Hierarchical Multi-Stage Intrusion Detection with Feature Inheritance and Prediction Verification

Dhuha A. Ali Kaream
Department of Computer Science
University of Baghdad
Baghdad, Iraq
doha.abd2301@sc.uobaghdad.edu.iq

Nada Hussien M. Ali
Department of Computer Science
University of Baghdad
Baghdad, Iraq
nada.husn@sc.uobaghdad.edu.iq

***Abstract*—One of the challenges faced by traditional intrusion detection systems based on machine learning or deep learning is instability when dealing with unbalanced network traffic, leading to failure in detecting certain attacks (minority classifications). Additionally, they struggle with multi-stage attacks, resulting in an increase in false alarms. This paper presents a hierarchical intrusion detection system supported by a Prediction Verification Layer (PVL) and a Feature Inheritance Mechanism (FIM). Where PVL contributes to documenting the system's final decision and increasing sensitivity to minority attacks, FIM also helps in inheriting features from previous layers and correcting errors as much as possible. Additionally, it allows for adaptive improvement without retraining the entire model. The results have shown a clear improvement in reducing false alerts, stabilizing attack detection, and increasing reliability with an accuracy of 78% and an F1-Score of approximately 77%.**

Keywords— Autoencoder Feature Extraction, Hieratical Intrusion Detection System (HIDS), Internet of Things (IoT) Security, Imbalanced Data Handling

I. INTRODUCTION

The rapid expansion of interconnected systems and intelligent services has significantly increased the attack surface of modern networks. As network traffic becomes more heterogeneous and behaviorally complex, traditional intrusion detection systems (IDS) struggle to provide reliable and consistent decisions. Recent deep-learning-based IDS approaches have reported high overall accuracy; however, they still suffer from unstable predictions and poor detection of minority attack categories due to overlapping behavioral patterns and severe class imbalance [1]. Intrusion detection (ID) systems still face two challenges despite growing attention. The heterogeneous structure of IoT traffic makes it difficult to extract meaningful features and design stable detection models, according to hybrid CNN–LSTM research on IoT datasets [2]. Many previous studies on deep-learning architectures like Attention-enhanced CNN–LSTM models and Transformer-based intrusion detection system (IDS) frameworks used reduced data subsets or failed to address imbalance, limiting their applicability to real-world IoT systems [3]. SMOTE, Borderline-SMOTE, and ADASYN were used to improve minority-class representation, but they can unintentionally create noisy samples when minority instances are close to complex decision boundaries. The UNSW-NB15 dataset [4] is used to propose a hierarchical intrusion detection framework. The proposed system divides detection into sequential layers with specialized functions. A third meta-classification layer verifies predictions using feature inheritance and probability validation after the first layer learns compact behavioral representations. Layered decision refinement reduces uncertainty and improves minority attack recognition without increasing model complexity. The actual contribution of this paper can be summarized by the usability of the feature inheritance mechanism from the previous layer and its correction, in addition to adding a layer to verify the final predictions, which contributed to increasing the detection opportunities for some minority attacks. The results showed that adding a third layer has the potential to validate the results of the previous layers, leading to increased reliability and stability compared to flat systems and two-layer hierarchical systems.

This paper's organization continues below. Section 2 discusses recent intrusion detection and hierarchical learning research. The hierarchical methodology and system architecture are detailed in Section 3. Experimental results and performance analysis are in Section 4. Section 5 concludes the paper and suggests future research.

II. LITERATURE REVIEW

Modern cybersecurity systems use Intrusion Detection Systems (IDSs) to monitor network traffic and system operations for unusual behavior and intrusion attempts. Traditional multi-category classification has many drawbacks, but many studies use it. Standard samples dominate the data, and attacks are rare. Type II errors—ignored attacks—are more likely. Grouping all attack types into one stage reduces the system's scalability as categories increase or patterns become more complex. One study by Alashjaee [1] used Attention–Convolutional Neural Network–Long Short-Term Memory (Attention– CNN–LSTM) and achieved high accuracy on Bot-IoT and NSL-KDD datasets. Due to its flat architecture, the model was limited. Similarly, Le [2] examined hybrid resampling strategies such as SMOTE–Tomek and Borderline-SMOTE-ENN combined with ensemble classifiers. The reported F1-score improved on IoTID20 and CHADC2020; however, the improvement was closely tied to the specific dataset distribution and did not consistently transfer to heterogeneous traffic environments. A related outcome was observed by Al-Shehari [5], where CNN models supported by ADASYN and SMOTE produced better class balance but still operated within a flat classification structure, which made minority attacks unstable when the imbalance became severe.

To overcome these limitations, several studies adopted hierarchical IDS designs in which detection is performed through multiple decision stages instead of a single classification step. This separation helps reduce confusion among attack families, yet it introduces another practical concern — errors generated in early layers tend to propagate forward and affect later decisions. As a result, the expected advantage of hierarchy is sometimes weakened. Cyber-Kill-Chain-based approaches illustrate this situation: they improve interpretability but remain sensitive to early misclassification. More recent research integrates generative models and explain ability mechanisms, such as CTGAN and XAI-based detection [6], and reports high accuracy. In

practice, however, these methods mainly enhance feature representation rather than the reliability of the final decision, and classifier confidence varies across attack categories when large-scale data is used. Consequently, the unresolved issue is not limited to class imbalance alone but also involves maintaining decision stability across hierarchical stages. Oversampling techniques, including Borderline-SMOTE and ADASYN, increase minority visibility yet often introduce noisy boundary samples. Optimization-guided variants such as GWO- or PSO-based sampling improve sample placement to some extent, but their interaction with hierarchical decision flow has not been adequately examined. Therefore, despite encouraging detection performance reported in previous studies, many solutions still face scalability constraints, depend on flat decision behavior, and struggle to recognize minority attacks reliably. Moreover, the combined effect of imbalance handling strategies and hierarchical architectures has not been thoroughly investigated on realistic large-scale datasets. These limitations motivate the development of a more robust hierarchical intrusion detection framework capable of progressively refining decisions across multiple stages while enhancing overall detection robustness.

III. METHODOLOGY

In this section, review the phases that followed to build the proposed IDS system in this study.

A. Phase1: Pre-Processing Dataset:

In this phase, the UNSW-NB15 dataset undergoes a structured pre-processed pipeline to ensure data consistency eliminate bias, and prepare the traffic feature for machine learning and deep learning

The steps that followed in this phase:

1. Remove duplicated recorded and check of miss value
2. Encoding categorical feature using One-Hote-Encoding [7]
3. Normalization dataset by using min-max technique [8].

B. Phase2: Data Balancing:

In this phase, address the imbalanced issue that suffers it the UNSW-NB15 dataset. This problem loads the model bias toward majority classes and increases the False alarm rate. To address this issue, a data balancing strategy was applied only to the training set in order to avoid data leakage. First, the network traffic features were compressed into a lower-dimensional latent representation using an Autoencoder [8]. Performing balancing in the latent space helps generate more realistic synthetic samples while preserving the intrinsic structure of the data. After that, a multiclass Borderline-SMOTE algorithm [9] was applied to generate new samples near the decision boundary of minority attack classes, since these samples are the most informative for classification. To further improve data quality, Edited Nearest Neighbors (ENN) [2], [10] was used to remove noisy or ambiguous generated instances. The hyperparameters used to train the Autoencoder for latent feature extraction are summarized in Table 1.

TABLE I. HYPERPARAMETER CONFIGURATION OF THE AUTOENCODER

Component	Parameter	Value
Architecture	Encoder Layers	256 → 128 → 32
	Decoder Layers	32 → 128 → 256
Latent Space	Dimension	32
Activation	Hidden Layers	ReLU
	Output Layer	Sigmoid
Loss Function	Reconstruction Loss	Mean Squared Error (MSE)
Optimizer	Adam	Learning rate = 0.001
Training	Batch Size	512
	Max Epochs	100
	Early Stopping	patience = 6

The configuration of the data balancing stage, including Borderline-SMOTE oversampling and ENN cleaning, is presented in Table 2.

TABLE II. PARAMETER SETTINGS OF BORDERLINE-SMOTE AND ENN

Technique	Parameter	Value
Borderline-SMOTE	k_neighbors	5
	Sampling Strategy	Equalize minority attack classes (≈ 8,000 samples per class)
	Random State	42
ENN Cleaning	n_neighbors	3
	Purpose	Remove noisy synthetic samples near decision boundary

C. Phase 3: Hierarchal Intrusion detection system (proposal model):

1) Stage1: Layer One

The first layer is a binary intrusion detection model trained to distinguish normal from malicious traffic. Rather than acting as a final classifier, it learns general attack characteristics that can be reused by the next stage. A fully connected DNN [11] is trained on the preprocessed UNSW-NB15 features. The network compresses the input into a low-dimensional bottleneck representation, which is transferred to Layer-2 for multi-class attack categorization.

Training uses a stratified split with early stopping based on validation AUC, and the decision threshold is selected to prioritize attack recall. Table 3 presents the hyperparameter configuration of Layer-1.

TABLE III. MODEL HYPERPARAMETERS – LAYER-1

Parameter	Value
Input Features	196
Hidden Layers	Fully Connected
Bottleneck Size	64
Activation	ReLU
Output Activation	Sigmoid
Loss Function	Binary Cross-Entropy
Optimizer	Adam
Batch Size	1024
Epochs	100
Early Stopping	Based on Validation AUC
Metric	AUC, Recall, Precision
Threshold Selection	Recall-prioritized tuning

2) Stage2: Layer Two

The second layer performs multi-class attack categorization using the latent representations learned by Layer-1. Instead of relying on raw network features, the model operates on the bottleneck embeddings, which encode high-level behavioral patterns of malicious traffic.

The classifier operates on the transferred representations instead of raw features, which helps separate similar attack types and reduces noise. A fully connected network is trained to classify the UNSW-NB15 attack categories using a stratified split with early stopping based on validation AUC, and the best-performing model is used for evaluation (Table 4)

TABLE IV. MODEL HYPERPARAMETERS – LAYER-2 (MULTI-CLASS CLASSIFIER)

Parameter	Value
Input Features	64 (Layer-1 bottleneck embeddings)
Hidden Layers	Fully Connected
Hidden Units	128 → 64
Batch Normalization	Applied after each hidden layer
Dropout	0.3
Activation	ReLU
Output Layer	SoftMax
Number of Classes	9 Attack Categories
Loss Function	Categorical Cross-Entropy
Optimizer	Adam
Learning Rate	0.001 (with decay during training)
Batch Size	1024
Max Epochs	100
Early Stopping	Based on Validation AUC
Evaluation Metrics	Accuracy, AUC, Precision, Recall, F1-Score

3) Stage3: Layer Three

This layer is used to verify the decision of Layer-2 rather than to classify independently. The class probabilities from Layer-2 are combined with the latent features obtained from Layer-1 and passed to a fully connected network to re-evaluate the prediction, especially for similar attack patterns. The model is trained using sparse categorical cross-entropy with early stopping and learning-rate reduction to avoid overfitting. The training settings are listed in Table 5.

TABLE V. HYPERPARAMETERS OF LAYER-3 CONFIDENCE MODE

Parameter	Value
Input features	73 (64 latent + 9 probabilities)
Hidden layers	256 → 128 → 64
Activation	ReLU
Output	Softmax (9 classes)
Loss function	Sparse Categorical Cross-Entropy
Optimizer	Adam
Learning rate	0.001
Batch size	1024
Epochs (max)	80
Early stopping	patience = 12
LR scheduler	ReduceLROnPlateau
Regularization	Dropout + BatchNorm

Algorithm 1 outlines the processing steps of the proposed hierarchical IDS.

Algorithm1: Proposed Hierarchical IDS Algorithm
Input: D = UNSW-NB15 Dataset (Raw Data)
Output: Final Verified Attack Prediction
Begin 1. Pre-process data (cleaning, encoding, normalization) and extract latent features using encoder. 2. Balancing dataset using Borderline-SMOTE + ENN. 3. Train Layer-1 to detect *Normal vs Attack* traffic. 4. Pass attack samples to Layer-2 to classify attack category (feature inheritance). 5. Pass Layer-2 outputs to Layer-3 to verify and refine attack classification. 6. Evaluate system using Accuracy, Precision, Recall and F1-score. End.

D. Data splitting and Evaluation

The dataset was stratified into training (70%), validation (15%), and testing (15%) sets to ensure fair evaluation. The proposed IDS was evaluated using Accuracy, Precision, Recall, and F1-score metrics across all classes.

IV. RESULT AND DATA ANALYSIS

A. Experimental Setup Recap:

Experiments were conducted using Google Colab Pro (GPU-accelerated) and a local machine (Intel i7, 16–32 GB RAM, Windows 10). The implementation was developed in Python using TensorFlow/Keras and Scikit-learn, with Pandas and NumPy for data handling and Matplotlib for visualization.

B. Dataset preprocessing Result:

Table 6 shows the details of the raw dataset before any processing and after Pre-processing Phase.

TABLE VI. THE UNSW-NB15 DETAILS

State	Number of Rows	Number of Columns	Number of Attack Types
Before	257,673	45	9 classes
After	257,673	198	9classes

C. Dataset Balancing Result:

Fig. 1 shows comparison of class distributions before balancing, after Borderline-SMOTE, and after SMOTE-ENN cleaning in the latent feature space.

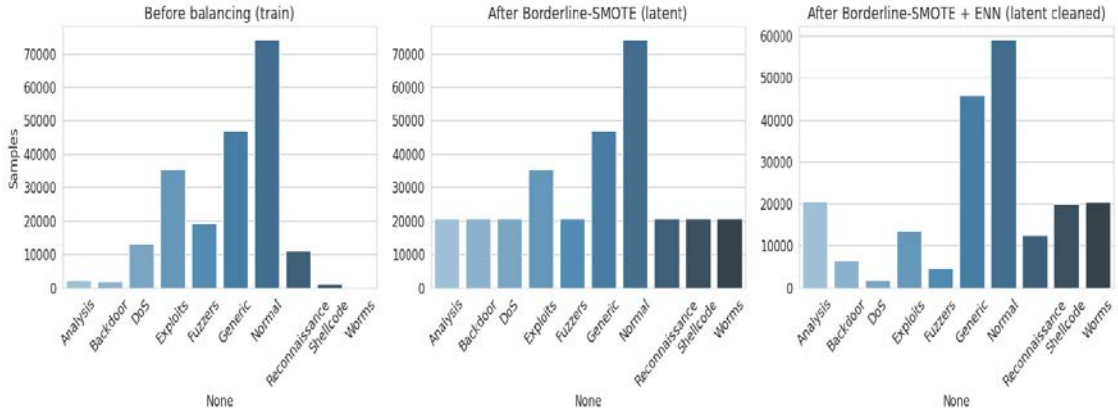

Fig. 1. The UNSW-NB15 Dataset's Class Distribution 4.4 The Proposal Model Result

The performance of the proposed hierarchical model is summarized in Table 7. The results show an overall improvement in accuracy, precision, recall, and F1-score after applying latent-space balancing and the meta-classification layer. The hierarchical learning stage enhanced the detection capability while maintaining stable performance across the majority classes.

TABLE VII. PERFORMANCE IMPROVEMENT ACROSS HIERARCHICAL IDS LAYERS

Model Stage	Accuracy	Precision	Recall	F1-Score	AUC
Baseline (Raw)	0.730	0.711	0.720	0.700	0.911
After Autoencoder	0.750	0.740	0.750	0.740	0.942
After Balancing	0.770	0.760	0.770	0.753	0.964
After two layers	0.780	0.770	0.780	0.750	0.970
Proposed model	**0.787**	**0.777**	**0.787**	**0.756**	**0.978**

Fig. 2 and 3 show Hierarchical refinement enhances classification performance compared to the standalone Layer-2 model.

	precision	recall	f1-score	support
0	0.792453	0.125749	0.217054	334.000000
1	0.000000	0.000000	0.000000	291.000000
2	0.455621	0.037671	0.069589	2044.000000
3	0.597277	0.922206	0.725000	5566.000000
4	0.851333	0.821841	0.836327	3031.000000
5	0.996662	0.973774	0.985085	7359.000000
6	0.768681	0.676387	0.719586	1749.000000
7	0.651376	0.375661	0.476510	189.000000
8	0.600000	0.136364	0.222222	22.000000

Fig. 2. Confusion Matrix of Layer Two

	precision	recall	f1-score	support
0	0.7692	0.1497	0.2506	334
1	0.3333	0.0069	0.0135	291
2	0.4444	0.0881	0.1470	2044
3	0.6103	0.9037	0.7286	5566
4	0.8238	0.8406	0.8321	3031
5	0.9974	0.9740	0.9856	7359
6	0.7808	0.6558	0.7129	1749
7	0.6636	0.3862	0.4883	189
8	0.3750	0.1364	0.2000	22

Fig. 3. Confusion Matrix of Proposal Model

The results show that the hierarchical architecture improves prediction stability by refining the decisions of the multi-class classifier using latent features and confidence information. Although minority classes still have lower recall due to dataset imbalance, the overall performance became more consistent compared to a single-stage model. This confirms that multi-stage learning enhances detection reliability in realistic multi-class intrusion scenarios.

V. CONCLUSION

This work introduced a hierarchical intrusion detection system enhanced by feature inheritance and an additional third classification layer. The proposed structure improved decision reliability and significantly enhanced the detection of minority attack classes on the UNSW-NB15 dataset, demonstrating better robustness than conventional single-stage models. In future work, the model will be evaluated on live network traffic and different threshold settings will be examined to observe their effect on prediction stability.

REFERENCES

[1] M. Alashjaee, "Deep learning for network security: an Attention-CNN-LSTM model for accurate intrusion detection," Scientific Reports, 2025.

[2] Shin, M. Kim, and H. Kim, “Towards unbalanced multiclass intrusion detection with hybrid sampling methods and ensemble classification”, Applied Soft Computing, Vol. 157, pp. 111517, 2024.

[3] M. A. Uddin, S. Aryal, M. R. Bouadjenek, M. Al-Hawawreh, and M. A. Talukder, “Hierarchical classification for intrusion detection system: Effective design and empirical analysis”, Ad Hoc Networks, pp. 103982, 2025.

[4] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems,”," Australia, 2015. [Online]. Available: //research.unsw.edu.au/projects/unsw-nb15-dataset.

[5] T. Al-Shehari, M. Kadrie, M. N. Al-Mhiqani, T. Alfakih, H. Alsalman, M. Uddin, S. S. Ullah, and A. Dandoush, “Comparative evaluation of data imbalance addressing techniques for CNN-based insider threat detection”, Scientific Reports, Vol. 14, No. 1, pp. 24715, 2024.

[6] A. A. a. F. Bajaber, "An Intrusion Detection System over the IoT Data Streams Using Explainable AI," Sensors, 2025.

[7] H. C. Altunay and Z. Albayrak, “A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks”, Engineering Science and Technology, an International Journal, Vol. 38, pp. 101322, 2023.

[8] Y. C. a. H. L. H. Xu, "Autoencoder-based feature extraction for intrusion detection in IoT networks," Computers & Security, 2021.

[9] Z. W. a. Y. Z. S. Liu, "Adaptive Borderline Oversampling for high-imbalance learning,," Information Sciences,, 2022.

[10] Wilson, D. L., “Asymptotic Properties of Nearest Neighbor Rules Using Edited Data,” IEEE Transactions on Systems, Man, and Cybernetics, 1972

[11] L. W. a. Y. L. J. Zhang, ""Hybrid PSO–Autoencoder for Feature Reduction in Deep IDS," IEEE, 2025.