CatBoost Integration for Intrusion Detection in Evolving IoT Environments

Jay Mark V. Agsoy
Bachelor of Science in Computer Science
University of Mindanao
Matina, Davao City, 8000
j.agsoy.545281@umindanao.edu.ph

Neil Robert J. Morales
Bachelor of Science in Computer Science
University of Mindanao
Matina, Davao City, 8000
n.morales.544543@umindanao.edu.ph

ABSTRACT
This study evaluates the performance and cross-domain robustness of the CatBoost algorithm compared with baseline models from Fathima et al., including Decision Tree, Random Forest, and Gradient Boosting. Using the UNSW-NB15 and CICIoT2023 datasets, along with a combined multi-domain "Master" dataset, we analyze model behavior under both in-domain and cross-domain configurations. In in-domain evaluations, CatBoost demonstrates superior discriminative capability, achieving 90.70% accuracy and 92.84% F1-score on UNSW-NB15, and 98.85% accuracy and 99.29% F1-score on CICIoT2023. However, under cross-domain evaluation, highly optimized ensemble methods experience significant performance degradation, with CatBoost's AUC dropping to 14.81% in the UNSW-to-CICIoT configuration. In contrast, simpler baseline models such as Decision Trees show greater resilience to domain shifts by relying on more stable decision boundaries. Finally, experiments with multi-domain training on the combined Master dataset demonstrate that domain sensitivity can be mitigated, restoring CatBoost's robustness across all test scenarios with 94.38% accuracy, 96.10% F1-score, and 99.25% AUC.

Categories and Subject Descriptors
Security and privacy→Intrusion/anomaly detection and malware mitigation→Intrusion detection systems

General Terms
Algorithms, Performance, Reliability, Experimentation, Security.

Keywords
Intrusion Detection; Model Decay; Data Drift; IoT Security; Categorical Encoding.

1. INTRODUCTION

1.1 Background of the Study
Effective Intrusion Detection Systems (IDS) must generalize across evolving network environments. While Fathima et al. [1] provide a robust baseline using the 2015 UNSW-NB15 dataset [2]-[6], decade-old data often fails to represent modern attack behaviors [7]. This study addresses this limitation by transitioning to the contemporary CICIoT2023 dataset [8], [9] and evaluating CatBoost's native categorical handling [10] to enhance model robustness against distribution shifts.

Both datasets contain high-cardinality categorical attributes (e.g., *proto*, *service*, *state*). Traditional models, like the Fathima et al. [1] baseline, rely on One-Hot Encoding (OHE). This causes a high-dimensional feature explosion, leading to the curse of dimensionality and significant computational overhead [11]. CatBoost resolves this using Ordered Target Statistics, processing thousands of categories natively without inflating the feature space or losing hierarchical data [10].

Recent studies highlight the need for adaptable models but often rely on heavy preprocessing. Yan, Zhou, and Chen [12] successfully cross-evaluated UNSW-NB15 and CICIoT2023 to detect zero-day attacks, but their contrastive learning approach required computationally expensive deep learning embeddings. This reliance on massive computational overhead is also evident in the work of Gulzar and Mustafa [13], who achieved 99.82% accuracy on the CICIoT2023 dataset using a complex DeepCLG hybrid model; however, this success depended heavily on resource-intensive deep learning ensembles and exhaustive Boruta feature selection. Similarly, while Hajjouz & Avksentieva [14] applied CatBoost to CICIoT2023, their methodology depended heavily on complex, external feature selection pipelines to manage dimensionality.

The challenge of model generalization across different network environments was explicitly demonstrated by Al-Riyami, Lisitsa, and Coenen [15]. They performed a cross-dataset evaluation using the legacy NSL-KDD and gureKDD datasets, revealing a drastic drop in model performance when transitioning between networks. Their study traced this failure to mismatched categorical features, specifically noting that variations in the service attribute caused traditional models to collapse. To resolve this and improve accuracy, the researchers had to rely on manual data relabeling and mapping to force compatibility between the datasets.

This study explores CatBoost's native categorical handling as a standalone, lightweight mechanism to resist distribution shifts without exhaustive preprocessing. Furthermore, there is a need to evaluate these optimized models under the hardware constraints of developing IT sectors. Locally, this research is urgent because the rapid, unregulated deployment of IoT devices in local institutions is outpacing their current cybersecurity infrastructures.

Based on the gaps identified in Table 1, this proposed model addresses the need for a modern, lightweight IDS that can seamlessly process high-cardinality network data without the computational penalty of traditional encoding or heavy preprocessing. CatBoost preserves the structure and statistical properties of network protocols without additional preprocessing, making it theoretically more robust against the

categorical distribution shifts seen between traditional networks (UNSW-NB15) and contemporary IoT environments (CICIoT2023).

Table 1: Related Studies

Ref No.	Year	Algorithm	Dataset	Summary	Limitation
[1]	2023	Random Forest, SVM, Logistic Regression, Gradient Boosting, Decision Trees, KNN	UNSW-NB15	Strong baselines for general network intrusion detection.	Relies on outdated datasets lacking modern IoT traffic.
[14]	2024	CatBoost	CICIoT2023	High DDoS/Mirai accuracy via advanced preprocessing.	Prioritizes external feature selection over CatBoost's native Ordered TS.
[12]	2025	Contrastive Learning	UNSW-NB15 & CICIoT2023	Zero-day detection using contrastive feature embeddings.	Uses costly DL embeddings without testing cross-dataset generalizability.
[13]	2025	DeepCLG Hybrid Model (CNN, LSTM, GRU + Capsule Network)	CICIoT2023	Introduced Hybrid DL and Boruta for IIoT (99.82% accuracy).	Depends on resource-intensive DL ensembles and exhaustive feature selection.
[15]	2021	ML/DL Ensembles, (Random Forest, CNN, AdaBoost, etc.)	NSL-KDD & gureKDD	Highlights cross-dataset drops from categorical mismatch.	Evaluates older data and requires manual categorical relabeling.

1.2 Purpose and Description

This study addresses the data drift in network security by evaluating the robustness of CatBoost's native categorical handling. As network paradigms shift from traditional environments [1] to modern IoT testbeds [9], existing One-Hot Encoding-based models suffer from feature explosion and performance decay. This research contributes to SDG 9 (Industry, Innovation, and Infrastructure) by enhancing the resilience of digital security frameworks and SDG 16 (Peace, Justice, and Strong Institutions) by keeping institutional data integrity against evolving threats. The primary beneficiaries are Security Operations Center (SOC) analysts and network administrators within local educational and enterprise sectors, who require models that remain reliable despite evolving attack signatures, and future researchers seeking a standardized mapping between legacy and IoT datasets.

The proposed project employs CatBoost to automate the detection of malicious traffic across heterogeneous network datasets. By utilizing Ordered Target Statistics, the system aims to maintain high predictive accuracy without the computational overhead typical of high-cardinality feature sets.

1.3 Objectives

1.3.1 General Objectives

To evaluate the impact of native categorical handling on the robustness and generalizability of network intrusion detection systems by implementing the baseline architectures from Fathima et al. [1] alongside CatBoost to quantify model performance and decay when transitioning between the UNSW-NB15 and CICIoT2023 datasets.

1.3.2 Specific Objectives

To achieve the general objective, the following specific tasks will be undertaken:

1.3.2.1 To implement the six machine learning architectures identified in the baseline study by Fathima et al. [1] on the UNSW-NB15 dataset to serve as a benchmark for modern IDS performance.

1.3.2.2 To integrate CatBoost into the comparative framework, utilizing its native categorical processing to optimize high-cardinality features.

1.3.2.3 To create a unified feature set that contains features that both UNSW-NB15 and CICIoT2023 have.

1.3.2.4 To conduct a cross-evaluation between models trained in UNSW-NB15 and CICIoT2023 to measure model decay.

1.3.2.5 To evaluate the dataset-agnostic [16] representation capabilities of the implemented architectures, in a comparative sense, to determine whether native categorical handling allows for a more stable transfer of predictive logic between traditional and IoT network paradigms.

1.4 Scope and Limitations

The scope of this study focuses on evaluating the integration of Categorical Boosting (CatBoost) into an Intrusion Detection System (IDS) framework to determine its capacity to maintain predictive performance across evolving network environments. The primary beneficiaries and target users of this framework are cybersecurity researchers, network administrators, and security operations center (SOC) analysts who require adaptable models for threat detection. Conducted within the current academic timeframe, the study is delimited to comparing CatBoost against six traditional, One-Hot Encoding-dependent baseline models established by Fathima et al. [1]. The core feature of the proposed "system" is a comparative machine learning pipeline that processes network

flow data through a unified feature space mapped from two distinct paradigms: the traditional UNSW-NB15 dataset [2] and the contemporary CICIoT2023 dataset [9]. In terms of user interaction, a security analyst would theoretically deploy this model to ingest offline network traffic logs, allowing the system to natively process high-cardinality categorical features (like protocol and service) to classify the traffic as either benign or malicious. Furthermore, the scope of the predictive modeling is restricted to binary classification, distinguishing exclusively between benign traffic and generalized malicious activity, rather than identifying specific attack types.

This study is heavily dependent on the inherent quality, labeling accuracy, and feature availability of the external UNSW-NB15 [2] and CICIoT2023 [9] datasets; any underlying biases in these open-source repositories will directly influence the model's learning phase. Furthermore, because the CICIoT2023 dataset [9] will be sourced from a pre-processed Kaggle repository to accommodate computational and time constraints, the study relies on the integrity of that third-party preprocessing. Finally, the system's performance is limited to the specific overlapping attributes shared between the two datasets; therefore, the model's predictive logic may not fully generalize to entirely novel network protocols or zero-day vulnerabilities outside the bounds of these specific historical and IoT testbed environments.

2. METHODS

This section outlines the experimental methodology used to develop the CatBoost-based Intrusion Detection System, detailing the dataset acquisition, feature engineering pipeline, algorithm architecture, and evaluation metrics.

2.1 Conceptual Framework

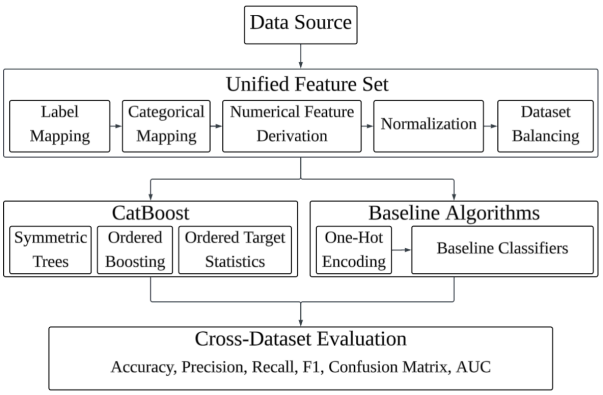

Figure 1: Conceptual Framework

2.2 Data Gathering

This study utilizes comprehensive network traffic data to evaluate the proposed CatBoost-based intrusion detection system across diverse network paradigms and time periods.

To ensure a standardized evaluation, the collected data will be mapped to a unified feature space, extracting core overlapping attributes such as flow duration, packet counts, and byte rates. Baseline models will use One-Hot Encoding (OHE) for categorical features, while CatBoost will process them natively through Ordered Target Statistics, preserving the statistical relationships of high-cardinality attributes [10], [17].

2.2.1 Data Sources

2.2.1.1 UNSW-NB15

Developed at the Australian Centre for Cyber Security using the IXIA PerfectStorm tool, this dataset captures modern network traffic with nine categories of attacks, including Fuzzers, Backdoors, and Shellcode. The version provided by the University of New South Wales will be used [2], [3], [5].

2.2.1.2 CICIoT2023

Developed by the Canadian Institute of Cybersecurity, this dataset focuses on IoT network traffic, including benign flows and attacks such as DDoS, DoS, reconnaissance, and data exfiltration. The pre-processed version from Kaggle by user himadri07 will be employed [9].

2.3 CatBoost

CatBoost (Categorical Boosting) is chosen for this study due to its specialized handling of categorical features and its robust resistance to overfitting, which is critical when generalizing across different network datasets.

Table 2: Sample CatBoost Target Encoding

Position (Time)	Flow ID	Proto	Target (Label)	Encoded Proto
1	1001	TCP	1.0	0.00
2	1002	TCP	0.0	1.00
3	1003	UDP	1.0	0.00
4	1004	TCP	1.0	0.50
5	1005	UDP	1.0	1.00
6	1006	UDP	0.0	1.00

2.3.1 Symmetric Trees

CatBoost uses symmetric trees or oblivious decision trees as its base predictors. In these trees, the same splitting feature and threshold are applied to all nodes at a given depth [10].

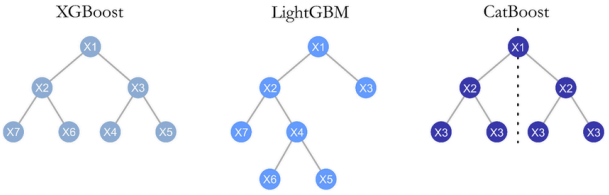

Figure 2: Tree Structures of Gradient Boosting Algorithms

2.3.2 Ordered Boosting

Standard boosting algorithms use the same data points to calculate the gradient and train the next tree, which introduces a statistical bias [10]. As such, they suffer from a problem called prediction shift, where the statistical properties of the

data a model is tested on differ from the data it was trained on, leading to a degradation in performance [18]. To address this, CatBoost implements a technique called Ordered Boosting [10].

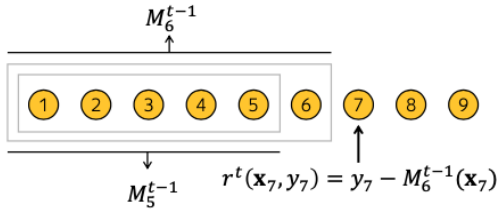

Figure 3: Ordered Boosting prediction for sample x_7 using only preceding samples

2.3.3 Ordered Target Statistics

Standard target encoding methods often induce target leakage by using a sample's own label to compute its categorical feature representation. CatBoost instead relies on the principle of ordering, where samples are gathered sequentially by introducing an artificial "time" via a random permutation of the dataset [10].

$$\hat{x}_i^k = \frac{\Sigma_{j \in S_i} [\, x_j^k = x_i^k \,] \cdot y_j + a \cdot p}{\Sigma_{j \in S_i} [\, x_j^k = x_i^k \,] + a}$$

Figure 4: Ordered Target Statistics

By restricting the encoding calculation strictly to past observations, this mechanism inherently prevents the model from cheating and memorizing specific target labels.

2.4 Unified Feature Set

2.4.1 Preprocessing & Feature Engineering

2.4.1.1 Label Mapping

The target variables across both datasets were transformed into a binary classification format. The native, multi-class labels for attack categories and benign traffic types were mapped to a universal binary label.

2.4.1.2 Categorical Feature Mapping

Categorical features required alignment to resolve discrepancies in nomenclature and distribution between the UNSW-NB15 and CICIoT2023 datasets. As a preliminary step, all categorical variables were explicitly cast to string data types to ensure consistent handling during subsequent encoding.

Overlapping features between the datasets were consolidated by retaining only shared categories and grouping all others into a unified "other" class. For Protocol, only tcp and udp were preserved as explicit categories, with all remaining protocols aggregated as "other." For Connection State, only "fin" and "rst" were maintained, while all other states were reclassified as "other." For Service, overlapping services were retained and standardized (e.g., mapping HTTP to http and HTTPS to ssl), and any non-overlapping services unique to a single dataset were grouped under "other."

2.4.1.3 Numerical Feature Derivation

The UNSW-NB15 and CICIoT2023 datasets were developed independently, resulting in limited overlap between their native feature sets. As a result, the unified feature set can only incorporate general network flow characteristics that are consistently defined and shared across both datasets.

To establish a unified feature space for cross-dataset evaluation, feature engineering was applied to bridge this gap. Several metrics explicitly defined in the CICIoT2023 dataset were mathematically derived from the raw packet and byte statistics available in the UNSW-NB15 dataset, ensuring consistent semantic interpretations across both environments.

Table 3: Cross-Dataset Numerical Feature Derivation

Feature Name	CICIoT2023	UNSW-NB15
Duration	flow_duration	duration
Overall Packet Rate	rate	(spkts + dpkts) / duration
Source Packet Rate	srate	spkts / duration
Destination Packet Rate	drate	dpkts / duration
Average Bytes per Packet	avg	(sbytes + dbytes) / (spkts + dpkts)
Total Transfer Bytes	tot_sum	sbytes + dbytes
Weight	weight	spkts * dpkts

2.4.1.3 Distribution Matching & Class Balancing

To prevent statistical bias during the training phase of cross-dataset evaluation, the CICIoT2023 training set was undersampled to strictly align with both the total instance count and the class ratio (55:45) of the UNSW-NB15 training baseline. For the testing phase, CICIoT2023 was sampled to match the UNSW-NB15 test instance count; however, due to the native scarcity of benign traffic within the raw CICIoT2023 dataset, the testing ratio naturally skewed toward malicious instances (81:19).

2.4.1.4 Master Dataset

To assess model generalization across diverse network traffic patterns and IoT attack scenarios, a master dataset was constructed by concatenating the UNSW-NB15 and CICIoT2023 datasets. Specifically, the training split of UNSW-NB15 was combined with the training split of CICIoT2023 to form the master training set. Similarly, the respective test splits of both datasets were merged to create the master test set.

2.4.1.5 Normalization

The MinMaxScaler from scikit-learn was used, as in the baseline study, to linearly transform each feature within the (0, 1) range. This prevents features with larger magnitudes from disproportionately affecting the models. The transformation is calculated as follows:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Figure 5: MinMaxScaler Formula

To ensure consistent representation of the global minimum and maximum values across all datasets, the MinMaxScaler was fitted exclusively on the master training set. Subsequently, the UNSW-NB15, the CICIoT2023 training and testing sets, as well as the master test set, were scaled using the parameters derived from the master training set.

2.4.1.6 One-Hot Encoding

A one-hot encoded version is constructed on all datasets to represent categorical values for the baseline algorithms. The non-one-hot encoded versions were retained for CatBoost.

2.4.2 Hyperparameters

To establish a fair comparative environment, hyperparameter tuning was conducted for both the proposed CatBoost architecture and the traditional machine learning baselines [1]. The objective was to empirically replicate the performance metrics reported in the baseline study.

The optimization process identified performant configurations for all models, but the empirical replication yielded lower metrics. The baseline study [1] reported 99% Accuracy and 98% F1-Score for Random Forest, while the optimized replication achieved 97.21% Accuracy and 98.14% F1-Score. The optimal hyperparameters and their corresponding replication scores are provided in Table 4.

Table 4: Hyperparameters & Replicated Baseline Metrics

Algorithm	Optimal Hyperparameters	Replicated Accuracy	Replicated F1-Score
CatBoost (Proposed)	iterations=1000, learning_rate=0.1, depth=6, l2_leaf_reg=5	96.00%	96.08%
SVM	kernel='rbf', C=1	97.26%	98.20%
Random Forest	n_estimators=100, max_depth=10, min_samples_split=2, min_samples_leaf=4, max_features='log2'	97.21%	98.14%
Logistic Regression	solver='liblinear', C=10	97.03%	98.05%
Gradient Boosting	n_estimators=100, learning_rate=0.01, max_depth=10	96.80%	97.87%
Decision Tree	criterion='gini', max_depth=20, min_samples_split=20	95.61%	97.06%
KNN	weights='uniform', n_neighbors=3, metric='manhattan'	95.33%	96.86%

2.5 Evaluation Metrics

Evaluation metrics are quantitative measures used to assess the performance of a model. In classification problems, these metrics help determine how well the model predicts the correct labels and how effectively it distinguishes between different classes. The following metrics are commonly used to evaluate classification models.

2.5.1 Accuracy

Accuracy measures the overall correctness of the model. It is defined as the ratio of correctly predicted instances to the total number of instances [19].

$$Accuracy = \frac{TP + TN}{TP+TN+FP+FN}$$

Figure 6: Accuracy Formula

2.5.2 Precision

Precision measures how many of the predicted positive instances are actually positive. It focuses on the quality of positive predictions [20].

$$Precision = \frac{TP}{TP+FP}$$

Figure 7: Precision Formula

2.5.3 Recall

Recall, also known as Sensitivity or True Positive Rate, measures how many actual positive instances were correctly identified by the model [20].

$$Recall = \frac{TP}{TP+FN}$$

Figure 8: Recall Formula

2.5.4 F-1 Score

The F1-Score is the harmonic mean of Precision and Recall. It provides a balance between both metrics, especially when there is an uneven class distribution [21].

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figure 9: F1-Score Formula

2.5.5 Area Under the Curve

Area Under the Curve (AUC) of a Receiver Operating Characteristic (ROC) curve serves as a scalar measure of a model's discriminative capacity, quantifying its ability to distinguish between binary classes across all possible thresholds [22].

This metric is derived by plotting the True Positive Rate against the False Positive Rate, where an AUC of 1.0 signifies perfect classification, 0.5 denotes performance equivalent to random chance or guessing, and values below 0.5 indicate poor classification capability [22].

2.5.6 Confusion Matrix

The Confusion Matrix is a standardized evaluation framework utilized to analyze the classification performance of a model by systematically comparing actual labels against predicted outputs [23].

By categorizing results into True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), this matrix provides a granular perspective on specific classification errors and serves as the fundamental basis for deriving higher-level metrics such as Accuracy, Precision, Recall, and the F1-Score [23].

3. RESULTS

The results will be divided by the dataset each model was tested on.

Table 5: All Models on UNSW test

Model	Acc (%)	Pr (%)	Re (%)	F1 (%)	AUC (%)
CatBoost UNSW	90.70	97.58	88.53	92.84	98.33
CatBoost CICIoT	64.75	78.59	66.27	71.91	59.72
CatBoost Master	90.64	97.66	88.37	92.78	98.28
Random Forest UNSW	87.09	97.22	83.42	89.79	97.53
Random Forest CICIoT	60.91	76.39	61.61	68.21	58.43
Random Forest Master	87.63	97.28	84.18	90.26	97.56
Logistic Regression UNSW	72.46	85.33	71.90	78.04	74.88
Logistic Regression CICIoT	49.53	62.06	66.49	64.20	32.22
Logistic Regression Master	72.07	82.30	75.12	78.54	77.17
Decision Tree UNSW	89.95	97.28	87.68	92.23	96.61
Decision Tree CICIoT	76.33	82.70	82.47	82.59	69.42
Decision Tree Master	90.01	97.58	87.49	92.26	97.00
Gradient Boosting UNSW	88.85	97.16	86.14	91.32	97.79
Gradient Boosting CICIoT	72.61	79.56	80.41	79.98	78.68
Gradient Boosting Master	88.02	96.81	85.21	90.64	97.18
SVM UNSW	72.48	85.33	71.92	78.06	74.98
SVM CICIoT	49.56	62.08	66.54	64.23	32.16
SVM Master	72.12	82.27	75.26	78.61	77.14
KNN UNSW	89.91	95.02	89.89	92.38	93.89
KNN CICIoT	57.11	66.08	75.99	70.69	31.34
KNN Master	89.62	94.95	89.51	92.15	93.74

Table 6: All Models on CICIoT test

Model	Acc (%)	Pr (%)	Re (%)	F1 (%)	AUC (%)
CatBoost UNSW	73.98	79.84	90.89	85	14.81
CatBoost CICIoT	98.85	99.96	98.62	99.29	99.75
CatBoost Master	98.85	99.97	98.61	99.28	99.74
Random Forest UNSW	73.10	79.57	89.95	84.44	13.92
Random Forest CICIoT	98.50	99.99	98.15	99.07	99.73
Random Forest Master	98.46	99.99	98.11	99.04	99.71
Logistic Regression UNSW	34.25	65.05	41.03	50.32	19.15
Logistic Regression CICIoT	87.04	94.16	89.59	91.82	96.00
Logistic Regression Master	83.76	93.44	86.03	89.58	91.12
Decision Tree UNSW	44.59	75.22	47.31	58.09	48.72
Decision Tree CICIoT	98.62	99.81	98.48	99.14	99.12
Decision Tree Master	98.71	99.81	98.60	99.2	99.26
Gradient Boosting UNSW	61.12	78.27	72.10	75.06	34.26
Gradient Boosting CICIoT	98.54	99.95	98.25	99.09	99.57
Gradient Boosting Master	98.53	99.95	98.24	99.09	99.55
SVM UNSW	34.25	65.05	41.03	50.32	19.14
SVM CICIoT	90.44	94.21	93.99	94.10	96.03
SVM Master	84.22	93.25	86.84	89.93	90.49
KNN UNSW	46.40	73.15	53.64	61.90	33.90
KNN CICIoT	97.97	99.67	97.82	98.74	98.92
KNN Master	97.91	99.65	97.76	98.70	98.89

Table 7: All Models on Master test

Model	Acc (%)	Pr (%)	Re (%)	F1 (%)	AUC (%)
CatBoost UNSW	83.07	87.70	89.71	88.70	76.29
CatBoost CICIoT	80.31	90.11	82.44	86.11	85.95
CatBoost Master	94.38	98.86	93.49	96.10	99.25
Random Forest UNSW	80.71	87.19	86.68	86.93	79.74
Random Forest CICIoT	78.05	89.35	79.88	84.35	85.43
Random Forest Master	92.57	98.72	91.15	94.78	98.93
Logistic Regression UNSW	55.03	76.65	56.47	65.03	49.88
Logistic Regression CICIoT	66.64	77.16	78.04	77.6	49.94
Logistic Regression Master	77.40	87.89	80.57	84.07	81.45
Decision Tree UNSW	69.26	88.21	67.50	76.48	82.98
Decision Tree CICIoT	86.49	91.21	90.48	90.84	83.29
Decision Tree Master	93.98	98.75	93.05	95.81	98.2
Gradient Boosting UNSW	76.20	87.53	79.12	83.12	84.03
Gradient Boosting CICIoT	84.44	89.61	89.33	89.47	87.30
Gradient Boosting Master	92.81	98.46	91.73	94.98	98.71
SVM UNSW	55.04	76.65	56.48	65.04	50.03
SVM CICIoT	68.21	77.57	80.27	78.9	50.12
SVM Master	77.64	87.81	81.05	84.30	81.30
KNN UNSW	70.06	85.47	71.76	78.02	71.22
KNN CICIoT	75.75	81.55	86.91	84.14	60.71
KNN Master	93.40	97.35	93.63	95.45	96.03

3.1 UNSW-NB15 Test

When trained and evaluated within the UNSW-NB15 dataset, CatBoost achieved the strongest in-domain performance, with 90.70% Accuracy, 92.84% F1-score, and 98.33% AUC.

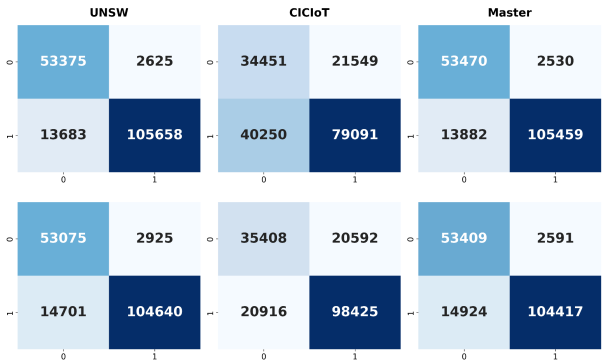

Figure 10: UNSW Confusion Matrices - CatBoost (Top), Decision Tree (Bottom)

When trained exclusively on CICIoT, CatBoost's performance decreased to 64.75% Accuracy, 71.91% F1-score, and 59.72% AUC. Under the same cross-domain configuration, the Decision Tree achieved 76.33% Accuracy, 82.59% F1-score, and 69.42% AUC. Lastly, when trained on the Master dataset, CatBoost achieved 90.64% Accuracy, 92.78% F1-score, and 98.28% AUC.

3.2 CICIoT2023 Test

When trained on UNSW-NB15 and evaluated on CICIoT, CatBoost achieved 73.98% Accuracy and 85.00% F1-score, with an AUC of 14.81%. When trained with the CICIoT dataset, CatBoost achieved 98.85% Accuracy, 99.28% F1-score, and 99.74% AUC. Training on the Master dataset yielded comparable results, maintaining near-perfect classification metrics. When trained exclusively on CICIoT and evaluated on the Master test set, the Decision Tree achieved 86.49% Accuracy and 90.84% F1-score. Standard Gradient Boosting achieved an AUC of 87.30%. Under the same configuration, CatBoost achieved 80.31% Accuracy, 86.11% F1-score, and 85.95% AUC.

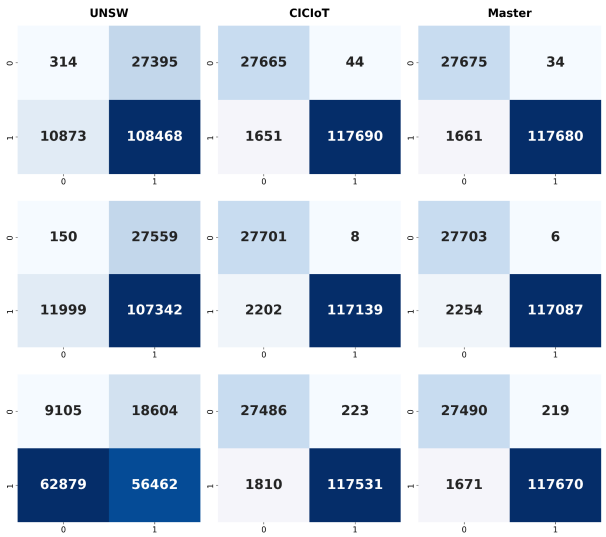

Figure 11: CICIoT Confusion Matrices - CatBoost (Top), Random Forest (Center), Decision Tree (Bottom)

3.3 Master Test

When evaluated on the UNSW dataset, CatBoost achieved 83.07% Accuracy and an F1-score of 88.70%, with an AUC of 76.29%. In comparison, the Random Forest baseline on the same dataset reached an Accuracy of 80.71% and an F1-score of 86.93%. Gradient Boosting showed varying results across its iterations for UNSW, with its most balanced run yielding an AUC of 84.03%.

In the CICIoT evaluation context, CatBoost maintained a strong profile with 80.31% Accuracy, a 90.11% Precision rate, and an 86.11% F1-score. This outperformed the Random Forest configuration, which recorded 78.05% Accuracy and an 84.35% F1-score. Interestingly, Gradient Boosting achieved its highest specific AUC of 87.30% within this category, though its Accuracy remained relatively lower at 84.44%.

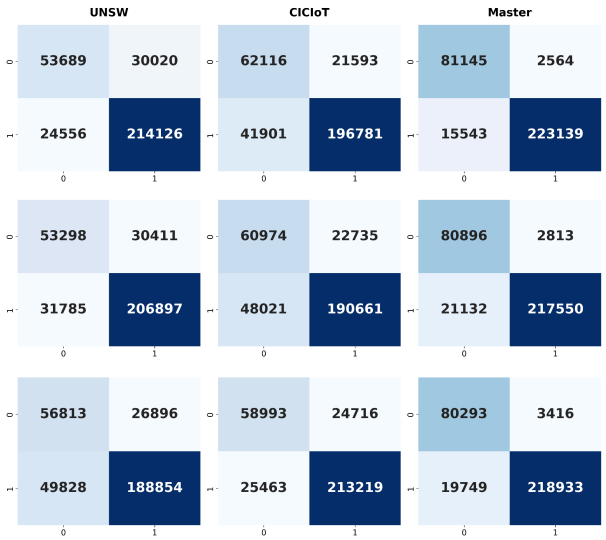

Figure 12: CICIoT Confusion Matrix CatBoost (Top), Random Forest (Center), Gradient Boosting (Bottom)

When trained on the Master dataset, CatBoost achieved a near-perfect 94.38% Accuracy, 96.10% F1-score, and a standout AUC of 99.25%. While the baselines were competitive with Random Forest hitting 96.10% F1 and Gradient Boosting reaching 94.98%. CatBoost consistently led in Precision and overall discriminative power.

4. DISCUSSION

Across all evaluation scenarios, consistent patterns emerged regarding model behavior under domain shift. First, CatBoost achieved the highest in-domain and multi-domain performance across all datasets. However, it also exhibited significant degradation when trained on a single dataset and evaluated on a different one. This was particularly evident in the UNSW-to-CICIoT configuration, where AUC dropped to 14.81%, indicating severe ranking instability.

In contrast, simpler tree-based models, particularly the Decision Tree, demonstrated comparatively stronger resilience under cross-domain evaluation. Although their peak in-domain performance was lower, their degradation under distribution shift was less extreme.

Highly optimized ensemble methods appear to leverage fine-grained feature interactions that may not transfer across heterogeneous network environments. Simpler models, while less powerful under ideal conditions, may rely on more stable decision boundaries.

The Master dataset experiments further demonstrate that exposure to heterogeneous training distributions significantly improves CatBoost's robustness. When trained on the combined dataset, its performance was restored across all test scenarios, albeit slightly lower, suggesting that multi-domain training mitigates domain sensitivity.

5. REFERENCES

[1] A. Fathima, A. Khan, M. F. Uddin, M. M. Waris, S. Ahmad, C. Sanin, and E. Szczerbicki, "Performance Evaluation and Comparative Analysis of Machine Learning Models on the UNSW-NB15 Dataset: A Contemporary Approach to Cyber Threat Detection," Cybernetics and Systems, vol. 56, no. 8, pp. 1160–1176, 2025, doi: 10.1080/01969722.2023.2296246. **[Journal Article]**

[2] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. Military Commun. Inf. Syst. Conf. (MilCIS), 2015, pp. 1-6, doi: 10.1109/MilCIS.2015.7348942. **[Conference Paper]**

[3] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset," Inf. Secur. J. Glob. Perspect., vol. 25, no. 1-3, pp. 18–31, 2016, doi: 10.1080/19393555.2015.1125974. **[Journal Article]**

[4] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," IEEE Trans. Ind. Informat., vol. 14, no. 11, pp. 4819–4830, Nov. 2018, doi: 10.1109/TII.2017.2771459. **[Journal Article]**

[5] N. Moustafa, J. Hu, and J. Slay, "Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models," in Data Analytics and Decision Support for Cybersecurity, I. Palomares, H. K. Kalutarage, and C. P. Huang, Eds. Cham, Switzerland: Springer, 2017, pp. 127–156. **[Book chapter]**

[6] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in Big Data Technologies and Applications (BDTA 2020), vol. 371, pp. 117–135, 2021, doi: 10.1007/978-3-030-72802-1_9. **[Conference Paper]**

[7] I. H. Sarker, A. S. M. Kayes, and S. Badsha, "A systematic literature review of cyber-security data repositories and performance assessment metrics for semi-supervised learning," IEEE Access, vol. 11, pp. 37706–37733, 2023, doi: 10.1109/ACCESS.2023.3266283. **[Journal Article]**

[8] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, pp. 1–37, 2014, doi: 10.1145/2523813. **[Journal Article]**

[9] HIMADRI07, CICIoT2023 Dataset, Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/himadri07/ciciot2023/code **[Dataset]**

[10] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in Advances in Neural Information Processing Systems, vol. 31, 2018, pp. 6638–6648. **[Journal Article]**

[11] E. Poslavskaya and A. Korolev, "Encoding categorical data: Is there yet anything 'hotter' than one-hot encoding?," arXiv preprint arXiv:2312.16930, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2312.16930**[Preprint]**

[12] Yan, L., Zhou, T., and Chen, X. 2025. Unknown category malicious traffic detection based on contrastive learning. In Machine Learning for Cyber Security: 6th International Conference, ML4CS 2024, D. Chen et al. (Eds.). Springer Nature Singapore, Singapore, 231–245. doi: 10.1007/978-981-96-4566-4

[13] Gulzar, Q. and Mustafa, K. 2025. Enhancing network security in industrial IoT environments: a DeepCLG hybrid learning model for cyberattack detection. *International Journal of Machine Learning and Cybernetics*. doi: 10.1007/s13042-025-02544-w **[Journal Article]**

[14] Hajjouz, A. and Avksentieva, E. 2024. Optimizing Intrusion Detection for DoS, DDoS, and Mirai Attacks Subtypes Using Hierarchical Feature Selection and CatBoost on the CICIoT2023 Dataset. Data and Metadata. 3 (2024), 577. doi: 10.56294/dm2024.577 **[Journal Article]**

[15] Al-Riyami, S., Lisitsa, A., and Coenen, F. 2021. Cross-datasets evaluation of machine learning models for intrusion detection systems. In *Proceedings of the 6th International Congress on Information and Communication Technology (ICICT 2021)*, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi (Eds.). *Lecture Notes in Networks and Systems*, Vol. 217. Springer, Singapore, 815–828. doi: 10.1007/978-981-16-2102-4_73.

[16] S. P. Ang, S. T. M. Duong, S. L. Phung, and A. Bouzerdoum, "DAP: A dataset-agnostic predictor of neural network performance," Neurocomputing, vol. 583, pp. 127566, May 2024, doi: 10.1016/j.neucom.2024.127566. **[Journal Article]**

[17] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," arXiv preprint arXiv:1810.11363, 2018. [Online]. Available: https://arxiv.org/abs/1810.11363 **[Preprint]**

[18] Boldini, Davide, Francesca Grisoni, Daniel Kuhn, Lukas Friedrich, and Stephan Sieber. 2023. *Practical guidelines for the use of gradient boosting for molecular property prediction.* Journal of Cheminformatics 15 (2023). doi: 10.1186/s13321-023-00743-7

[19] I. D. Dinov, *Data Science and Predictive Analytics: Biomedical and Health Applications using R*, 2nd ed. Cham, Switzerland: Springer Nature, 2023. ISBN: 3031174836. [Book]

[20] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, p. 6086, 2024, doi: 10.1038/s41598-024-56706-x. [Journal Article]

[21] K. Sujon, R. Hassan, K. Choi, and M. A. Samad, "Accuracy, precision, recall, F1-score, or MCC? Empirical evidence from advanced statistics, ML, and XAI for evaluating business predictive models," *J. Big Data*, vol. 12, 2025, doi: 10.1186/s40537-025-01313-4. [Journal Article]

[22] J. Li, "Area under the ROC Curve has the most consistent evaluation for binary classification," *PLOS ONE*, vol. 19, no. 12, p. e0316019, 2024, doi: 10.1371/journal.pone.0316019. [Journal Article]

[23] Confusion Matrix-Based Performance Evaluation Metrics", *AJBR*, vol. 27, no. 4S, pp. 4023–4031, Nov. 2024, doi: 10.53555/AJBR.v27i4S.4345. [Journal Article]