2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC) | 979-8-3315-2389-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISACC65211.2025.10969343

Ensemble-Based Lightweight Machine Learning Optimization for IoT Network Intrusion Detection

Milan Samantaray
Department of Computer Science and Engineering, C.V. Raman Global University,
Bhubaneswar, Odisha, India
ORCID 0000-0002-2870-5227

Ram Chandra Barik
Department of Computer Science and Engineering, C.V. Raman Global University,
Bhubaneswar, Odisha, India
ORCID 0000-0002-2803-5868

Anil Kumar Biswal
Department of Computer Science, Udayanath Autonomous College of Science and Technology, Cuttack,
Odisha, India
ORCID 0000-0001-7341-216X

***Abstract*- The rapid development of the Internet of Things (IoT) presents various security concerns, necessitating the implementation of robust and effective Intrusion Detection Systems (IDS). With the growth of IoT networks, cyber risks have increased, putting data integrity and stability at risk. As consequently, the growth of IoT networks has coincided with an increase in cyber threats, threatening the security and integrity of data. Various IDSs have been suggested to identify malicious activity based on previously established attack patterns; this should help to protect data from abuse and odd attempts. Existing intrusion detection systems need upgrading due to the fast rise in these types of threats. This study proposes an Ensemble-Based Lightweight Machine Learning (ELML) specifically optimized for IoT network IDS, leveraging the UNR-IDD dataset. The UNR-IDD dataset is a recent and extensive benchmark that includes several contemporary network attack scenarios, facilitating effective training and assessment of detection algorithms. The proposed approach utilizes feature engineering and lightweight machine learning models including K-Nearest Neighbors (KNN), Random Forest (RF), Logistic Regression (LR), AdaBoost (AB), and Decision Tree (DT). In addition, the Ensemble-Based Lightweight Machine Learning (ELML) IDS model is being optimized for performance using a hyperparameter optimization technique that is based on the Crow Search Algorithm (CSA). The results show significant accuracy improvements over existing algorithms in multi-class classification contexts.**

Keywords- Internet of Things (IoT), Intrusion Detection, Lightweight Machine Learning, Cyber Security, Anomaly Detection.

I. INTRODUCTION

The Internet of Things (IoT) has only recently been recognized as an innovative computing technology due to its capacity to influence all aspects of human existence. An industry evaluation projects that between 2020 and 2025, there will be approximately 50 billion to 75 billion internet connections in smart devices, positioning it as one of the most rapidly growing sectors in computer history [1]. The IoT enables the effortless transfer of data between machines, hence obviating the necessity for human intervention through the facilitation of machine-to-machine communication. The IoT can be applied in several contexts, including power grids, home automation, transportation systems, monitoring devices, agriculture, and intelligent environments [2].

The IoT can be applied in several contexts, including power grids, home automation, transportation systems, monitoring devices, agriculture, and intelligent environments. Security and privacy concerns have consistently been viewed as a significant challenge in IoT adoption, attributed to the swift advancements in information and communication technology and global vulnerability issues [3]. The large-scale deployment of the Internet of Things in open environments has rendered networks susceptible to various security threats and intrusions. Physical attacks, encryption attacks, distributed denial-of-service (DDoS), man-in-the-middle (MITM) attacks, denial of service (DoS), firmware hijacking, botnets, ransomware, eavesdropping, brute force password attacks, and various other cyberattacks continue to pose significant threats. Consequently, it is imperative to develop an encryption method that is both efficient and capable of rapidly and consistently adapting to identify threats such as DoS assaults on IoT networks. Various Intrusion Detection Systems (IDS), including signature-based and anomaly-based systems, are available to prevent assaults and intrusions in IoT networks. The desire to steal data, make illegal money, and find new victims motivates cybercriminals globally [4].

Intrusion detection systems (IDS) are designed to monitor network traffic and detect malicious activities or intrusions. First proposed in 1980, IDS focus on identifying threats that traditional firewalls might miss, such as malicious network traffic and suspicious online activities. They are essential for safeguarding the availability, integrity, and confidentiality of computer systems. One type of IDS is Host Intrusion Detection Systems (HIDS), which operate on individual hosts to detect and track potential threats [5]. Intrusion detection systems (IDS) can monitor various segments, with Host Intrusion Detection Systems (HIDS) focusing on individual host behaviors without relying on network traffic, making them effective against insider attacks. Examples like Tripwire and AIDE may miss network-based threats. Network Intrusion Detection Systems (NIDS), on the other hand, are cost-effective, easy to install, and rely on established security mechanisms but can fail to detect new exploits. Combining the strengths of NIDS and HIDS provides greater versatility, and this approach is known as a "Hybrid IDS" [6]. Effective management is crucial for maintaining HIDS functionality. The primary motivation to upgrade IDS with modern technologies is the emergence of new and undiscovered threat types. One important tool for dealing with such issues is machine learning.

By integrating nature-based methods like harmony search algorithm (HSA), simulated annealing (SA), particle swarm optimization (PSO), Artificial Bee Colony algorithm (ABC), Spider Monkey Optimization (SMO), etc., into the intrusion detection strategy, it is possible to attain high accuracy [7]. Reason being, these algorithms can enhance prediction accuracy while decreasing execution time. Therefore, the suggested method, CSA-ML, uses the Crow Search Algorithm (CSA) to optimize the ML parameters. The UNR-IDD dataset is used to test the efficacy of six different lightweight ML methods for multi-class classification: K-Nearest Neighbors (KNN), Random Forest (RF), Logistic

Regression (LR), AdaBoost (AB), and Decision Tree (DT). If an impending data packet is an attack, the multi-class classifications will specify which kind of attack it is, and if it is a normal entry, it will discriminate between the two. Low computing power and resources are needed by the suggested model, which significantly improves accuracy.

The work presented in this study includes the following significant contributions:

- Finding a specific set of features that would improve the detecting mechanism's effectiveness is the purpose of this paper, which aims to use the ensemble-based lightweight ML (ELML) technique.
- Maximize the rate of intrusion detection by optimizing several ELML classifiers using the Crow Search Algorithm.
- Test the proposed approach using the UNR-IDD dataset and assess its performance. Its efficacy in comparison to state-of-the-art approaches is determined using the experimental results.

The rest of the article is structured like this: Related work is covered in Section II. The proposed model is fully detailed and illustrated in Section III. Section IV provides a detailed explanation of the methodologies. The results analysis is covered in Section V. Section VI serves as the paper's conclusion.

II. Related work

The safety of Internet of Things devices is in risk because cyberattacks are becoming more common. Modern research proposes a number of ways to stop these assaults from happening, many of which include using machine learning to find and identify them. This follows are a discussion of some of the efforts made in this area.

A rule-based classification system for cloud networks was proposed in [8] as a means to identify DoS assaults. Feature selection was accomplished with the assistance of ranking and scoring algorithms. Then, a classifier based on expert knowledge—a rule-based classification algorithm—was put into place. Five datasets including five thousand assault cases each were created for the purpose of performance evaluation. We compared the performance of four pre-existing algorithms: Naive Bayes, Multilayer perceptron, Support Vector Machine, and Decision Tree. In order to create the hybrid IDS, it was recommended by [9] to combine different machine learning techniques. First, the transformation and normalization were carried out. The next step was to apply several machine learning techniques to the classification process. The accuracy, false positive rate, true positive rate, F-measure, and Matthews Correlation Coefficient (MCC) were the metrics used to assess the performance.

In [10], a model for intrusion detection was developed utilizing an ensemble of ML classification algorithms, including DT, J48, and SVM. The nine most relevant and critical features in the KDD99 intrusion detection dataset were selected using particle swarm optimization. With a low FAR of 0.9%, the results obtained by the proposed model showed a higher accuracy of 90%.

In [11], the authors compared the efficacy of various classical ML systems for attack traffic detection on various ID-based datasets. Next, three machine learning methods, namely SVM, KNN, and DT, were applied to the normalized datasets (CICIDS2018, UNSW-NB15, ISCX2012, NSLKDD, and CIDDS001). DT's detecting accuracy rate of 99–100% across all datasets puts it ahead of other classifiers. With a 75% false-positive rate (FPR), they discovered that SVM was misclassifying. Both the amount of features and the quantity of training samples were used to determine the algorithm's complexity.

In [12] proposed a new encryption approach to anticipate and combat cyberattacks on cyber-physical systems. The method optimizes the LightGBM algorithm's hyperparameters using Bayesian techniques and was trained on the UNR-IDD intrusion detection dataset. Tested in Reno, the approach achieved high performance with 0.9918 accuracy, 0.9922 precision, and 0.9922 recall. The method strengthens security by improving accuracy and AUC values, offering robust protection for user information.

III. PROPOSED MODEL

The machine learning-driven IDS's ability to be lightweight and compatible with the limited nodes processing capabilities is one of our primary objectives. As a result, each node of an IoT cannot have an active intrusion detection agent owing to power consumption and computational limitations, as stated in [13]. Consequently, we have chosen a centralized intrusion detection system (IDS) design to address both the restricted capacity and peripheral heterogeneity issues. This design places the IDS on the network layer of the Internet of Things (IoT), above the Gateway component.

In order to keep the network safe, the IDS is essential. As a whole, intrusion detection systems gather data that comes in via the network. After receiving this data, a pre-processing method is used to remove any noise and replace any superfluous or misinterpreted properties. The pre-processed data is evaluated and sorted based on its seriousness. No further tweaks are required if the record is normal; otherwise, it is sent to reporting generation, which raises an alarm. We use the issue's intricacy to set up notifications. The IDS system provides reports on detection rates for false negatives, false positives, true negatives, and true positives. When an intrusion detection system (IDS) reports a real danger, this is known as a true positive outcome. A true negative happens when an intrusion detection system (IDS) does not trigger an alarm even if no attack has taken place.

An important systemic defect occurs when an intrusion detection system (IDS) identifies no attack, a phenomenon called a false positive or false alert. In intrusion detection systems, a false negative occurs when an actual or potential attack goes unnoticed. If these technologies fail to produce the desired outcomes, it can result in low detection rates or false-negative rates. The goal of our research is to develop a IoT infrastructure attack detection system that can effectively address this issue in Fig. 1 that shows the proposed framework's Ensemble-Based Lightweight ML optimized IDS for IoT network. There are three processes that need to be followed: pre-processing, feature selection, and classifier. First, there is pre-processing; second, there is XGBoost for feature selection; and third, there is lightweight ML classifiers for intrusion detection in our proposed system. Before proceeding with processing, the features are pre-processed. The XGBoost method for selecting features is the main subject of this paper. Xtreme gradient boosting (XGBoost) is an enhancement approach that is included in the ensemble-

based method. By utilizing parallel processing to eliminate missing data and prevent overfitting, the XGBoost algorithm has become well-known as an effective means of optimizing the gradient boosting algorithm. The scalability and performance of XGBoost are contributing to its rising popularity.

A dataset's numerical columns can have their values normalized to a common scale after preprocessing so that the ranges of possible values are not distorted. We divide the selection process into two parts, for example, training and testing. A multi-class lightweight ML classifiers are then used to pick features for classification. Furthermore, the Crow Search Algorithm optimizes the parameters of the classifier.

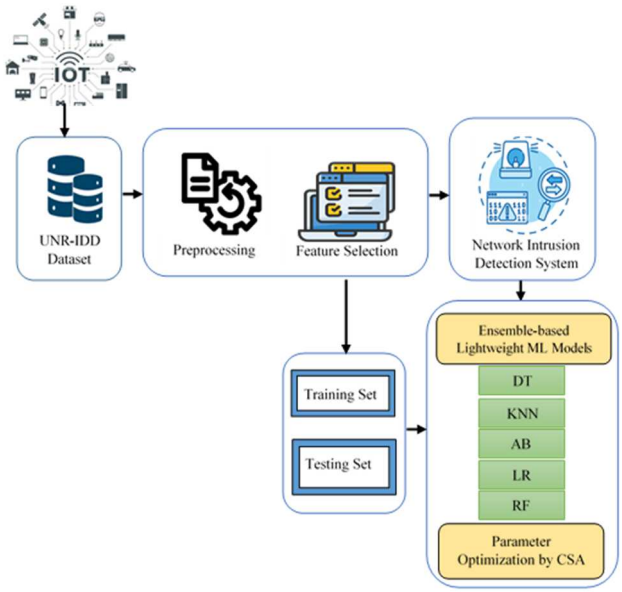


Fig. 1: The Proposed Ensemble-based Lightweight ML-Driven IDS for IoT Network

Table I: UNR-IDD Dataset Description

Label	Type	Description
0	Normal	Normal Network Functionality
1	TCP-SYN	TCP-SYN Flood
2	PortScan	Port Scanning
3	Overflow	Flow Table Overflow
4	Blackhole	Blackhole Attack
5	Diversion	Traffic Diversion Attack

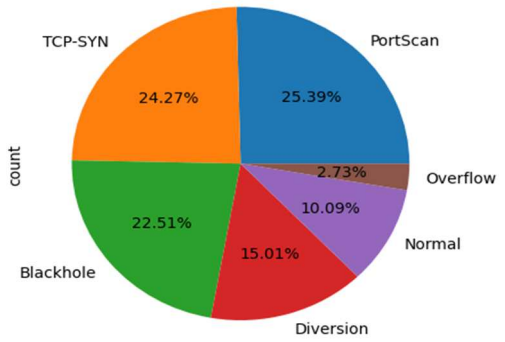


Fig. 2: Label count (%) of UNR-IDD Dataset

A. Dataset

In this case, we make use of the datasets maintained by the University of Neveda-Reno Intrusion Detection (UNR-IDD). There are three possible types of feature values in a dataset: symbolic, discrete, and continuous. There are 29 features in the UNR-IDD dataset and the label counting percentage is shown in Fig. 2. Normal operations and various forms of cyberattacks, such as TCP-SYN Flood, Port Scan, Flow Table Overflow, Blackhole and Traffic Diversion, are recorded in this data collection of network traffic that represents in Table I. Including protocol kinds, IP addresses, port numbers, and time-based elements, all of the things have thorough branding.

IV. METHODOLOGY

The literature suggests that deep learning and artificial neural networks (ANN) need a lot of processing resources to run with many hidden layers. On above that, DL is complex, therefore increasing the number of hidden layers may provide optimal or better results. Because of this, the systems are prone to faults, and those errors can have a significant impact. The over-training caused an increase in the number of hidden layers relative to the difficulty of the task. Time and complexity are affected by this condition. Not only that, it becomes useless when used to testing datasets and has a devastating impact on available resources.

Anomaly-based intrusion detection systems (AIDS) have a low false alarm rate but may produce high false-positive rates for zero-day attacks. To improve detection, machine learning algorithms like fuzzy logic, support vector machines, neural networks, and Markov models are used. Network-based intrusion detection systems (NIDS) can overcome some of AIDS's limitations, and their performance is enhanced by regularly updating machine learning models.

The dataset used in this study is UNR-IDD, and we are attempting to solve a classification challenge. For the purpose of intrusion detection, this study selects five distinct ensemble-based lightweight ML classification methods. Stacking classifier ensembles use K-Nearest Neighbors (KNN), Decision tree (DT), AdaBoost (AB), Random Forest (RF), and Logistic Regression (LR). These six Lightweight ML algorithms have been chosen based on their optimal performance in the literature, as explained below.

A. K-Nearest Neighbors (KNN)

The k-Nearest-Neighbors ML method can do supervised and unsupervised jobs. Many modern clustering approaches are based on KNN. We used KNN supervised ML in this work. This approach is based on the Euclidean (EM) metric. A three-dimensional EM estimates the distance between two instances. The EM between p and q in Euclidean space(Z) is determined as follows.

$$\Delta(p,q)=\sqrt{\sum_{i=1}^{r}(p_k - q_k)^2} \quad (1)$$

Equation 1 shows the value of r is the maximum possible occurrence of Z. Using the class of its k related neighbour instances as a basis, the k-NN approach determines the identity (label) of an instance r0 in Z by calculating the EM separating r0 from its k-Nearest-Neighbors examples in Z.

B. Decision tree (DT)

The decision tree is a supervised machine learning technique that helps with both regression and classification. The model that resulted from the discussion was shaped like a tree. Users will have an easier time understanding DT as a result. Furthermore, the produced trees can be viewed graphically in a number of ML applications. So this is denoted by;

$$G = 1 - \sum_{j=1}^{c}(pj)^2 \quad (2)$$

where G represents as gini impurity, pj is the proportion of samples belonging to class j, and c is the number of classes as seen in (2).

C. AdaBoost (AB)

AdaBoost assigns weights to training samples and iteratively trains weak learners. After each iteration, the algorithm adjusts the weights of the samples based on whether they were correctly or incorrectly classified. Misclassified samples are given higher weights, emphasizing their importance in the next iteration. The final prediction is made by combining the outputs of all weak learners, with each learner's contribution weighted by its accuracy. It is determined by;

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \quad (3)$$

Here H(x): Final prediction (strong learner), sign: Sign function that outputs +1 for positive values and −1 for negative values, T: Total number of weak learners, α_t: Weight of the t^{th} weak learner, $h_t(x)$: Output of the t^{th} weak learner for input x (+1 or -1) as seen in (3).

D. Random Forest (RF)

Classification and regression are two applications of the supervised learning method known as random forest in machine learning. This strategy improves prediction accuracy by combining the outputs of multiple classifiers, specifically using decision trees to analyze different aspects of the input data and averaging their results. Random Forest (RF) is often used in machine learning due to its speed and ability to handle missing or faulty data. However, when predicting a dataset's class, some decision trees may yield accurate results while others may not. However, when considered collectively, the trees only produce accurate predictions. It is mathematically explained by;

$$RF = \frac{1}{n} \sum_{j=1}^{n} (fj - yj)^2 \quad (4)$$

where n denotes number of data points, *fj* is the value returned by model and *yj* is the actual value for data point *j* as shown (4).

E. Logistic Regression (LR)

Despite being referred to as a "regression" technique, the binary classification problem is where logistic regression (LR) is most frequently applied. The LR can also be useful for one-versus-rest learning techniques used in multiclass classification situations. In the LR model, a linear ML model is subjected to the sigmoid function or variations thereof. A narrower interval between 0 and 1 is the end product of this operation. As the result approaches 1, the likelihood of a specific class's existence increases. Logistic Regression is denoted in the mathematical formula:

$$\sigma(y) = \frac{1}{1+e^{-y}} \quad (5)$$

here the above expression 'σ' operates on y as seen in (5).

F. Crow Search Algorithm(CSA)

Crows have the largest brains relative to their body size, earning them the reputation of being the most intelligent birds. A crow's brain is proportionally smaller than its body. The dazzling antics of crows are well-documented. When one of them detects the approach of a stranger, they can communicate by recognizing each other's faces. On top of that, they have strong memory abilities and may recall the exact location of their food cache for months. They covertly track different birds to learn where they store their food. As soon as the birds go, they gobble up all the food. If a crow steals food, it may change hiding places to avoid detection. Due to their intellect, they can predict when other thieves will be active and plan to prevent food theft.

Crowd Search Algorithm is based on these principles:

- Colonies are their home.
- They have the remarkable ability to remember exactly where they hide.
- To steal, they swam up to each other.
- They use probability to protect their hideouts from theft.

Contemplate an N-dimensional environment populated by multiple crows. The variable C represents the size of the group, while the position of the crow, denoted as i, at a given moment in the iteration within the "*iter*" environment is articulated as:

$$N^{x,iter+1} = (p = 1,2,\dots,C; iter = 1,2,\dots,iter_{max}) \quad (6)$$

where $N^{x,iter}$, $iter_{\max}$ considered as maximum iteration. Every crow remembered exactly where their nests were. Iteration defines the position of the crow's hidden place as (Su) as shown in (6). When it comes to Crow (u), this is the pinnacle of success. A crow's memory will keep track of the improved place it reached. Up until then, it will try to improve its search space position. Assume that crow(j) needs to get to its hidden position during iteration t. Meanwhile, crow (u) follows crow (v) to its hidden place.

IV. RESULT ANALYSIS

The proposed experiment has been run on a Windows 11 system with a Core i5 CPU and 16 GB of RAM, using Python 3.9. According to the ensemble-based lightweight ML optimization techniques, Tables II to VI display the results of the multiclass classification methodology that utilized the use of the entire UNR-IDD feature space.

For the UNR-IDD datasets, the confusion matrices of various ensemble-based lightweight ML intrusion detection system optimization are displayed in Fig. 3.

Table II: Classification Report of KNN-CSA Model for UNR-IDD Dataset

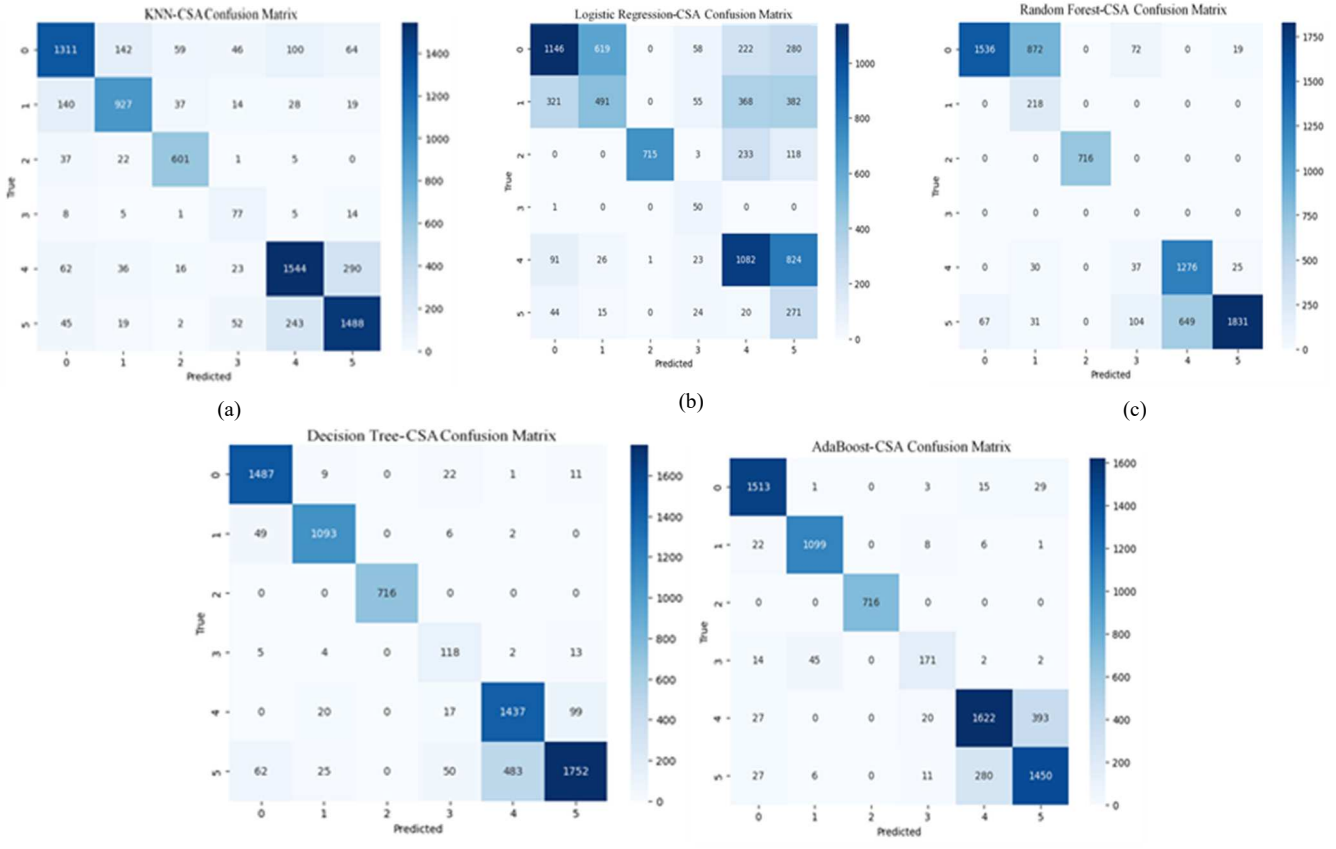
Label	Precision	Recall	F1-score	support
0	0.76	0.82	0.79	1603
1	0.80	0.81	0.80	1151
2	0.90	0.84	0.87	716
3	0.70	0.36	0.48	213
4	0.78	0.80	0.79	1925
5	0.80	0.79	0.80	1875
accuracy			0.79	7483
Macro avg	0.79	0.74	0.75	7483
Weighted avg	0.79	0.79	0.79	7483

Table III: Classification Report of LR-CSA Model for UNR-IDD Dataset

Label	Precision	Recall	F1-score	support
0	0.49	0.71	0.58	1603
1	0.30	0.43	0.35	1151
2	0.67	1.00	0.80	716
3	0.98	0.23	0.38	213
4	0.53	0.56	0.54	1925
5	0.72	0.14	0.24	1875
accuracy			0.50	7483
Macro avg	0.62	0.51	0.48	7483
Weighted avg	0.56	0.50	0.47	7483

Table IV: Classification Report of RF-CSA Model for UNR-IDD Dataset

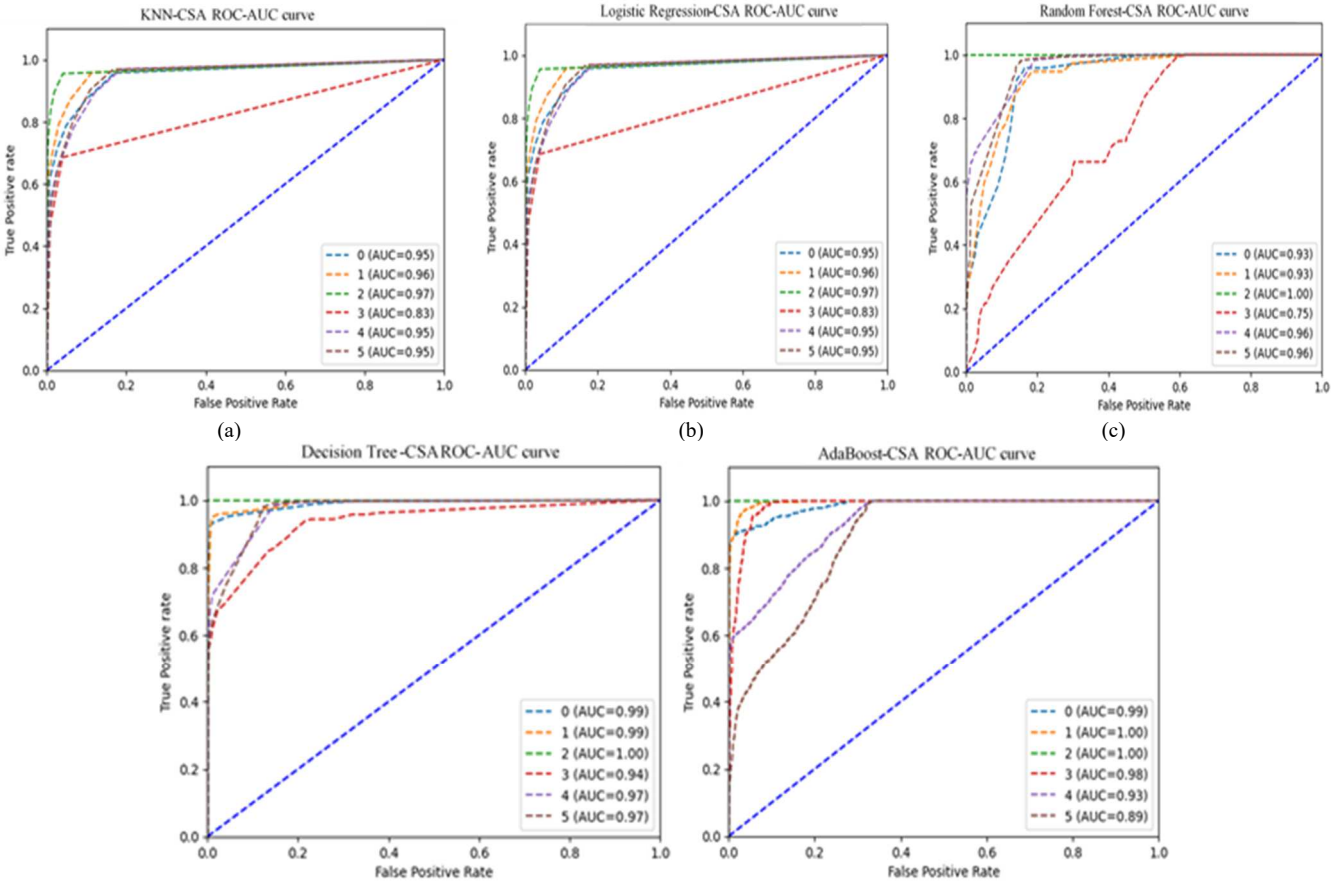
Label	Precision	Recall	F1-score	support
0	0.61	0.96	0.75	1603
1	1.00	0.19	0.32	1151
2	1.00	1.00	1.00	716
3	0.00	0.00	0.00	213
4	0.93	0.66	0.77	1925
5	0.68	0.98	0.80	1875
accuracy			0.75	7483
Macro avg	0.71	0.63	0.61	7483
Weighted avg	0.79	0.75	0.71	7483



(a) (b) (c)

(d) (e)

Fig. 3: Confusion Matrix of Lightweight ML models using CSA for UNR-IDD Dataset



(a) (b) (c)

(d) (e)

Fig. 4: ROC and AUC Curve of Lightweight ML models using CSA for UNR-IDD Dataset

Table V: Classification Report of DT-CSA Model for UNR-IDD Dataset

Label	Precision	Recall	F1-score	support
0	0.97	0.93	0.85	1603
1	0.95	0.95	095	1151
2	1.00	1.00	1.00	716
3	0.83	0.55	0.66	213
4	0.91	0.75	0.82	1925
5	0.74	0.93	0.83	1875
accuracy			0.87	7483
Macro avg	0.90	0.85	0.87	7483
Weighted avg	0.89	0.88	0.88	7483

Table VI: Classification Report of AB-CSA Model for UNR-IDD Dataset

Label	Precision	Recall	F1-score	support
0	0.97	0.94	0.96	1603
1	0.97	0.95	0.96	1151
2	1.00	1.00	1.00	716
3	0.73	0.80	0.77	213
4	0.79	0.84	0.81	1925
5	0.82	0.77	0.79	1875
accuracy			0.88	7483
Macro avg	0.88	0.89	0.88	7483
Weighted avg	0.82	0.77	0.88	7483

Table VII: Result Analysis of Ensemble-based Lightweight ML using CSA for UNR-IDD Dataset

Ensemble Models	Time cost (Sec.)	Testing Acc.
KNN-CSA	0.04053	0.79
LR-CSA	4.3695	0.50
RF-CSA	3.9485	0.75
DT-CSA	16.1502	0.87
AB-CSA	60.7099	0.88

Table VIII: Evaluation of the proposed model in relation to existing methods using UNR-IDD Dataset

References	Preprocessing	Classification	Accuracy
[21]	Feature Selection, Normalization	DNN	77%
[22]	Feature Scaling, Normalization	RF	73%
[23]	Data normalization and Data transformation	AdaBoost	82%
[24]	Feature Scaling, Resampling	RF	75%
Proposed Approach	Feature Transformation, Feature Selection and Data Normalization	KNN-CSA, LR-CSA, RF-CSA, DT-CSA, AB-CSA	88%

Average receiver performance as determined by the area under the receiver's curve (AUCROC). At various thresholds, this graph indicates how effective a categorization task is. AUC is a measure for evaluating the degree to which two groups may be differentiated, while ROC is a probability curve. This measure shows how well the model can distinguish between various kinds of attacks. The area under the curve (AUC) measures how well a model can distinguish between the 0 and 1 classes. A model's ability to distinguish between an attack and a non-attack scenario is evaluated by the area under the curve (AUC). On the ROC curve, TPR is shown on the y-axis and FPR is shown on the x-axis; these two variables are used to compare TPR and FPR Fig. 4 displays the results on the UNR-IDD dataset in terms of ROC-AUC curves.

The five ensemble-based lightweight ML algorithms are optimized by Crow Search Algorithm such as KNN-CSA, LR-CSA, RF-CSA, DT-CSA, and AB-CSA were applied to all features after the UNR-IDD dataset was pretested. Among these algorithms, AB-CSA achieved the best average accuracy of 0.88 in multi-class classification situations that is shown in Table VII. We compared our results with those of other well-established approaches used for data preparation in Table VIII.

IV. CONCLUSION

This paper introduces a model for intrusion detection that utilizes ensemble-based lightweight ML(ELML) intrusion detection system optimization techniques for IoT network. The proposed model ensures the detection of all types of attacks. The approach offers notable precision while requiring minimal computational power and resources, along with a reduced false alarm rate, by employing ensemble-based lightweight machine learning algorithms rather than artificial neural networks and deep learning techniques within an ensemble framework. The proposed model incorporates five Lightweight ML algorithms like KNN, AB, LR, RF, and DT utilizing a Crow Search Algorithm (CSA), and it has been evaluated on the UNR-IDD dataset in multi-class scenarios. The findings show that AB outperforms all other methods in terms of accuracy, precision, recall, and F-Score.

In the future, the proposed ensemble model will be expanded to include deep and recurrent neural networks with the aim of enhancing accuracy in detecting intrusions in IoT network.

REFERENCES

[1] Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S., & Mustaqim, M. (2020). Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. Ieee Access, 8, 23022-23040.

[2] Huda, N. U., Ahmed, I., Adnan, M., Ali, M., & Naeem, F. (2024). Experts and intelligent systems for smart homes' Transformation to Sustainable Smart Cities: A comprehensive review. Expert Systems with Applications, 238, 122380.

[3] Alguliyev, R., Nabiyev, B. R., & Dashdamirova, K. (2023, October). Cyber Threats and Their Intellectual Analysis Issues in the Context of Technological Challenges of the IV Industrial Revolution. In 2023 IEEE 17th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-6). IEEE.

[4] Omolara, A. E., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A., & Arshad, H. (2022). The internet of things security: A survey encompassing unexplored areas and new insights. Computers & Security, 112, 102494.

[5] Samantaray, M., Barik, R. C., & Biswal, A. K. (2024). A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems. Decision Analytics Journal, 11, 100478.

[6] Samantaray, M., Satapathy, S., & Lenka, A. (2022). A systematic study on network attacks and intrusion detection system. In Machine Intelligence and Data Science Applications: Proceedings of MIDAS 2021 (pp. 195-210). Singapore: Springer Nature Singapore.

[7] Hassan, I. H., Mohammed, A., & Masama, M. A. (2023). Metaheuristic algorithms in network intrusion detection. Comprehensive Metaheuristics, 95-129.

[8] Rajendran, R., Santhosh Kumar, S. V. N., Palanichamy, Y., & Arputharaj, K. (2019). Detection of DoS attacks in cloud networks using intelligent rule based classification system. Cluster Computing, 22, 423-434.

[9] Çavuşoğlu, Ü. (2019). A new hybrid approach for intrusion detection using machine learning methods. Applied Intelligence, 49, 2735-2761.

[10] Kumari, A., & Mehta, A. K. (2020, October). A hybrid intrusion detection system based on decision tree and support vector machine. In 2020 IEEE 5th International conference on computing communication and automation (ICCCA) (pp. 396-400). IEEE.

[11] Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188, 107840.

[12] Dalal, S., Poongodi, M., Lilhore, U. K., Dahan, F., Vaiyapuri, T., Keshta, I., ... & Simaiya, S. (2023). Optimized LightGBM model for security and privacy issues in cyber-physical systems. Transactions on Emerging Telecommunications Technologies, 34(6), e4771.

[13] Dalal, S., Lilhore, U. K., Faujdar, N., Simaiya, S., Ayadi, M., Almujally, N. A., & Ksibi, A. (2023). Next-generation cyber attack prediction for IoT systems: leveraging multi-class SVM and optimized CHAID decision tree. Journal of Cloud Computing, 12(1), 137.