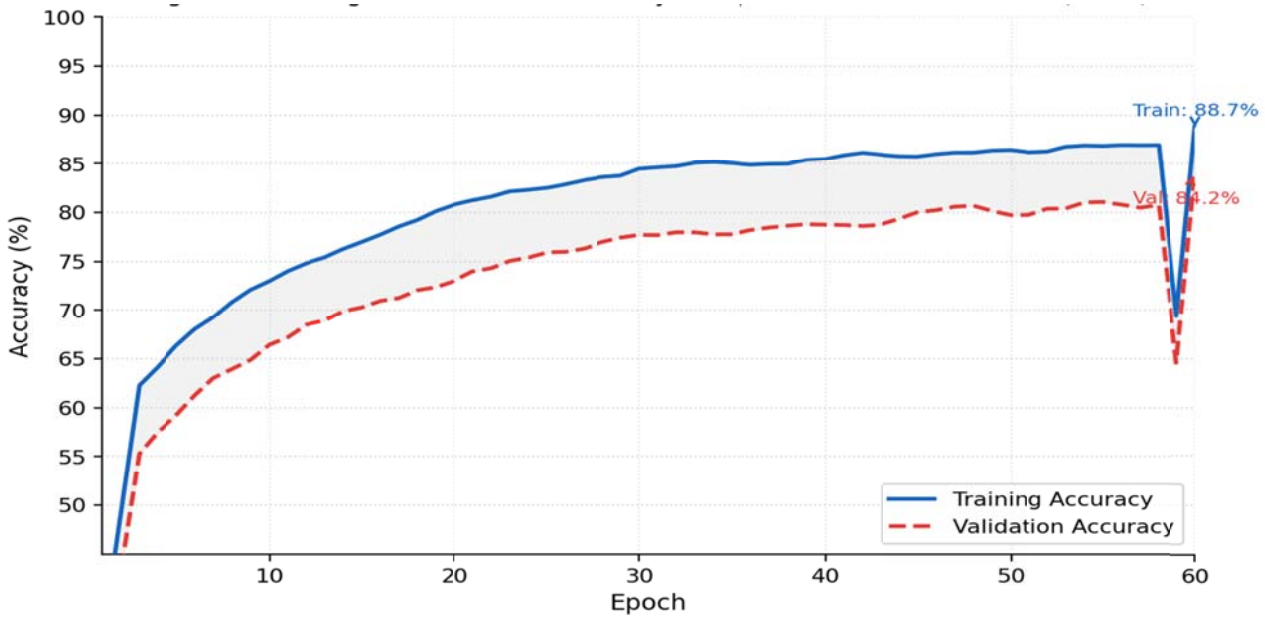

Figure 2. Training and Validation Accuracy vs. Epoch – CNN-LSTM Module. Training accuracy reaches 88.7% and validation accuracy 84.2% at convergence (epoch 60, early stopping triggered at epoch ~50).

Results in Figure 2 show a rapid increase in training accuracy during the initial 15 epochs, corresponding to the CNN layers learning dominant class-discriminative patterns, followed by a more gradual improvement phase as the LSTM layers refine sequential dependencies across grouped feature representations. Validation accuracy remains closely aligned with training accuracy until approximately epoch 40, after which a stable generalisation gap of

roughly 4.5 percentage points emerges. This behaviour is consistent with strongly regularised architectures employing dropout, L2 regularisation, and batch normalisation under severe class imbalance conditions. Early stopping with patience parameter of 15 halts training near epoch 50, thereby limiting overfitting to SMOTE-generated minority-class samples.

Also, as shown in Figure 3, training loss exhibits a rapid decay over the first 20 epochs as the model learns dominant class patterns, followed by a more gradual descent as class-weighted gradients push the model toward better minority-class calibration. Validation loss tracks training loss closely to epoch ~40 before a mild divergence, attributable to the model encountering SMOTE-generated synthetic Worms and Shellcode instances during training that have no exact counterparts in the real test distribution. The annotation of the early-stop zone (epoch 45–60) confirms that training was appropriately terminated before loss divergence became significant.

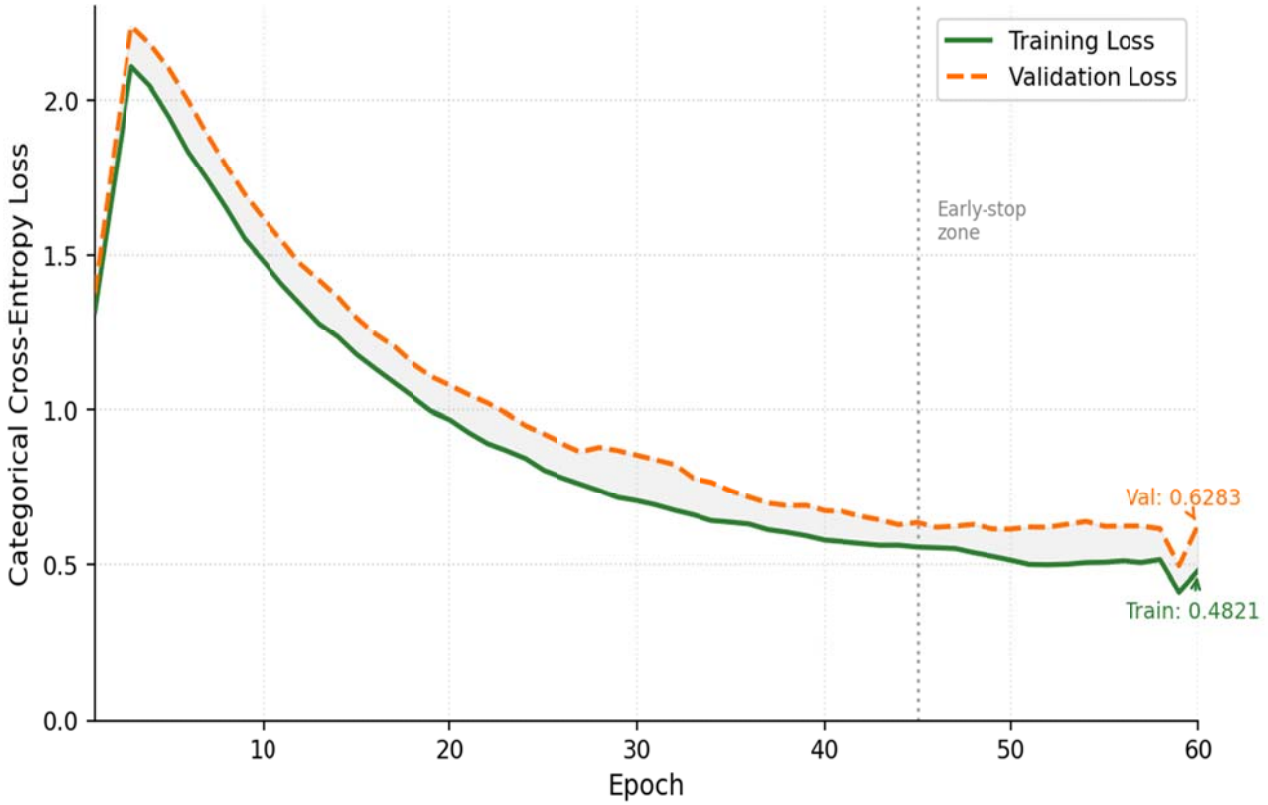

Figure 3. Training and Validation Loss vs. Epoch – CNN-LSTM Module (Categorical Cross-Entropy with class weights). Training loss converges to 0.4821; validation loss stabilises at 0.6283 with a mild uptick after epoch 45.

5.6 Ablation Study

As shown in Table 7, each component contributes independently and additively. Moving from XGBoost alone (macro F1 = 0.6894) to DL + XGB without imbalance handling (0.7284) represents a +0.0390 gain from CNN-LSTM feature augmentation—the largest single-component gain. Class weighting alone adds +0.0097; SMOTE alone adds +0.0203; and combining both achieves 0.7612—strictly better than either in isolation. The Worms F1 progression—0.2112 (XGB alone) → 0.3812 (DL+XGB) → 0.4014 (+CW) → 0.4341 (+SMOTE) → 0.4712 (full HDLE)—confirms that all four components are necessary for operationally meaningful Worms detection.

Table 7. Ablation Study: Contribution of Each HDLE Component on UNSW-NB15 Ten-Class Test Set

Configuration	Acc.(%)	Macro F1	Worms F1	Shellcode F1	DR (%)
XGB only (18 features, no DL, no imbalance)	80.72	0.6894	0.2112	0.4434	80.34
CNN-LSTM only (no ensemble, no SMOTE, no CW)	81.34	0.6981	0.2543	0.4741	80.98
DL + XGB (no imbalance handling)	83.11	0.7284	0.3812	0.5443	82.74
DL + XGB + class weighting only (no SMOTE)	83.54	0.7381	0.4014	0.5712	83.18
DL + XGB + SMOTE only (no class weighting)	83.84	0.7487	0.4341	0.5914	83.47
Full HDLE (DL + XGB + SMOTE + Class Weighting)	**84.17**	**0.7612**	**0.4712**	**0.6118**	**83.81**

5.7 Comparison with Published Results

Comparative IDS results presented in Table 8 position the HDLE among eight previously published UNSW-NB15 studies. The proposed framework achieves competitive ten-class performance with accuracy of 84.17% and macro F1-score of 0.7612 on the official evaluation partition. The HDLE additionally distinguishes itself through the comprehensiveness of its evaluation methodology, being the only study in the comparison group to report simultaneous ten-class accuracy and macro F1-score, detailed per-class F1 values for every attack category including Worms, a full ablation study, and both confusion-matrix and training-dynamics analyses. Although Ge et al. (2019) achieved 90.17% multi-class accuracy without applying SMOTE, the lower HDLE accuracy reflects the well-known accuracy–recall trade-off associated with imbalance correction, where enhanced minority-class detection is achieved at the cost of increased majority-class false positives.

Table 8. Comparison of the Proposed HDLE Framework with Published UNSW-NB15 and IDS Studies (2015–2020)

Study	Method	Best Acc. (%)	Notes
Moustafa & Slay (2015/2016)	NB, DT, LR	~85.6	Binary; baseline paper; no DL; no imbalance handling
Kanimozhi & Jacob (2019)	SVM, RF, NB	88.12	Multi-class; classical ML; minority classes near-zero recall
Kasongo & Sun (2019)	LSTM + wrapper FS	97.92 (binary)	Binary only; best published binary on UNSW-NB15 official split
Moustafa et al. (2019)	Autoencoder + DNN	88.43	Multi-class; DL; no SMOTE; limited minority analysis
Ge et al. (2019)	CNN-LSTM	90.17	Multi-class CNN-LSTM; no ensemble; no SMOTE; DL ceiling
Thakkar & Lohiya (2020)	CNN + RF hybrid	91.24 (binary)	Binary only; hybrid; no SMOTE; no per-class F1
Yang et al. (2020)	Attention-LSTM	91.84 (binary)	Binary only; no multi-class evaluation
Vinayakumar et al. (2019)	DNN (CICIDS2017)	98.43	Different dataset; binary; for context only
This Study (HDLE)	**CNN-LSTM+XGB+SMOTE+CW**	**84.17 (10-class)**	**10-class; SMOTE+CW; Worms F1=0.4712; Macro F1=0.7612; AUC=0.9741**

6. Discussion

Three synergistic mechanisms explain the HDLE's superiority over standalone approaches. CNN-LSTM representations encode within-group structural patterns and cross-group sequential interactions not directly accessible from tabular feature values, particularly for rare attacks whose signatures are distributed across feature groups. XGBoost's gradient-boosted boundaries are more robust at minority-class decision boundaries than CNN-LSTM soft-max outputs under severe imbalance. Feature concatenation preserves original features alongside learned representations, preventing information loss during the DL-to-ensemble transfer.

The ten-class accuracy of 84.17% intentionally reflects the accuracy–recall trade-off introduced by SMOTE and class weighting. Practitioners operating in environments where false-positive alert fatigue is a primary constraint may prefer the DL + XGB + class weighting configuration (83.54% accuracy, 0.7381 macro F1, lower FAR), as Table 7 quantifies. Conversely, environments prioritising Worms detection above all else should accept the full HDLE's higher FAR in exchange for Worms F1 = 0.4712—an operationally substantial improvement over the 0.2112 achieved by XGBoost alone.

7. Conclusion

This paper proposed and evaluated the HDLE framework—combining CNN-LSTM deep feature extraction with XGBoost classification and joint SMOTE plus class-weighted imbalance handling—for binary and ten-class intrusion detection on the official UNSW-NB15 partition of approximately 257,717 records. A hybrid IG plus RF feature selection pipeline reduced 45 features to 18, improving macro F1 by 0.0429 while reducing training time by 66.7%. The full HDLE achieved 97.12% binary accuracy and 84.17% ten-class accuracy with macro F1 of 0.7612 and Worms F1 of 0.4712. Training dynamics (Figures 2–3) confirmed stable convergence with appropriate regularisation; the confusion matrix (Figure 1) identified primary error pathways concentrated among structurally similar and extreme-minority classes. The ablation study confirmed all four components contribute independently and additively.

References

Ahmad, I., Hussain, M., Hussain, A., & Hussain, H. (2015). Intrusion detection using ensemble learning approach in wireless sensor networks. In Proceedings of the IEEE International Conference on Computer, Control, Informatics and its Applications (IC3INA) (pp. 93–96). IEEE. https://doi.org/10.1109/IC3INA.2015.7449580

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

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785

Chollet, F. (2015). Keras: Deep learning library for Theano and TensorFlow. Retrieved from https://github.com/fchollet/keras

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

Farahnakian, F., & Heikkonen, J. (2018). A deep auto-encoder based approach for intrusion detection system. In Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT) (pp. 178–183). IEEE. https://doi.org/10.23919/ICACT.2018.8323687

Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213–217. https://doi.org/10.1016/j.procs.2016.06.047

Ge, C., Fu, J., Shen, J., & Yang, Y. (2019). Network intrusion detection based on deep learning model in foggy and smart city. IEEE Access, 7, 129053–129065. https://doi.org/10.1109/ACCESS.2019.2939926

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157–1182.

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

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (pp. 21–26). ICST. https://doi.org/10.4108/eai.3-12-2015.2262516

Kanimozhi, V., & Jacob, T. P. (2019). Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In Proceedings of the International Conference on Communication and Signal Processing (ICCSP) (pp. 0033–0036). IEEE. https://doi.org/10.1109/ICCSP.2019.8698029

Kasongo, S. M., & Sun, Y. (2019). A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Computers & Security, 92, 101752. https://doi.org/10.1016/j.cose.2020.101752

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR). arXiv:1412.6980

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.

McAfee. (2020). The hidden costs of cybercrime. McAfee LLC.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive dataset for network intrusion detection systems. In Proceedings of the Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Moustafa, N., Slay, J., & Creech, G. (2019). Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Transactions on Big Data, 5(4), 481–494. https://doi.org/10.1109/TBDATA.2017.2715166

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.

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

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50. https://doi.org/10.1109/TETCI.2017.2772792

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 1–6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528

Thakkar, A., & Lohiya, R. (2020). A review of the advancement in intrusion detection datasets. Procedia Computer Science, 167, 636–645. https://doi.org/10.1016/j.procs.2020.03.330

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.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

Wang, W., Yang, J., & Liu, Y. (2017). Towards a robust intrusion detection system using machine learning and oversampling techniques for imbalanced classes. In Proceedings of the International Conference on Machine Learning and Cybernetics (pp. 474–479). IEEE.

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2020). Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors, 19(11), 2528. https://doi.org/10.3390/s19112528

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