International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 3, June 2025, pp. 3095~3106
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i3.pp3095-3106

Exploring the effectiveness of multiclass decision jungle for internet of things security

Smitha Rajagopal[1], Abhik Sarkar[2], Venkat Narayanan Manjunath[1]
[1]Alliance School of Advanced Computing, Alliance University, Bengaluru, India
[2]Alliance School of Liberal Arts, Alliance University, Bengaluru, India

Article Info

Article history:

Received Aug 25, 2024
Revised Feb 4, 2025
Accepted Mar 3, 2025

Keywords:

CICIOT 2023
Decision jungle
Internet of things security
Network intrusion detection
Permutation feature importance

ABSTRACT

Network intrusion detection systems (NIDS) are vital in protecting computer networks against cyber security incidents. The relationship between NIDS and internet of things (IoT) security is pivotal and NIDS plays a significant role in ensuring the security and reliability of IoT ecosystems. Ensuring the security of IoT devices is critical for several reasons. It helps safeguard sensitive information, guarantees the dependability of crucial infrastructure, meets regulatory obligations, and fosters user confidence. As the IoT ecosystem expands, prioritizing security is essential to minimize risks and maximize the benefits of connected devices. Given the ever-expanding cyber threat landscape, the multiclass classification task is essential to empower the NIDS with an ability to distinguish between various attack patterns in less computational time. The multiclass decision jungle algorithm is investigated to optimize the performance of NIDS. The research has considered permutation feature importance to include only the relevant features from the data. Using a contemporary dataset such as CICIOT 2023, the study has demonstrated an impressive attack detection rate of over 90% for 20 modern attack types. This research has investigated the effectiveness of IoT security measures and its prospective contributions to the field of cyber security.

Corresponding Author:

Smitha Rajagopal
Alliance School of Advanced Computing, Alliance University
Bengaluru, India
Email: smitha.research1012@gmail.com

1. INTRODUCTION

The network landscape is highly vulnerable to security threats. The constant emergence of newer threats is paramount in developing reliable and adaptive security measures. The method by which an attacker gains access to a network, thereby sending malicious packets to a user's system to steal, modify, or ravage confidential data is termed intrusion [1]. Existing system vulnerabilities, such as human errors, misconfiguration, or software bugs can lead to an attack on the server or device. As and when humungous data is being exchanged on the Internet, networks become increasingly susceptible to attacks. Therefore, intrusion detection systems (IDS) are required to safeguard organizational networks.

The quantity of internet of things (IoT) connections has improved over the past decade. This trend is anticipated to maintain in numerous sectors in the coming years, however demanding situations continue to be. It ought to be fixed to make certain that IoT gadgets operate appropriately and effectively [2]. Implementing security features under IoT is more complicated than implementing them on traditional networks because of the extensive form of protocols such as all applicable node quantities [3]. Through

distributed denial-of-carrier (DDoS) assaults, targeting IoT gadgets such as virtual cameras and digital video recorders (DVRs), is an instance of a cyber hazard that constantly happens in the IoT landscape.

Internet of things (IoT) networks have unique characteristics that set them apart from wireless sensor networks (WSN) and cyber physical systems (CPS). This distinction arises from the heterogenous layers of protocols employed in IoT networks. Additionally, the challenges encountered during the deployment of various IoT use cases vary significantly from those associated with WSN networks, due to the specific context and requirements of each application [4], [5]. To enhance security in an IoT environment, it is crucial to design a data-oriented security mechanism. This mechanism should prioritize data confidentiality and integrity to mitigate threats. Traditional cryptography-based security measures may not be ideal for IoT due to the vast amount of data involved [6]. Figure 1 illustrates the IoT security landscape.

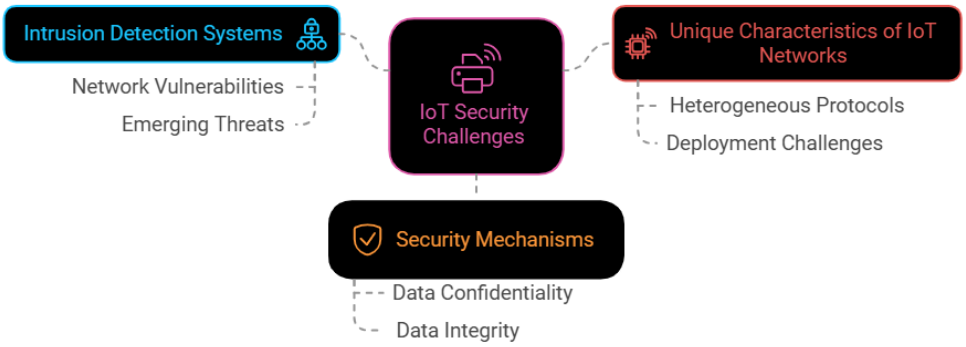

Figure 1. IoT security landscape

IDS can be divided into different types. For example, they can be grouped as host-based or network-based. Network-based systems watch over network traffic and devices for any unusual activities, while host-based systems check individual devices for changes in files, logs, or behaviors that don't match what's expected [7], [8]. Typically, IDS can be analyzed using two main methods: signature-based and anomaly-based systems. Signature-based IDS, like SIDS and AIDS, detect attacks by looking for known patterns stored in a database. SIDS specifically searches for these predefined patterns, while AIDS keeps track of normal system behavior and sends alerts when there are big differences from what's considered normal [9], [10].

This research aims to strengthen security in the field of IoT by creating and evaluating an enhanced intrusion detection system (IDS) that uses multiclass decision jungle (MDJ) algorithm [11]. By employing a sophisticated algorithm like MDJ, the IDS aims to detect and guard against a broad spectrum of cyber threats that target IoT protocols. The IDS framework incorporates the MDJ complemented by feature engineering and data cleansing techniques to improve classification accuracy. Through exhaustive performance testing, this research intends to offer cutting-edge insights into the field of IoT security. The rationale behind using multiclass decision jungle for multiclass classification is delineated below:

a. Network intrusion detection systems (NIDS) need to classify network traffic into different categories, not limiting to only benign or nefarious. The multiclass decision jungle is robust enough to detect various types of intrusions.
b. The constantly changing patterns of network traffic and the growing number of cyber threats require feasible solutions. Decision jungles, which use appropriate decision-making processes and being an ensemble of DAGs can handle these changes better than simpler methods.
c. The decision jungle algorithm is light-weight and can easily scale to handle more data, making it a good fit for systems with limited resources, such as IoT devices. It can quickly manage large amounts of data without compromising on the effectiveness.
d. False positives and false negatives pose a considerable challenge for NIDS, as they can inundate administrators and undermine their confidence in the system. Decision jungles optimize the parameters to reduce false alerts while efficiently detecting the majority of threats.
e. The multiclass decision jungle algorithm demonstrates exceptional proficiency in managing imbalanced datasets, which is a prevalent challenge in the network intrusion detection study where instances of

attacks are substantially outnumbered by regular traffic patterns. By utilizing its ensemble mechanism, a decision jungle significantly improves detection accuracy for both frequently occurring and rare attack categories.

Network intrusion detection systems face challenges in detecting anomalies efficiently. The first challenge pertains to the fact that malicious attacks and threats are constantly evolving, making it difficult for existing intrusion detection systems to contend with. This often leads to low detection rates and high false alarms. The second challenge is that the traditional machine learning algorithms applied in the field of network intrusion detection suffer from problems such as overfitting, where the model performs well on training data but does not perform well on unknown instances and the presence of high bias due to irrelevant features. Additionally, the class distribution of network traffic is often unbalanced, with normal traffic being much more common than nefarious traffic due to which the machine learning model may fail to learn the network patterns effectively [12], [13].

2. LITERATURE REVIEW

The research community is striving hard to explore the areas of the IoT and intrusion detection systems (IDS) to develop new ways to improve security. This section focuses on some of the latest IDS ideas that have been suggested. A key resource in this area is the Edge-IIoTset dataset [14], which is a detailed collection of data about cybersecurity for IoT that can be used by machine learning (ML) algorithms. Experts have experimented with different techniques that fuse various algorithms to detect network intrusions effectively and improve how well they work. One of these techniques uses a combination of a neutrosophic logic classifier (a more advanced version of fuzzy logic) and a genetic algorithm to create rules. This method has been successful in lowering the rate of false alarms to just 3.19%, which is better than many other methods [15].

In a study by Prazeres N, Costa R, Santos L, and their team, they proposed a distributed IDS with a strong and dependable design for fog computing. They used two datasets, IoT-23 and MQTT-IoT-IDS2020, to create IDS models. For the IoT-23 dataset, they used random forest (RF) and naive Bayes (NB) models, and for the MQTT-IoT-IDS2020 dataset, they used logistic regression (LR) and decision tree (DT) models. These models were then compared to three other IDS architectures designed for IoT traffic. The comparison was based on performance metrics like accuracy, precision, recall, and F1-score [16].

Researchers utilized the UNSW-NB15 network traffic dataset to evaluate ML models for IDS enhancement in More et al. Specifically, they employed logistic regression (LR), support vector machines (SVM), decision trees (DT), and random forest (RF). To optimize the performance of LR, DT, and linear SVM, the study conducted hyperparameter tuning that led to a good performance [17]. In a study published in [18], researchers evaluated the effectiveness of eight different tree-based classification algorithms for predicting network events using the NSL-KDD dataset. Amongst these algorithms, the decision tree algorithm was employed for feature selection, while a random forest algorithm was utilized as the primary classifier.

Aamir and Zaidi [19] proposed an intrusion detection model that used principal component analysis (PCA) and a subset of the benchmark dataset containing novel attack vectors. They used clustering to label the data and identify attacker classes based on normal traffic patterns. After labeling, they employed three machine learning algorithms (SVM, k-nearest neighbor (K-NN), and RF) to achieve detection accuracy levels of 92%, 95%, and 96.66%, respectively.

Alqahtani and his team [20] used popular machine learning methods like Bayesian network, naive Bayes, decision trees, random decision forest, random tree, decision table, and artificial neural network to detect network intrusions. They tested these methods using the KDD cup 99 dataset. In another study [21], researchers looked into detecting network intrusions using the UNSW-NB15 and CICIDS 2017 datasets. They incorporated gradient boosting tree for binary classification and deep neural network for multiclass classification. Del-IoT [22], an ensemble learning model for anomaly detection using software defined networks (SDN) was proposed to identify the attack scenarios on the dynamic cyber threat landscape. Deep feature extraction was used to input the features to a probabilistic neural network (PNN) in order to optimize the performance.

Table 1 provides a summary of datasets with and without IOT traces available for network intrusion detection. In the proposed study, CICIoT 2023 dataset has been used to for experimentation. The CICIoT 2023 dataset is primarily used for testing IoT systems because it simulates real-world IoT environments with a wide range of devices. These devices can behave as both targets of attacks or sources of attacks. With the growing prevalence of the IoT, it has essayed a critical role in our daily lives. As a result, ensuring its security has become paramount to facilitate its seamless, secure, and reliable operation.

Table 1. Summary of relevant network intrusion detection datasets

Sl. No	Dataset	Description	No of Features	Whether an IOT dataset Yes/No
1.	Edge-IIOT [14]	Researchers proposed the edge-industrial internet of things (IIoT), a comprehensive dataset to enhance intrusion detection systems for IoT and IIoT applications. It enables machine learning-based detection in two modes: centralized and federated learning. The researchers generated an Edge-IIoT set using a custom IoT/IIoT testbed, incorporating a wide range of devices, sensors, protocols, and cloud/edge configurations to provide a realistic representation of these systems.	61	Yes
2.	N-BaIOT [23]	Researchers developed a new dataset for detecting abnormal activities in IoT networks called network based internet of things (BaIoT). "Snapshots" of the network's behavior were extracted and advanced algorithms called deep autoencoders were employed. These algorithms analyse the snapshots and identify unusual traffic patterns that may indicate compromised or malicious IoT devices within the network.	115	Yes
3.	BoT-IoT [24]	BoT-IoT dataset, termed as a big data dataset by the authors, consists of 73 million instances. As discussed [24], the usage of this dataset is recommended after a thorough data cleaning procedure followed by the usage of valid features.	43	Yes
4.	MQTT-IoT-IDS 2020 [16]	The MQTT-IoT-IDS2020 dataset comprises data generated by a simulated MQTT network. It consists of unprocessed pcap files, along with unidirectional and bidirectional flow features. The prospective architectures, when proposed using this dataset may allow for effective feature engineering to produce better results	32	Yes
5.	UNSW NB-15 [25]	The UNSW-NB15 dataset aims to create realistic network environments by including common low-profile cyberattacks. It features ten attack types. The article [25] provides a detailed breakdown of the number of records per attack type and their distribution in the training and testing sets.	49	No
6.	CICIDS 2017 [26]	The dataset includes modern attacks that accurately resemble actual real-world data. It also includes the findings of network traffic analysis conducted by the CIC-Flow Meter, which labels data flows based on factors like timestamp, source and destination IP addresses, source and destination ports, protocols, and types of attacks stored in CSV files.	17	No

The CICIoT 2023 dataset was created using 105 devices and 33 real-world attacks. These attacks cover seven categories, such as DDoS, DoS, Recon, web-based attacks, brute force, spoofing, and Mirai. In all cases, harmful IoT devices attack other IoT devices. This dataset includes new types of attacks that aren't found in other IoT datasets. It helps IoT experts build new security tools by providing data in different formats. The CICIoT2023 dataset advances the field of IoT security by introducing a comprehensive network topology featuring diverse IoT devices. It incorporates multiple cyberattacks not previously included in a single IoT security dataset. Assessing the performance of popular machine learning methods against different attack types present in the CICIoT 2023 dataset provides a significant advantage compared to other studies.

3. METHODOLOGY

This section provides a summary of the experimental findings. It proposes a method for carrying out classification and detection tasks using multiclass decision jungle. The classification framework of the proposed work is depicted in Figure 2. Decision jungles are an improved algorithm in the field of machine learning that build on decision trees and decision forests. Instead of having a strict branching structure like trees, decision jungles use directed acyclic graphs (DAGs). This allows decision points (represented by nodes) to be used in multiple branches. By doing this, decision jungles can be more efficient and accurate for making predictions. The number of instances considered for training and testing in the ratio 80:20 is enumerated in Table 2.

In the proposed work, we have used Azure machine learning as a platform to create the machine-learning pipeline. Azure machine learning, a machine-learning platform hosted on the cloud, allows users to create machine-learning models that can be adjusted to various workloads, shared across multiple devices, and deployed directly onto the cloud [27]. Azure machine learning platform [28] was used to compare the performance of various classifiers.

The study uses multiclass decision jungle, an algorithm suitable for tasks involving classifying data into multiple attack categories. This is particularly appropriate for the task of identifying different types of attacks, as there are 34 distinct attack types to distinguish between in this case. Decision jungles have a structured hierarchy, providing better overall performance and stability. On the other hand, decision forests prioritize variety by using bagging and selecting features randomly. The decision jungle model has a more

intricate structure organized in a hierarchical manner. This design allows for enhanced identification of patterns and potentially improves the model's precision and resilience.

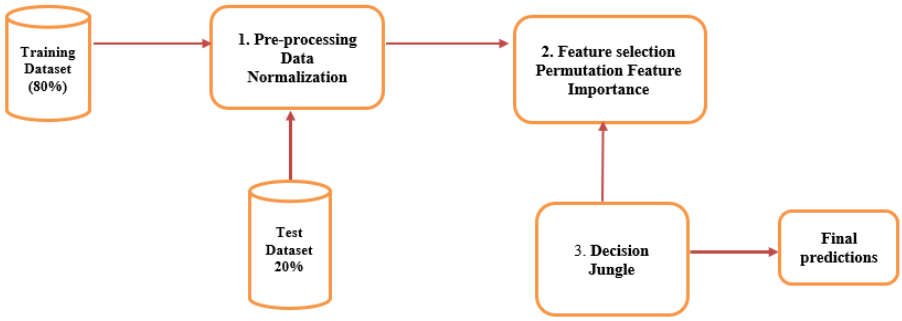

Figure 2. Methodology of the proposed work

Table 2. Training and testing samples considered for experimentation

Sl. No	Attack Category	Training set	Testing set
1	Backdoor_Malware	61	15
2	BenignTraffic	19581	4895
3	BrowserHijacking	112	28
4	CommandInjection	84	21
5	DDoS-ACK_Fragmentation	5145	1286
6	DDoS-HTTP_Flood	501	125
7	DDoS-ICMP_Flood	129025	32256
8	DDoS-ICMP_Fragmentation	8178	2045
9	DDoS-PSHACK_Flood	73916	18479
10	DDoS-RSTFINFlood	72658	18165
11	DDoS-SlowLoris	394	99
12	DDoS-SYN_Flood	73315	18329
13	DDoS-SynonymousIP_Flood	64544	16136
14	DDoS-TCP_Flood	81034	20259
15	DDoS-UDP_Flood	96964	24241
16	DDoS-UDP_Fragmentation	5145	1286
17	DictionaryBruteForce	259	65
18	DNS_Spoofing	3227	807
19	DoS-HTTP_Flood	1845	461
20	DoS-SYN_Flood	109481	27370
21	DoS-TCP_Flood	128880	32220
22	DoS-UDP_Flood	156794	39198
23	Mirai-greeth_flood	17692	4423
24	Mirai-greip_flood	13562	3390
25	Mirai-udpplain	16133	4033
26	MITM-ArpSpoofing	5615	1404
27	Recon-HostDiscovery	2406	601
28	Recon-OSScan	1780	445
29	Recon-PingSweep	33	10
30	Recon-PortScan	1490	373
31	SqlInjection	98	24
32	Uploading_Attack	18	5
33	VulnerabilityScan	647	174
34	XSS	58	14

DAGs typically produce decisions that lead to relatively minimum data storage, resulting in excellent overall performance. The multiclass aspect of decision jungle does not rely on specific distribution assumptions, enabling it to capture nonlinear boundaries between classes effectively. Decision Jungle can also identify relevant features and classify data while being resistant to noisy features during training, making it robust.

As compared to the traditional trees, each node in the decision jungle can have multiple paths leading to the reduction in the number of nodes. The decision jungle model enhances the random forest model by structuring trees in distinct layers, akin to a layered forest. Each layer is treated as an individual forest, and the predictions from one layer become the inputs for the subsequent layer, enabling a hierarchical decision-making process.

Mathematically, within the expanse of the decision jungle, there exist numerous decision trees, each denoted as T_1, T_2 ,..,…T_m where 'm' represents the total number of trees present. The final prediction is determined by combining the predictions made by each individual decision tree in the ensemble.

$$y = \arg\max \sum_{j=1}^{m} I(Tj(X) = y \quad (1)$$

The indicator function (denoted as "I") assigns a value of 1 to elements that satisfy the specified condition and 0 to elements that do not. In a multiclass classification task, the objective is to determine whether an instance from the provided data collection belongs to a particular predefined group or not.

3.1. Pre-processing

Data cleansing is a essential step in the machine learning domain. It involves handling incomplete or wrong records, as well as doing away with outliers and duplicates. The dataset used in this research included 33 specific types of IoT attack patterns and 46 numerical features. characteristic scaling is significant for algorithms that depend on distance-based calculations. If the records is on diverse scales, some statistics factors might affect on the model's performance. This pre-processing step allows algorithm work better and more reliably. A module known as "configure normalize data" from Azure service was used to normalize the data.

3.2. Feature selection

In order to select the most relevant features from the dataset, the component considered from Azure Machine Learning Studio is permutation feature importance (PFI), a filter-based feature selection method. This component evaluates the importance of features in a machine-learning model. It shuffles feature values randomly for each column, then compares the model's performance before and after each shuffle. Higher performance changes indicate more important features, as these are more affected by the shuffling. The twelve features that were considered for further processing are as follows, Header_length, rate, drate, syn_count, totsum, IAT, magnitude, Avg, std, covariance and radius. It is noteworthy that the above mentioned twelve features contributed towards the prediction outcome.

3.3. Decision Jungle as the classifier

The number of optimization steps configured for each layer in the decision DAG specifies how many steps the system should take to optimize that particular layer. 2048 was assigned the value for the number of optimization steps. Eight decision DAGs were configured in order to perform the classification task. The maximum depth of the decision DAGs was considered as 32. A value of 32 was assigned as the width of the DAG.

4. RESULTS AND DISCUSSION

Given 1,048,575 instances, 838,860 (80%) were used for model training, while the remaining 209,715 (20%) were utilized for testing. To ensure the reliability of the model, a rigorous technique called ten-fold cross-validation was carried out. Typically, cross-validation is often applied to validate the effectiveness of the proposed model thereby avoid overfitting [29]. This process divides the data into ten equal parts and trains the model on nine of the parts while evaluating it on the remaining part. By repeating this process for all ten combinations, the algorithm is subjected to a variety of data subsets, providing a more accurate assessment of its overall performance. Table 3 summarizes the performance metrics of the decision jungle classifier, showcasing its effectiveness.

The performance metrics, normally Table used for a predictive study in machine learning pertaining to multiclass classification are precision, recall and F1-score as shown in the (2) to (4).

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (2)$$

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (3)$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

The proposed model could successfully identify benign patterns. However, there have been a few misclassifications wherein benign patterns have been incorrectly classified to be DNS_spoofing and

MITM_Arpspoofing. The proposed version successfully discovered instances of browser hijacking, detecting 99% of these attacks. It additionally detected many different sorts of attack vectors, including specific forms of DDoS attacks (like ACK fragmentation, HTTP flood, ICMP flood, ICMP fragmentation, PSHACK Flood, RSTFIN Flood), DoS assaults (like HTTP Flood, SYN Flood, TCP Flood, UDP Flood), Mirai botnet assaults (such as greeth flood, greip flood, udpplain flood), and vulnerability scans.

The number of samples found in datasets pertaining to different categories may not be the same. This is a common issue that people working with machine learning face, typically known as having an imbalanced dataset [30], [31]. When testing for backdoor malware and command injection, decision jungle classifier was able to correctly identify 62.5% of the cases it was supposed to classify. The proposed version successfully observed cases of browser hijacking, detecting 99% of these assaults. It also did fairly in detecting many other forms of attacks, such as different kinds of DDoS attacks (like ACK fragmentation, HTTP flood, ICMP flood, ICMP fragmentation, PSHACK flood, RSTFIN flood), DoS assaults (like HTTP Flood, SYN Flood, TCP Flood, UDP Flood), Mirai botnet attacks (such as greeth flood, greip flood, udpplain flood), and vulnerability scans.

Table 3. Precision, recall and F1-scores of 34 attack types

Sl. No	Attack category	Precision	Recall	F1-score
1	Backdoor_Malware	0.625	0.5882353	0.7507042
2	BenignTraffic	0.8792244	0.9766154	0.9253644
3	BrowserHijacking	1	0.3478261	0.516129
4	CommandInjection	0.625	0.2941176	0.4
5	DDoS-ACK_Fragmentation	0.9977099	0.9992355	0.9984721
6	DDoS-HTTP_Flood	0.9916667	0.9444444	0.9674797
7	DDoS-ICMP_Flood	0.9999065	0.9999065	0.9999065
8	DDoS-ICMP_Fragmentation	0.998009	0.998009	0.998009
9	DDoS-PSHACK_Flood	0.9998371	0.9998371	0.9998371
10	DDoS-RSTFINFlood	0.9999447	0.9999447	0.9999447
11	DDoS-SlowLoris	0.9081633	0.9175258	0.9128205
12	DDoS-SYN_Flood	0.9997315	0.9998389	0.9997852
13	DDoS-SynonymousIP_Flood	0.9996886	0.9998754	0.999782
14	DDoS-TCP_Flood	0.9998524	0.9998032	0.9998278
15	DDoS-UDP_Flood	0.9996705	0.9997528	0.9997116
16	DDoS-UDP_Fragmentation	0.9992424	0.9954717	0.9973535
17	DictionaryBruteForce	0.7173913	0.55	0.6226415
18	DNS_Spoofing	0.7841823	0.6874266	0.7326237
19	DoS-HTTP_Flood	0.9871383	0.9871383	0.9871383
20	DoS-SYN_Flood	0.9992244	0.9991137	0.999169
21	DoS-TCP_Flood	0.9996645	0.9997484	0.9997064
22	DoS-UDP_Flood	0.99973	0.9995277	0.9996288
23	Mirai-greeth_flood	0.9991013	0.9986526	0.9988769
24	Mirai-greip_flood	0.9988338	0.9985427	0.9986882
25	Mirai-udpplain	0.9990354	0.9997587	0.9993969
26	MITM-ArpSpoofing	0.8809148	0.795584	0.8360778
27	Recon-HostDiscovery	0.8428835	0.7862069	0.8135593
28	Recon-OSScan	0.7765043	0.6008869	0.6775
29	Recon-PingSweep	1	0.1111111	0.2
30	Recon-PortScan	0.7754386	0.594086	0.6727549
31	SqlInjection	0.5555556	0.3333333	0.4166667
32	Uploading_Attack	1	0.7142857	0.8333333
33	VulnerabilityScan	0.9830508	1	0.991453
34	XSS	0.8333333	0.3571429	0.5

It is noteworthy that 20 attack types considered in the dataset have been detected with a precision score of 90% and above, a feat quite vital in any multiclass classification task. Figure 3 demonstrates clearly the performance exhibited by decision jungle. Figure 4 depicts the precision score pertaining to the 34 attack types. Figure 5 represents the recall score pertaining to the 34 attack vectors. The decision jungle algorithm marks a significant step forward in the field of machine learning, especially when it comes to handling multiclass classification tasks. Precision is based on the proportion of accurate predictions made for a specific class out of all the instances predicted as that class. Sensitivity or true positive rate (also known as recall) is used to measure the proportion of actual instances of a given class that were accurately predicted.

The decision jungle model proposed in this research work has reported a promising precision score for three attack types namely Browser-hijacking, Recon-Pingsweep and Uploading-attack. Hijacked browsers have the capability to acquire sensitive information, including login credentials, financial data, or private

communications and thus can be a potential risk to the privacy and security of both individuals and organizations. Recon-Pingsweep is more often the initial step for a wide scale cyber attack.

Cybercriminals can make a focused attack, if they find the active devices and their IP addresses. IoT devices normally support file uploads for both firmware updates and configurations. An uploading attack which is successful can affect these devices and attackers can assume the control of them. Therefore, the intrusion detection model should possess the capability to correctly identify these attacks. A few attack types namely Recon-Osscan, sqlinjections and XSS were not correctly identified by the proposed model and initiatives are taken in this regard to address the issues of scalability and computational challenges.

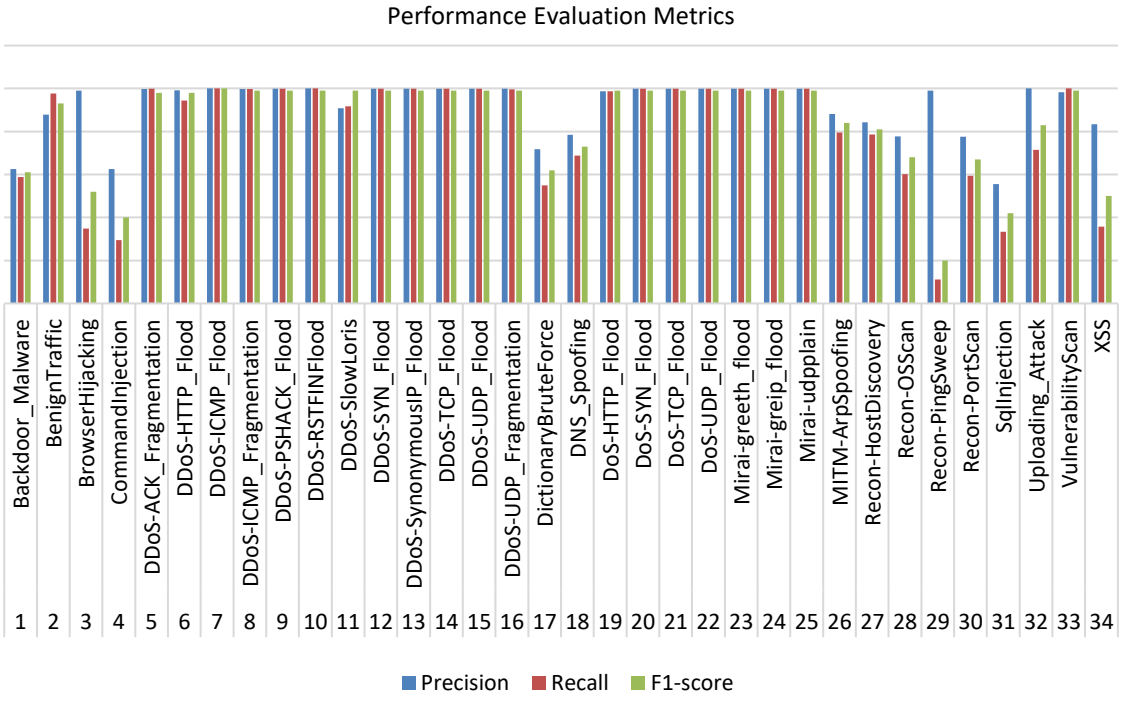

Figure 3. Performance of the model concerning precision, recall and F1-score

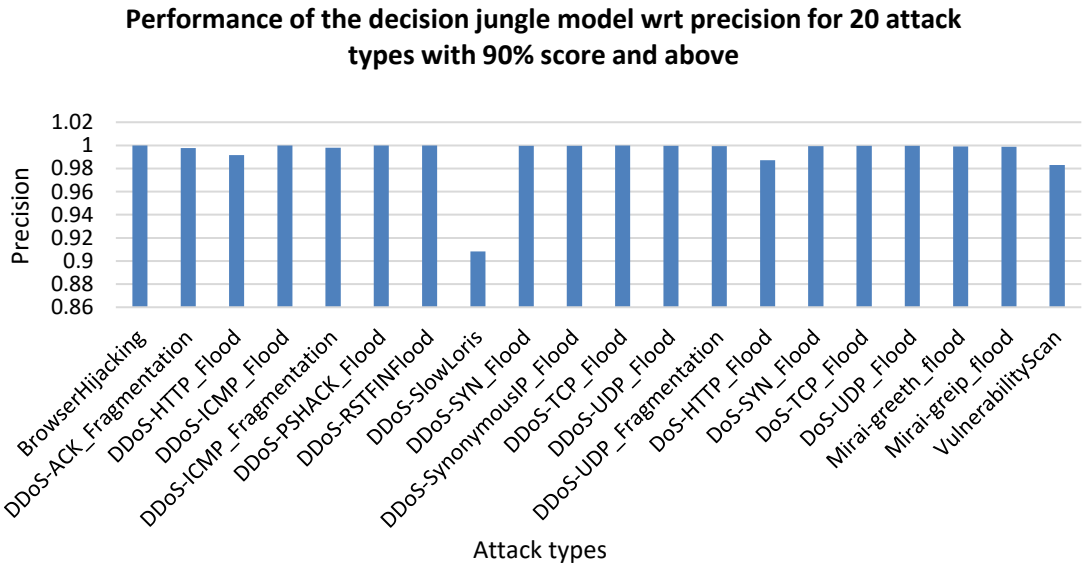

Figure 4. Precision score of 90% and above for 20 attack types

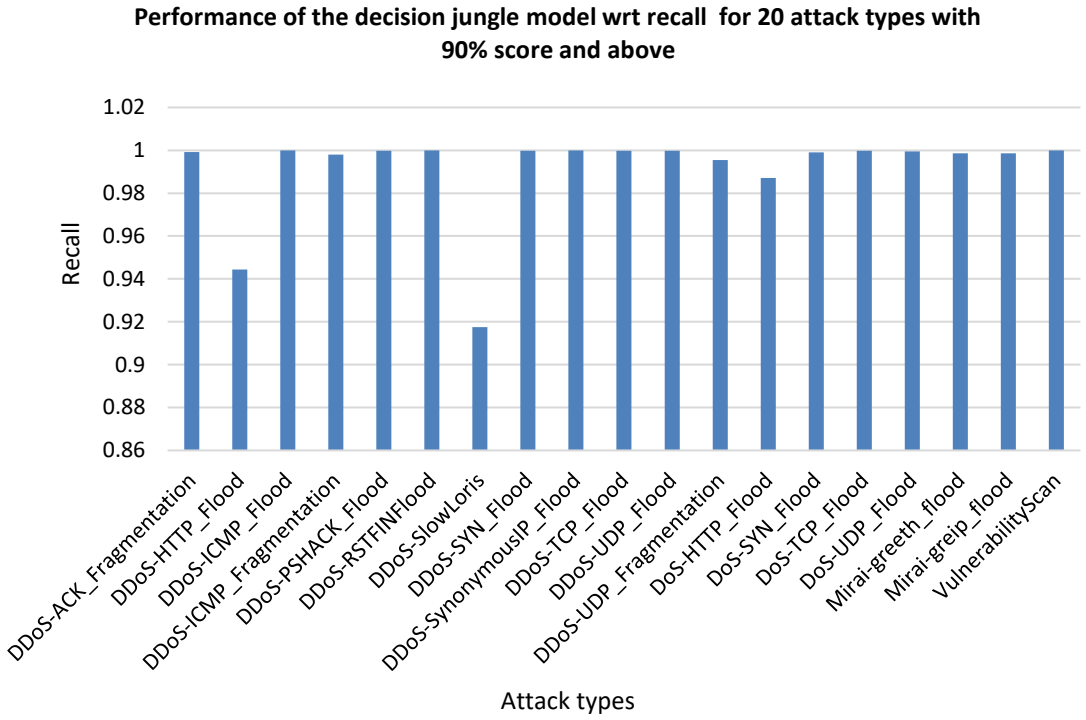

Figure 5. Recall score of 90% and above for 20 attack types

The F1-score evaluates performance for each category separately and combine these into a single metric to ensure that minority classes are not overshadowed by majority classes. Therefore, F1-score is a critical evaluation parameter. In the context of multiclass classification, the F1-score gives a balanced way to validate the effectiveness of the model. This is especially useful in real-time situations where it is hard to have both high precision and high recall at the same time [32], [33]. As discussed in [34], F1-score balances recall and precision scores using the harmonic mean. The proposed decision jungle model has displayed an F1-score of 0.9 and above for 20 attack types as presented in Figure 6.

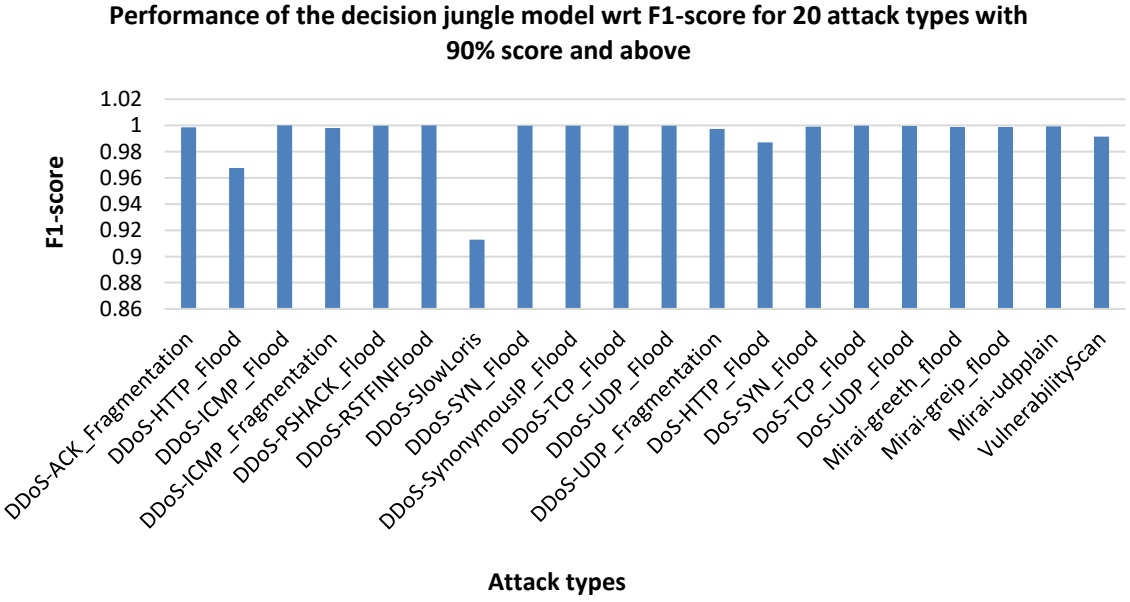

Figure 6. F1-score of 90% and above for 20 attack types

5. CONCLUSION

The goal of this study was to establish the importance of multiclass classification for network intrusion detection from the perspective of decision jungle. The proposed work affirms that decision jungle is indeed a sophisticated algorithm to detect various attack patterns that belong to different attack categories. To establish that our method works well, we used the CICIOT 2023 dataset, which comprises IOT intrusion patterns. When we tested it on Azure cloud, we found that the decision jungle algorithm works well for detecting network intrusions, especially for classifying different types of attacks. Instead of using single systems, in our work, we mainly focused on how fast the system runs and performed the tests using Microsoft Azure machine learning studio (MAMLS). The multiclass decision jungle predictor took one minute and fifty seconds to execute successfully. The decision jungle algorithm possesses immense capabilities to delineate and counter an expansive array of security risks. It also exhibits a good performance while doing classification tasks that are so technically challenging, per se. The proposed model developed using decision jungle algorithm perfomed at low computing costs hence establishing its viability in real-time protection of IoT applications. The proposed study also suggests that the decision jungle algorithm enhances the appropriate detection of attacks, but even more important, it decreases the number of false positives primarily needed in critical applications such as network intrusion detection.

ACKNOWLEDGMENTS

We thank Alliance University for all the resources provided to us for the successful execution of this research work.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Smitha Rajagopal	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	
Abhik Sarkar	✓				✓	✓		✓	✓	✓	✓	✓	✓	
Venkat Narayanan		✓	✓	✓			✓			✓				
Manjunath														

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in UNB at https://www.unb.ca/cic/datasets/iotdataset-2023.html

REFERENCES

[1] M. A. Talukder *et al*., "A dependable hybrid machine learning model for network intrusion detection," *Journal of Information Security and Applications*, vol. 72, p. 103405, Feb. 2023, doi: 10.1016/j.jisa.2022.103405.
[2] M. binti Mohamad Noor and W. H. Hassan, "Current research on internet of things (IoT) security: a survey," *Computer Networks*, vol. 148, pp. 283–294, Jan. 2019, doi: 10.1016/j.comnet.2018.11.025.
[3] M. Wang, Y. Lu, and J. Qin, "A dynamic MLP-based DDoS attack detection method using feature selection and feedback," *Computers and Security*, vol. 88, p. 101645, Jan. 2020, doi: 10.1016/j.cose.2019.101645.

[4] R. Mitchell and I. R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–29, Mar. 2014, doi: 10.1145/2542049.
[5] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for IoT: Toward universal and resilient systems," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018, doi: 10.1109/COMST.2018.2844742.
[6] Dr. S. Smys, Dr. Abul Basar, and Dr. Haoxiang Wang, "Hybrid intrusion detection system for internet of things (IoT)," *Journal of ISMAC*, vol. 2, no. 4, pp. 190–199, Sep. 2020, doi: 10.36548/jismac.2020.4.002.
[7] S. N. Mighan and M. Kahani, "A novel scalable intrusion detection system based on deep learning," *International Journal of Information Security*, vol. 20, no. 3, pp. 387–403, Jun. 2021, doi: 10.1007/s10207-020-00508-5.
[8] M. M. Obaid and M. H. Saleh, "Efficient intrusion detection through the fusion of AI algorithms and feature selection methods," *Journal of Engineering*, vol. 30, no. 07, pp. 184–201, Jul. 2024, doi: 10.31026/j.eng.2024.07.11.
[9] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021, doi: 10.1016/j.procs.2021.05.025.
[10] H. Albasheer *et al.*, "Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey," *Sensors*, vol. 22, no. 4, p. 1494, Feb. 2022, doi: 10.3390/s22041494.
[11] J. Shotton, S. Nowozin, T. Sharp, J. Winn, P. Kohli, and A. Criminisi, "Decision jungles: Compact and rich models for classification," *Advances in Neural Information Processing Systems*, 2013.
[12] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: A two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019, doi: 10.1109/ACCESS.2019.2899721.
[13] J. Li, Y. Qu, F. Chao, H. P. H. Shum, E. S. L. Ho, and L. Yang, "Machine learning algorithms for network intrusion detection," in *Intelligent Systems Reference Library*, vol. 151, 2019, pp. 151–179. doi: 10.1007/978-3-319-98842-9_6.
[14] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022, doi: 10.1109/ACCESS.2022.3165809.
[15] B. Kavitha, D. S. Karthikeyan, and P. Sheeba Maybell, "An ensemble design of intrusion detection system for handling uncertainty using Neutrosophic Logic Classifier," *Knowledge-Based Systems*, vol. 28, pp. 88–96, Apr. 2012, doi: 10.1016/j.knosys.2011.12.004.
[16] N. Prazeres, R. L. de C. Costa, L. Santos, and C. Rabadão, "Engineering the application of machine learning in an IDS based on IoT traffic flow," *Intelligent Systems with Applications*, vol. 17, p. 200189, Feb. 2023, doi: 10.1016/j.iswa.2023.200189.
[17] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced intrusion detection systems performance with UNSW-NB15 data analysis," *Algorithms*, vol. 17, no. 2, p. 64, 2024, doi: 10.3390/a17020064.
[18] S. Thaseen and C. A. Kumar, "An analysis of supervised tree based classifiers for intrusion detection system," in *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, PRIME 2013*, Feb. 2013, pp. 294–299. doi: 10.1109/ICPRIME.2013.6496489.
[19] M. Aamir and S. M. Ali Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 4, pp. 436–446, 2021, doi: 10.1016/j.jksuci.2019.02.003.
[20] H. Alqahtani, I. H. Sarker, A. Kalim, S. M. Minhaz Hossain, S. Ikhlaq, and S. Hossain, "Cyber intrusion detection using machine learning classification techniques," in *Communications in Computer and Information Science*, vol. 1235 CCIS, Springer Singapore, 2020, pp. 121–131. doi: 10.1007/978-981-15-6648-6_10.
[21] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *ACMSE 2019 - Proceedings of the 2019 ACM Southeast Conference*, Apr. 2019, pp. 86–93. doi: 10.1145/3299815.3314439.
[22] E. Tsogbaatar *et al.*, "DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT," *Internet of Things (Netherlands)*, vol. 14, p. 100391, Jun. 2021, doi: 10.1016/j.iot.2021.100391.
[23] Y. Meidan *et al.*, "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, Jul. 2018, doi: 10.1109/MPRV.2018.03367731.
[24] J. M. Peterson, J. L. Leevy, and T. M. Khoshgoftaar, "A review and analysis of the Bot-IoT dataset," in *Proceedings - 15th IEEE International Conference on Service-Oriented System Engineering, SOSE 2021*, Aug. 2021, pp. 20–27. doi: 10.1109/SOSE52839.2021.00007.
[25] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.
[26] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "A detailed analysis of the CICIDS2017 data set," in *Communications in Computer and Information Science*, vol. 977, 2019, pp. 172–188. doi: 10.1007/978-3-030-25109-3_9.
[27] A. Shivanna and D. P. Agrawal, "Prediction of defaulters using machine learning on Azure ML," in *11th Annual IEEE Information Technology, Electronics and Mobile Communication Conference, IEMCON 2020*, Nov. 2020, pp. 320–325. doi: 10.1109/IEMCON51383.2020.9284884.
[28] S. Rajagopal, K. S. Hareesha, and P. P. Kundapur, "Performance analysis of binary and multiclass models using azure machine learning," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 978–986, Feb. 2020, doi: 10.11591/ijece.v10i1.pp978-986.
[29] M. T R, V. K. V, D. K. V, O. Geman, M. Margala, and M. Guduri, "The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification," *Healthcare Analytics*, vol. 4, p. 100247, Dec. 2023, doi: 10.1016/j.health.2023.100247.
[30] S. Ayoub, Y. Gulzar, J. Rustamov, A. Jabbari, F. A. Reegu, and S. Turaev, "Adversarial approaches to tackle imbalanced data in machine learning," *Sustainability*, vol. 15, no. 9, p. 7097, Apr. 2023, doi: 10.3390/su15097097.
[31] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, Apr. 2016, doi: 10.1007/s13748-016-0094-0.
[32] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," *arXiv:2008.05756*, Aug. 2020.
[33] B. Sidumo, E. Sonono, and I. Takaidza, "An approach to multi-class imbalanced problem in ecology using machine learning," *Ecological Informatics*, vol. 71, p. 101822, Nov. 2022, doi: 10.1016/j.ecoinf.2022.101822.
[34] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, "Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer," *Neuroscience Informatics*, vol. 2, no. 4, p. 100034, 2022, doi: 10.1016/j.neuri.2021.100034.

BIOGRAPHIES OF AUTHORS

Smitha Rajagopal is serving as an assistant professor at Alliance University, Bengaluru. She obtained her Ph.D. from Manipal Academy of Higher Education (MAHE). She did her Master's and Bachelor's from BMS College of Engineering, Bengaluru, and Presidency College, Bengaluru, respectively. Her Ph.D. dissertation integrates cloud computing and machine learning, focusing on the implementation of a cloud-based intrusion detection and prevention system. She has published articles in esteemed journals. Her research interests include cloud computing, artificial intelligence, cybersecurity, and machine learning. With patented research ideas, she has actively contributed to advancing current and emerging technologies, showcasing her commitment to innovation in these dynamic fields. She can be contacted at smitha.research1012@gmail.com.

Abhik Sarkar is a visual art academician working presently at the School of Liberal Arts, Alliance University, Bengaluru. He previously served as a JRF and SRF in the Department of Visual Arts at Assam University. He is also a practising Artist who has participated in several National and International art exhibitions at galleries like Indian council for Cultural Relations (ICCR), Indian Society of Oriental Art, Academy of Fine Arts, Birla Academy. Having authored books, he is always eager to broaden his critical assessments to encompass papers from diverse fields, ensuring objective and unbiased reviews. He can be contacted at abhiksarkar822@gmail.com.

Venkat Narayanan Manjunath is pursuing a Bachelor of Technology (B.Tech) in Computer Science Engineering at Alliance University, Bengaluru. He is passionate about emerging technologies and aspires to become a software engineer and pursue intradisciplinary research. He is inquisitive about artificial intelligence (AI) and machine learning (ML). Venkat is eager to learn and explore to make a difference in the fast-changing tech world. In this research endeavor, he has been supported and guided by Dr. Smitha Rajagopal. Venkat is enthusiastic to delve into the opportunities in AI and ML and is excited to build a successful career in these fields. He can be contacted at mayurvenkat185@gmail.com.