Naif Arab University for Security Sciences
Journal of Information Security and Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR

JISCR

A Comparative Evaluation of Machine Learning-Based Intrusion Detection Systems for Securing Cloud Environments

Mohammad Shadi Alhakeem*, Khawla Bin Ajlan
Naif Arab University for Security Sciences, Riyadh, Saudi Arabia

Received 21 Oct. 2024; Accepted 23 Dec. 2024; Available Online 31 Dec. 2024

Abstract

Cloud computing has advanced significantly alongside the growth of communication technology and data exchange. Many businesses and organizations now adopt cloud computing solutions and services to enhance flexibility and scalability. However, despite its numerous advantages, cloud computing remains increasingly susceptible to various security threats that can disrupt services and business operations. This highlights the critical need to strengthen the security of cloud environments. In this context, implementing robust protection measures, such as Intrusion Detection Systems (IDS), is essential to mitigate potential threats and safeguard sensitive data. To effectively counter the ever-evolving cyber threats landscape, IDS must possess adaptive capabilities. Hence, integrating Machine Learning (ML) technologies is imperative for the detection of a broad and diverse range of cyber threats, thereby enhancing the overall bolstering the security posture of the environment.

This research explores the integration of ML technologies in IDS and examines the application of feature selection methods to identify the key and most significant indicators for attack detection. The study conducts a comparative analysis of five ML techniques, employing two distinct feature selection methods to evaluate their effectiveness in strengthening the security of cloud environments. Using a recently developed, reputable dataset and concentrating on attack types that pose significant threats to cloud environments, our experimental results offer a comprehensive evaluation of these techniques, including a variety of machine learning algorithm performance metrics.

I. Introduction

With the growing advancement in communication technology and data exchange, many technologies have emerged and developed. One of these technologies is cloud computing, which shares computing resources and services to provide quick and easy access with minimal management effort. The National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1].

Keywords: cloud environments, cybercrimes, feature selection, intrusion detection systems, machine learning

Production and hosting by NAUSS

* Corresponding Author: Mohammad Shadi Alhakeem
Email: malhakeem@nauss.edu.sa

doi: 10.26735/RSND3740

Since its inception in the early 2000s, cloud computing has become an indispensable component of contemporary technological infrastructure. A multitude of applications and programs now rely on cloud-based services, either exclusively or in conjunction with traditional on-premises solutions. Numerous organizations have transitioned their computing resources to the cloud, thereby enhancing the accessibility, availability, and scalability of their services while concurrently reducing costs and operational overhead.

However, while cloud computing offers numerous benefits, it is not without its inherent challenges, particularly in terms of security and privacy. Cloud environments are susceptible to a wide range of cybersecurity threats, including unauthorized access to legitimate resources, IP Spoofing, SQL Injection, Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), and Brute Force attacks. A significant portion of cyberattacks target cloud services, underscoring the critical importance of robust security measures. These threats can disrupt services and business operations, compromise sensitive data, and damage an organization's reputation. In 2020, data revealed that attacks on cloud service platforms are accounted for 20% of all cyber-attacks, which makes these platforms the third most-targeted cyber environment [2]. For example, in March 2020, a data breach occurred on the CAM4 (live streaming) website, exposing over 10 billion sensitive records of data (7 TB). These exposed records included user and company information such as IP addresses, payment logs, usernames, and more. In another breach of 2020, a leaked Elasticsearch database believed to belong to a UK-based security company contained two datasets with 5 billion and 15 million entries, respectively [3]. Moreover, A recent report by CrowdStrike found that cloud environment intrusions increased by 75% from 2022 to 2023 [4].

Accordingly, there is a critical need to strengthen the security of cloud environments. Ensuring the security of computing resources is paramount to providing secure cloud services and safeguarding user privacy [5] [6]. In this scope, Intrusion Detection Systems (IDS) are considered a crucial tool for detecting cyberattacks within cloud infrastructures. IDS is a security management tool designed to monitor networks and systems for anomalous activity indicative of potential security breaches [7] [8]. By analyzing network traffic and system logs, IDS can proactively detect malicious or suspicious activities, and alert administrators to potential threats. This timely detection and response can prevent disruptions to critical services, unauthorized access, and data breaches, thereby safeguarding the key cybersecurity objectives of confidentiality, integrity, and availability (CIA) for various resources within the cloud environment.

Typically, IDSs can be categorized into two primary techniques: signature-based and anomaly-based (or behavior-based). Signature-based IDSs are designed to detect known attack patterns, but they are limited in their ability to identify novel or zero-day attacks. In contrast, anomaly-based IDS can effectively detect novel attack patterns [6] [9]. However, effective IDS implementation requires careful configuration and ongoing maintenance. Organizations should regularly update IDS signatures to stay abreast of emerging threats. Additionally, it is crucial to analyze the alerts generated by the IDS to determine whether they accurately indicate genuine threats or are false positives. Given the constantly evolving threat landscape and the emergence of increasingly sophisticated malicious attacks, there is a pressing need for innovative methods to prevent and mitigate these advanced threats within cloud environments.

Researchers are actively exploring and developing cutting-edge solutions to effectively analyze data and respond to threats in a timely manner. Within this context, Machine Learning (ML) algorithms have recently witnessed a surge in application to intrusion detection systems, with the goal of detecting intrusions and adapting to evolving patterns of normal behavior [10] [11].

Therefore, in this paper, we delve into the evaluation of machine learning (ML) techniques integrated into intrusion detection systems (IDS) specifically designed for cloud environments. In addition, as it is crucial to consider not only accuracy metrics but also computational efficiency, we explore the application of feature selection methods to identify the most significant indicators for

attack detection, which can reduce computational costs and improve model training time. The study conducts a comparative analysis of five ML algorithms, utilizing two distinct feature selection methods to assess their effectiveness in enhancing the security and protection of cloud environments. Our objective is to evaluate the efficiency of these approaches in distinguishing between malicious and benign activities, encompassing a variety of machine learning algorithm performance metrics, thereby providing valuable insights for improving IDS capabilities in cloud environments.

However, A significant challenge in developing effective machine learning solutions lies in ensuring that they are trained on realistic, real-world data to accurately learn and adapt to real-world scenarios. To address this issue, we conducted our comparative analysis using a recently developed, reputable dataset focusing on attack types that pose significant threats to cloud environments.

The remainder of this paper is organized as follows. Section II presents a comprehensive literature review of related works in the field of IDS and ML-based IDS specifically designed for cloud environments. Section III outlines the methodology employed for implementing and evaluating the ML algorithms. Section IV presents an analysis and discussion of the experimental evaluation results. Finally, Section V concludes the paper and proposes potential directions for future work.

II. Related Work

The design of anomaly-based Intrusion Detection Systems (IDS) typically encompasses several key stages, including: data preprocessing (removal of redundant data and data cleansing), feature selection or extraction, and classification [12]. These stages are crucial for constructing robust IDS solutions capable of effectively detecting cyberattacks. Each stage presents unique challenges and opportunities for optimization. Accordingly, this area continues to attract significant research interest in many fields concerned with mitigating cyber threats. For example, in [13] Musleh et al. focused on enhancing the accuracy and efficiency of IDS for Internet of Things (IoT) traffic. They evaluated several feature extraction techniques, including image filters and transfer learning models like VGG-16 and Dense Net. Additionally, they considered various machine learning algorithms, including random forest, K-nearest neighbors, SVM, and different stacked models. The study conducted a comprehensive evaluation of these combined models using the IEEE Data port dataset.

As cloud computing services proliferate across diverse applications demanding confidentiality, integrity, and availability, the landscape of cyber threats within this domain has also evolved significantly. This necessitates expanded research efforts to detect malicious intrusions in cloud environments. Within this scope, numerous research studies have focused on the development and evaluation of IDS, with a particular emphasis on ML-based approaches specifically designed for cloud environments, as well as the integration of feature selection techniques. Here, we present a comprehensive literature review of recent advancements in this field, and the key papers explored in this section are summarized in Error! Reference source not found..

Many research papers have focused on comparing the performance of different classifiers, such as [14], [15], [16], [17], [18] and [19]. Others have explored hybrid approaches that combine multiple classifiers, like [20], [21], [22], [23], [24], [25] and [26]. Additionally, several papers have highlighted the benefits of feature selection techniques, including [27], [28], [29] and [30].

Azeez et al. [14] conducted a comparative analysis of three different classifiers—Naïve Bayes, decision tree, and random forest—using the KDD'99 dataset. Preprocessing to remove unnecessary or incomplete values and perform relevant features extraction. All classifiers demonstrated promising results, with the random forest algorithm exhibited superior performance, achieving the highest accuracy in this study, followed by decision tree and Naïve Bayes.

Nathiya and Suseendran [15] also presented a comparative analysis of three classification algorithms for anomaly-based IDS: Support Vector Machine (SVM), Naïve Bayes, and Decision Tree (J48). The experiment was conducted on the NSL-KDD dataset. The results demonstrated that the

decision tree algorithm outperformed SVM and Naïve Bayes in terms of True Positive Rate (TPR) and False Positive Rate (FPR), while SVM excelled in detection time.

The study conducted by Peng et al. [16] initially subjected the data to preprocessing and cleansing, comprising string digitization and data normalization. Subsequently, the processed data was independently fed into three different classifiers: Naïve Bayes (NB), Decision Tree (DT), and K-Nearest Neighbors (k-NN) algorithms, using the KDD'99 dataset for comparison. The results indicated that the DT classifier achieved the highest accuracy and precision but exhibited a longer detection time compared to the other classifiers.

Kanimozhi and Jacob [17] conducted a comparative analysis to detect malicious traffic generated by botnet attacks. To achieve optimal results, they employed six classification algorithms: K-Nearest Neighbor, Naïve Bayes, Adaboost with Decision Tree, SVM, Random Forest, and Multi-Layer Perceptron (MLP). The CSE-CIC-IDS2018 dataset underwent preprocessing, including null value removal and standardization using "StandardScaler". The experimental results demonstrated that the Multi-Layer Perceptron (MLP) classifier outperformed the others, followed by Adaboost and Naïve Bayes.

In [18], Fitni and Ramli initially conducted data preprocessing, removing infinite values, missing data, and unnecessary columns. Subsequently, they applied correlation-based feature selection to extract 23 features from the original 80. Finally, they compared the performance of various classifiers: Gaussian Naïve Bayes, Random Forests, Decision Trees, Quadratic Discriminant Analysis (QDAs), Multi-Layer Perceptrons (MLPs), Logistic Regression, and Gradient Boosting, on the CSE-CIC-IDS2018 dataset. The results demonstrated that Gradient Boosting, Decision Trees, and Logistic Regression achieved excellent performance with rapid prediction times.

A recent study by Rathod et al. [19] conducted a comparative study demonstrating the superior performance of ML-based IDSs over traditional methods in intrusion detection. This suggests that ML-based approaches hold significant potential to improve the usefulness of intrusion detection in cloud computing environments. However, the study emphasized the need to address the challenges related to the substantial amounts of training data required for optimal performance.

In [20], Feng et al. introduced a novel ML approach called CVAC, which combined the Support Vector Machine (SVM) and Clustering based on Self-Organized Ant Colony Network (CSOACN). They employed the DARPA fxA'98 dataset and utilized the principal component analysis (PCA) method for feature selection. Both PCA and CVAC demonstrated promising results, validating the effectiveness of the combined approach compared to SVM and CSOACN used separately.

Kevric et al. [21] developed a framework that also combines ML classifiers. They combined Random Tree (multiple decision trees), NBTree (a hybrid of decision tree and Naïve Bayes classifiers), and C4.5 Decision Tree algorithms based on the sum rule scheme. Then an experimental comparison of individual and combined classifiers was conducted on the NSL-KDD dataset, with the combined classifiers demonstrating superior performance.

Similarly, Ahmim et al. [22] proposed a hybrid IDS based on the combination of the probability predictions of a tree of classifiers: Naïve Bayes (NB), Random Forest, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Radial-basis function network (RBFN), and Ripple Down Rule Learner (RIDOR). The system consists of two stages: a tree of classifiers and a final classifier that integrated the probabilistic predictions generated by the individual classifiers. Experimental results on the KDD'99 and the NSL-KDD datasets demonstrated the model's effectiveness in achieving reasonable detection rates.

Muttappanavar and Challagidad [23] proposed a hybrid machine learning-based IDS designed to detect novel attacks, including zero-day attacks, by combining supervised machine learning (Artificial Neural Network, ANN) and unsupervised machine learning (K-Means clustering). The proposed approach involved: feature selection by applying Principal Component Analysis (PCA) to select the most relevant features, using the K-Means algorithm to cluster the data and identify unknown attacks, and employing the pretrained ANN algorithm to

categorize unknown attacks based on behavioral similarities. However, while the paper provides a detailed explanation of the system's strategy and architecture, it lacked specific information regarding the practical experiments, such as the dataset used, and the tools employed.

Aljamal et al. [24] presented a hybrid approach that combines supervised and unsupervised machine learning techniques. Initially, K-Means clustering was applied to automatically predict labels for the unlabeled UNSW-NB15 dataset. Subsequently, an SVM model was built and trained using the labeled training dataset, serving as a detection system for new instances. Comparative analysis with other studies revealed that the K-Means clustering model effectively extracted relevant features and characterized the behaviors of different traffic types, contributing to an acceptable accuracy of 0.847 for the SVM classifier.

Sharma et al. [25] proposed a hybrid intrusion detection system that combined SVM, ELM (Extreme Learning Machine), and K-means clustering algorithms. The ensemble learning system demonstrated improved overall performance on the KDD'99 dataset, achieving a total accuracy of 95.75%.

Zhou et al. [26] employed Correlation-based Feature Selection (CFS) for feature selection and combined Random Forest (RF), C4.5, and Forest by Penalizing Attributes (Forest PA) in their intrusion detection system. Individual classifiers were applied separately, and the classification results were subsequently combined using the average of probabilities (AOP) rule to enhance the performance of a single classifier in distinguishing between benign and malicious traffic. The proposed model demonstrated superior predictive performance, achieving lower False Alarm Rate (FAR), higher Attack Detection Rate (ADR), and improved F-Measure on three different datasets: KDD'99, NSL-KDD, and CIC-IDS2017.

Ikram and Cherukuri [27] focused on the impact of feature reduction, demonstrating its impact through a comparative analysis. They used the NSL-KDD and the GURE-KDD datasets and applied PCA to select a subset of features and subsequently employed a SVM model as an anomaly detector. The feature reduction improved the detection accuracy from 0.9471 without PCA to 0.997 with PCA for NSL-KDD, while for GURE-KDD, it increased from 0.833 to 0.997.

Al-Yaseen [28] also focused on feature selection, demonstrating its effectiveness in improving overall performance. By employing the firefly algorithm (FA) for feature selection and SVM for classification on the NSL-KDD dataset, the accuracy achieved was 78.89%, significantly surpassing the 75.81% accuracy obtained using SVM alone.

Abdulraheem and Ibraheem [29] emphasized the significance of feature selection methods as well. They conducted a comparative analysis of their findings with those of Sharafaldin et al. [30], and evaluated their results on the CICIDS2017 dataset. By comparing the performance of a 36-feature dataset with a 23-feature dataset extracted by Sharafaldin et al. [30], and using the Random Forest algorithm, the results demonstrated that the 36-feature dataset yielded superior performance, achieving an accuracy of 0.992 compared to 0.988.

In a recent survey paper [31], Lata and Singh provided a coherent view of security concerns in each cloud service model. Additionally, they explored existing security techniques, highlighting their strengths and weaknesses, with a particular focus on state-of-the-art IDS and the importance of feature selection and dimensionality reduction.

Building upon this existing research as detailed in the above table, which has primarily focused on comparing classifier performance, the impact of applying feature selection to these classifiers, or the influence of selected feature quantity, our study aims to extend this investigation and provide a more comprehensive evaluation. To the best of our knowledge, this work is the first to examine the effectivness of combining various ML algorithms with different feature selection techniques, while considering a wide range of ML performance metrics for cloud environments.

III. Research Problem and Proposed Approach

This section presents the research objectives and provides an overview of our proposed approach.

TABLE I
SUMMARY TABLE OF LITERATURE REVIEW

Researcher	Feature Selection Techniques	Algorithms	Dataset
Feng et al. [20]	PCA	A new method called CVAC (combined the Support Vector Machine (SVM), and the Clustering based on Self-Organized Ant Colony Network (CSOACN)) (CSVAC)	DARP fxA'98
Ikram & Cherukuri [27]	PCA	Support Vector Machine (SVM)	NSL-KDD and GURE-KDD
Kevrick et al. [21]	All features	developed a framework for combining Random Tree (multiple decision trees), and NBTree (a hybrid of decision tree and Naïve Bayes classifiers), and C4.5 Decision Tree algorithms	NSL-KDD
Ahmim et al. [22]	-	Combination of multiple probability predictions tree of classifier (Naive Bayes (NB), Random Forest, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Radial-basis function network (RBFN), and Ripple Down Rule Learner (RIDOR).	KDD CUP 1999 and NSL-KDD
Azeez et al. [14]	-	Compare three different classifiers (Naïve Bayes, choice tree, and arbitrary timberland algorithm)	KDDCUP99
Nathiya & Suseendran [15]	-	Support Vector Machine (SVM), Naïve Bayes, and Decision Tree (J48).	NSL-KDD
Peng et al. [16]	-	Comparing three different classifiers: Naïve Bayes (NB), Decision Tree (DT), and K-Nearest Neighbors (k-NN) algorithms	KDDCUP99
Muttappanavar & Challagidad [23]	PCA	They present a hybrid machine learning technique-based intrusion detection system (Artificial Neural Network (ANN)) and (K-Means algorithms)	-
Aljamal et al. [24]	-	Using the K-means clustering algorithm to produce labels automatically and using SVM to build learning models.	UNSW-NB15
Al-Yassin [28]	Based on Firefly Algorithm	SVM	NSL-KDD
Sharma et al. [25]		Combine SVM, ELM (Extreme learning machine: single hidden layer feedforward neural network (Huang et al., 2006)), and K-means clustering algorithm.	KDD CUP 1999
Abdulraheem et al. [29] Sharafaldin et al. [30]	1. Feature Importance 2. Correlation.	Random Forest Algorithm	CICIDS2017
Kanimozhi & Jacob [17]	-	Six classification algorithms used such K-Nearest Neighbor, Naïve Bayes, Adaboost with Decision Tree, SVM, Random Forest, and Artificial Neural Network Classifier as MLP (Multi-layer perceptron)	CSE-CIC-IDS2018
Zhou et al. [26]	Correlation	Combined Random Forest (RF), C4.5, and Forest by Penalizing Attributes (Forest PA)	KDDCup'99 and NSL-KDD and CIC-IDS2017
Fitni & Ramli [18]	Correlation	Gaussian Naïve Bayes, Random Forests, Decision Trees, QDAs, MLPs, Logistic Regression, and Gradient Boosting.	CSE-CIC-IDS2018
Musleh et al. [13]	VGG-16 DenseNet	Random Forest, K-nearest neighbors, SVM, and different stacked models	IEEE Dataport
Rathod et al. [19]	-	Decision Trees, Neural Networks (NN), Support Vector machines (SVM), Random Forests, and k-nearest neighbors	KDDCup99 and NSL-KDD
This Study	Recursive Feature Elimination (RFE) and Feature Importance	Logistic Regression, Naïve Bayes Classifier (BernoulliNB), Decision Tree Classifier, Support Vector Machines, and Gradient Boosting Classifier	CSE-CIC-IDS2018

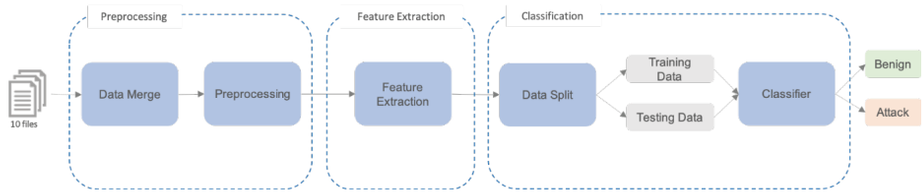

Fig. 1. Architecture of the proposed three-phase approach

Additionally, it delves into the implementation details and data specifications.

Our aim is to compare the performance of different machine learning algorithms using various feature selection techniques, focusing on optimizing computational efficiency while maintaining high accuracy. To achieve these objectives, we conducted experiments involving multiple methods to identify the most effective combinations for achieving both high accuracy and low processing time.

Fig. 1 illustrates the architecture of the proposed comparison and evaluation approach, which consists of three primary phases:

1. **Data Preprocessing:** Preparing the data for classification by removing insignificant features and addressing any incomplete, inconsistent, or inaccurate instances.
2. **Feature Selection:** Selecting the most relevant features from the dataset that are crucial for distinguishing abnormal traffic from normal traffic.
3. **Classification:** Comparing the performance of various classifiers in distinguishing abnormal traffic.

A representative dataset is essential for accurate machine learning model evaluation. The quality and diversity of the data directly influence the model's performance. In this research, we utilized the CSE-CIC-IDS2018 dataset, a collaborative effort between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC), designed to generate a comprehensive cybersecurity dataset in a systematic manner [32].

CSE-CIC-IDS2018 is a realistic cyber defense network dataset encompassing seven distinct attack scenarios: Brute-Force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and internal network infiltration. However, we concentrated on evaluating our model against three prevalent attack types: DoS attacks, DDoS attacks, and Brute Force attacks, as these attack types pose significant threats to cloud environments by disrupting their normal operations [33]. The CSE-CIC-IDS2018 dataset is accessible on AWS (Amazon Web Services) [34].

A. Data Pre-processing

Data preprocessing is a crucial step in preparing data for supervised classification, ensuring data cleanliness for subsequent phases. The CSE-CIC-IDS2018 dataset, employed in this research, is a real-world dataset collected over a period of time. As such, it may contain missing values, outliers, errors, noisy features, and inconsistencies in feature names or values. To address these issues, we inspected the dataset and implemented the following measures (using the Pandas and NumPy libraries in Python):

1. Feature Consistency: The CSE-CIC-IDS2018 dataset comprises 10 files. One of these files includes four additional attributes (Src IP, Src Port, Flow ID, and Dst IP) that are not present in the other files. To ensure consistency, we removed these features, resulting in a standardized dataset with 80 columns in each file.
2. Duplicate Rows: Some files contained duplicate rows, which were removed due to their relatively low frequency.
3. Feature Name Rows: Certain rows within some files included feature names, which were also removed.
4. Missing Values: Some rows contained missing values (NaN or infinity), which were removed due to their relatively low frequency too.

5. Class/Label Conversion: The label values were converted as follows: "BENIGN" to 0, and other attacks to numeric numbers.
6. Columns with Zero Values: Several columns contained exclusively zero values, indicating their irrelevance and lack of impact on the dataset. These features were removed.
7. Removal of Unnecessary Columns: The [Timestamp] feature was deemed unnecessary and removed.
8. Dataset Normalization: Various methods can be used for dataset normalization, such as Min-Max scaling or Standard scaling. We applied Standard scaling to all features except the 'Label' feature class.

The preprocessing measures applied to the CSE-CIC-IDS 2018 dataset reduced the number of columns from 80 to 69. These 69 columns were subsequently used in the feature selection phase.

In addition, given the substantial size of the CSE-CIC-IDS2018 dataset, which contains approximately 16 million records, working with the entire dataset can be computationally demanding, requiring specialized hardware and computing resources. To address this challenge, we employed data sampling, a common technique that involves selecting a representative subset of the original dataset. In our experiments, we focused on three of the ten files within the CSE-CIC-IDS2018 dataset. Various sampling methods exist, but we utilized simple random sampling. This involved generating random samples from each of the selected files and subsequently combining them into a single file.

The following t (TABLE I, TABLE II , and TABLEIII)

TABLE II
FILES USED FROM THE CSE-CIC-IDS2018 DATASET

Dataset file name	Columns	Rows	Traffic Type		Numbers of rows dropped		Numbers of columns ignored		Columns after	Rows after
Wednesday-14-02-2018_ TrafficForML_ CICFlowMeter	80	1048575	Benign	667626	Duplicate	225628	Containing zero value	10	69	819126
			FTP-BruteForce	193360	Null and Infinity values	3821	Unnecessary (Timestamp)	1		
			SSH-Brute force	187589	Rows containing attribute names	0				
Friday-16-02-2018_ TrafficForML_ CICFlowMeter	80	1048575	Benign	446772	Duplicate	147586	Containing zero value	10	69	900988
			DoS attacks-Hulk	461912	Null and Infinity values	0	Unnecessary (Timestamp)	1		
			DoS attacks-SlowHTTPTest	139890	Rows containing attribute names	1				
Wednesday-21-02-2018_ TrafficForML_ CICFlowMeter	80	1048575	Benign	360833	Duplicate	17557	Containing zero value	10	69	1031018
			DDOS attack-HOIC	686012	Null and Infinity values	0	Unnecessary (Timestamp)	1		
			DDOS attack-LOIC-UDP	1730	Rows containing attribute names	0				

TABLE III
DATA SAMPLING FILES

Dataset file name	Columns	Dataset after pre-processing			Data Sampling		
Wednesday-14-02-2018_TrafficForML_CICFlowMeter	69	819126	Benign	662458	204782	0	165501
			BruteForce	156668		1	39281
Friday-16-02-2018_TrafficForML_CICFlowMeter	69	900988	Benign	446653	90099	0	44706
			DoS	454335		2	45393
Wednesday-21-02-2018_TrafficForML_CICFlowMeter	69	1031018	Benign	360827	82481	0	28938
			DDOS	670191		3	53543

TABLE IV
DATA SAMPLE

Dataset file name	Columns	Total Rows	Label	Class	Rows for each class
Combined File	69	377362	Benign	0	239145
			BruteForce	1	39281
			DoS	2	45393
			DDoS	3	53543

present the details related to the implementation of the preprocessing phase on our dataset.

B. Features selection

Given the substantial number of features (80) in our dataset, which exhibit diverse characteristics, feature selection is a crucial step. By reducing the dimensionality of the data, we can significantly improve computational efficiency and potentially enhance model performance.

There are various feature selection methods, broadly categorized into unsupervised and supervised approaches. Unsupervised methods, such as correlation analysis, do not consider the target feature and focus on removing redundant variables. In contrast, supervised methods, including wrapper and filter approaches, utilize the target feature to identify and remove irrelevant variables. Wrapper methods select subsets of features based on their performance in a specific machine learning model, using metrics like accuracy or F1-score. Recursive Feature Elimination (RFE) is a well-known example of a wrapper method. Filter methods evaluate features based on their relationship with the target feature. This category includes statistical methods and feature importance methods. Statistical methods assess the correlation

```
['Dst Port',
 'Tot Fwd Pkts',
 'Tot Bwd Pkts',
 'TotLen Fwd Pkts',
 'Fwd Pkt Len Max',
 'Fwd Pkt Len Mean',
 'Flow Pkts/s',
 'Flow IAT Mean',
 'Fwd IAT Mean',
 'Fwd Header Len',
 'Bwd Header Len',
 'Fwd Pkts/s',
 'Bwd Pkts/s',
 'Pkt Len Max',
 'Pkt Len Mean',
 'Fwd Seg Size Avg',
 'Subflow Fwd Pkts',
 'Subflow Fwd Byts',
 'Subflow Bwd Pkts',
 'Subflow Bwd Byts',
 'Init Fwd Win Byts',
 'Fwd Act Data Pkts',
 'Fwd Seg Size Min']
```

Fig. 2: Feature subset using RFE.

or mutual information between features and the target, while feature importance methods evaluate the contribution of each feature to the model's performance.

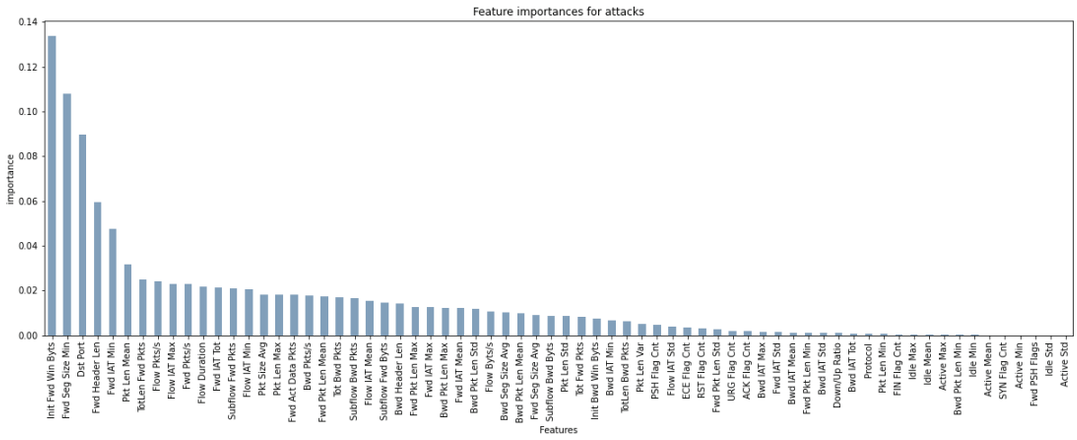

Fig. 3: Feature importance

In our experiments, we focused on supervised techniques and employed two well-established wrapper and filter methods to identify and select the most informative features: Recursive Feature Elimination (RFE) and feature importance. These methods were applied to extract two distinct subsets of features for each attack type. When using RFE, two key decisions need to be made: the number of features to select and the algorithm employed to guide the selection process [35].

- In the first subset (Fig. 2), we applied RFE using the Random Forest Classifier and selected 23 features. This process identified the most informative features for the three attack types, reducing the dataset's dimensionality from 69 features to 23 features.
- In the second subset (Fig. 4), we applied feature importance analysis using the Random Forest Classifier to identify the most relevant features, reducing the dimensionality of the original feature set while preserving essential information. By examining the histogram of feature importance scores depicted in Fig. 3, we established a threshold of 0.0010 and removed features with scores below this threshold, resulting in the elimination of 17 features from the original set of 69. Subsequently, we applied RFE using the Decision Tree Regressor to further refine the feature set, extracting 23 features from the remaining 52 features.

```
['Dst Port',
 'Flow Duration',
 'TotLen Bwd Pkts',
 'Fwd Pkt Len Max',
 'Fwd Pkt Len Std',
 'Bwd Pkt Len Max',
 'Bwd Pkt Len Std',
 'Flow Byts/s',
 'Flow Pkts/s',
 'Flow IAT Mean',
 'Flow IAT Max',
 'Flow IAT Min',
 'Fwd IAT Tot',
 'Fwd IAT Std',
 'Fwd IAT Min',
 'Fwd Header Len',
 'Bwd Header Len',
 'Fwd Pkts/s',
 'Bwd Pkts/s',
 'URG Flag Cnt',
 'Init Fwd Win Byts',
 'Fwd Act Data Pkts',
 'Fwd Seg Size Min']
```

Fig. 4: Feature subset using Feature Importance Analysis

C. Classification

To achieve the objective of our IDS in cloud environments, the system must be trained using a suitable machine learning algorithm. Given our aim to obtain results for each specific attack type, we focus on algorithms capable of multi-class classification. Many algorithms originally designed for binary classification can be adapted for multi-class classification too, including Decision Trees, Random Forest, Naïve Bayes, Gradient Boosting, k-Nearest Neighbors, Support Vector Machines, and Logistic Regression. However, some algorithms, such as the AdaBoost Classifier, are

TABLE II
ACCURACY & TRAINING TIME FOR EACH CLASSIFIER

Attack type	Classifier Type	First subset of features		Second subset of features	
		Accuracy	Training Time (Seconds[1])	Accuracy	Training Time (Seconds)
CSE-CIC-IDS2018 dataset (Brute Force, DoS, DDoS attack)	Naive Baye Classifier	0.958	3.318	0.974	3.446
	Decision Tree Classifier	0.995	18.002	0.995	20.305
	Logistic Regression	0.987	179.374	0.990	162.646
	Support Vector Machines	0.993	4320.284	0.993	3530.986
	Gradient Boosting Classifier	0.995	8082.679	0.995	24594.344

inherently limited to binary classification. In addition, each algorithm possesses distinct advantages and disadvantages. For example, decision trees are often effective for tasks requiring speed, while SVM and gradient boosting excel in achieving high accuracy [10] [11].

In our experiments, we employed five classification models, namely, Logistic Regression, Naïve Bayes Classifier (BernoulliNB), Decision Tree Classifier, Support Vector Machines, and Gradient Boosting Classifier. Considering our proposed approach (Fig. 1), the dataset was randomly partitioned into training and testing subsets to initiate the classification phase, allocating 70% for training and 30% for testing. Each classifier was then trained independently using the training subset. The resulting models were subsequently validated on the testing subset to assess their training effectiveness and estimate model properties, including accuracy and training time. The output results were categorized as follows: (0) normal, (1) Brute Force attack, (2) DoS attack, and (3) DDoS attack. TABLE II presents the training time and accuracy for each classifier.

IV. Experimental Results and Observations

This section discusses the experimental results in detail, highlighting their alignment with the study's objectives. We describe the performance metrics obtained from the primary experiments outlined in section (III). The main objective of these experiments is to validate the intrusion detection model using the CSE-CIC-IDS2018 dataset.

Our proof-of-concept model was designed as a flexible and adaptable framework, allowing for easy modifications and future enhancements while ensuring efficient performance. The implementation was carried out using Python, incorporating libraries such as Scikit-learn, Pandas, NumPy, and Matplotlib. Experiments were conducted on Python3, using Anaconda3 and Jupyter Notebook, on a macOS system powered by an Intel Core m7 processor (1.3 GHz) with 8 GB of RAM.

As detailed in the previous section, the classification algorithms began by randomly partitioning the dataset into training and testing subsets, allocating 70% for training and 30% for testing. Each classifier was then trained independently using the training subset. The resulting models were validated on the testing subset to evaluate their effectiveness and estimate key performance metrics.

To compare the performance of five classifiers, we conducted experiments using two distinct feature sets. These experiments included 138,217 attack records and 239,145 benign records, randomly selected from the dataset (Table III). The first experiment utilized the initial subset of features (Fig. 2), while the second experiment employed the refined subset of features (Fig. 4), both of which were derived through the feature extraction methods outlined in Section III.

Table V presents the accuracy results for both subsets of features alongside the testing times for

(1) The "time" module in Python is used to measure execution time. This module provides functions to obtain time in seconds, with a resolution of approximately one millisecond.

TABLE VI
TESTING DATA RESULTS, ACCURACY AND TIME FOR EACH CLASSIFIER

Attack type	Classifier Type	First subset of features		Second subset of features	
		Accuracy	Testing Time (Seconds)	Accuracy	Testing Time (Seconds)
CSE-CIC-IDS2018 dataset the three types of) (attack	Decision Tree Classifier	0.995	0.385	0.995	0.349
	Gradient Boosting Classifier	0.995	1.668	0.995	1.292
	Support Vector Machines	0.994	93.528	0.994	79.605
	Logistic Regression	0.988	0.372	0.990	0.338
	Naive Bayes Classifier	0.959	0.357	0.974	0.349

TABLE VII
THE FILE USED FOR BRUTE FORCE ATTACKS

Dataset file name	Columns	Rows	Traffic Type		Numbers of rows dropped		Numbers of columns ignored		Columns after	Rows after
Wednesday-14-02-2018_TrafficForML_CICFlowMeter	80	1048575	Benign	667626	duplicate	225628	Containing zero value	10	69	819126
			FTP-Brute-Force	193360	Null and Infinity values	3821	Unnecessary ((Timestamp	1		
			SSH-Brute-force	187589	Rows containing attribute names	0				

each classifier. The results reveal that the second feature extraction method, which combines a filter technique (feature importance) with a wrapper technique (Recursive Feature Elimination, RFE), demonstrates superior performance in certain classification algorithms, such as the Naïve Bayes Classifier and Logistic Regression. Conversely, other classification algorithms, including Decision Tree Classifier, Gradient Boosting Classifier, and Support Vector Machines, show comparable accuracy across both feature selection methods.

Furthermore, the second feature selection method consistently achieves lower processing times across all five classifiers. This improvement in performance can be attributed to the enhanced contribution of the features selected by the second approach to the classification process. While these findings support the study's objectives, further investigation is required to fully understand the underlying factors driving this improvement.

To further validate the efficacy of the classifiers and feature selection techniques, we conducted additional experiments focusing on a comparative analysis of their performance in detecting Brute Force attacks. TABLE VI below provides details about the data file used in these experiments.

The performance metrics considered in these experiments include training and testing times, model accuracy, recall (TPR), false negative rate (FNR), and false positive rate (FPR). Recall (TPR) measures the proportion of attacks correctly identified by the IDS, while false negative rate (FNR) represents the proportion of missed attacks. False positive rate (FPR) indicates the proportion of normal traffic incorrectly classified as attacks by the IDS.

TABLE VIII presents a comparative analysis of the performance metrics for the five classifiers

TABLE VIII
COMPARISON OF CLASSIFIER PERFORMANCE USING THE FIRST FEATURE SELECTION METHOD

Classifier	Training Time (Seconds)	Testing Time (Seconds)	Accuracy	Recall/TPR	FNR	FPR
Logistic Regression	468.314	0.974	0.999153	0.9999	0.0000	0.0010
Naïve Baye Classifier	6.404	0.986	0.845107	0.8303	0.1696	0.1514
Decision Tree Classifier	15.535	0.763	1.000000	1.0000	0.0000	0.0000
Support Vector Machines	1083.514	57.44	0.999853	0.9998	0.0001	0.0001
Gradient Boosting Classifier	7979.241	2.501	1.000000	1.0000	0.0000	0.0000

TABLE IX
COMPARISON OF CLASSIFIER PERFORMANCE USING THE SECOND FEATURE SELECTION METHOD

Classifier	Training Time (Seconds)	Testing Time (Seconds)	Accuracy	Recall/TPR	FNR	FPR
Logistic Regression	495.185	0.842	0.999674	0.9999	0.0000	0.0003
Naïve Baye Classifier	6.376	0.907	0.945873	0.9981	0.0018	0.0664
Decision Tree Classifier	18.223	1.255	1.000000	1.0000	0.0000	0.0000
Support Vector Machines	427.993	18.951	0.999963	0.9999	0.0000	0.0000
Gradient Boosting Classifier	41020.03	2.424	1.000000	1.0000	0.0000	0.0000

using the first feature subset (Fig. 2). TABLE IX provides a similar comparison for the classifiers using the second feature subset (Fig. 4).

An analysis of the values in TABLE VIII (first feature set) and TABLE IX (second feature set) further substantiates our findings that the second feature selection approach consistently yields improved performance. comparing the metric values presented in TABLE VIII and TABLE IX, we observe the following:

- Naïve Bayes Classifier demonstrated notable improvements in performance using the second feature set. Accuracy increased from 0.84 to 0.94, and sensitivity (TPR) rose from 0.83 to 0.99. Concurrently, the false negative rate (FNR) and false positive rate (FPR) decreased, enhancing the overall efficiency and effectiveness of the IDS.
- Support Vector Machines also exhibited slight improvements in accuracy and significant reductions in processing time when employing the second feature set. Moreover, there is a decrease in false negative rate (FNR) and false positive rate (FPR) which further enhances the overall effectiveness of the IDS.
- Both Decision Tree Classifier and Gradient Boosting Classifier demonstrated comparable performance in both feature selection methods, consistently achieving the highest accuracy. However, the second feature selection method resulted in considerable increase in processing times for both classifiers.

Additionally, we can observe the following key findings

- Decision Tree Classifier and Gradient Boosting Classifier consistently demonstrate the highest accuracy. However, Decision Tree Classifier offers a distinct advantage in terms of speed, making it a more computationally efficient option. While Gradient Boosting Clas-

sifier exhibits superior accuracy, its computational expense, as evidenced by the longest training time, should be carefully considered.

- Although Naïve Bayes Classifier is the fastest in terms of training time, it exhibits the lowest performance overall. Moreover, its relatively high false negative rate (FNR) diminishes its effectiveness in an intrusion detection system.

V. Conclusion and Future Work

Within the context of the proposed architecture for an intrusion detection system in cloud environments, this paper has provided a comprehensive evaluation of five classification algorithms and two feature selection methods. Using the CSE-CIC-IDS2018 dataset, we extracted a subset of data focusing on attack types that pose significant threats to cloud security, including Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), and Brute-Force attacks.

Our experimental results demonstrate that feature selection effectively reduces dataset size and processing time while maintaining high detection performance—key factors for real-time attack detection in cloud environments. This study confirms that feature selection and engineering are critical components of any machine learning model, significantly influencing both accuracy and efficiency. Continued research and experimentation with various feature selection and extraction techniques are recommended to further enhance the performance of machine learning models in this domain.

The experiments also revealed that the Decision Tree Classifier and Gradient Boosting Classifier exhibit superior accuracy in detecting attacks. However, these classifiers demand significantly more processing time compared to others when utilizing the feature selection methods employed in this study. We recognize that each classification technique has unique strengths and limitations, and its performance can vary depending on the specific attack types being analyzed.

To address the dual objectives of achieving high detection accuracy and enabling real-time analysis, combining multiple classifiers presents a promising avenue for future research. Additionally, investigating the application of deep learning algorithms and leveraging recent advancements in machine learning and artificial intelligence offer further potential for enhancing both classification techniques and feature extraction methods.

Conflict of Interest

Authors declare that they have no conflict of interest.

Funding

This article did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *National institute of science and technology, special publication*, 2011.

[2] Triskele Labs, "Cloud cyber attacks: The latest cloud computing security issues," September 2024. [Online]. Available: https://www.triskelelabs.com/blog/cloud-cyber-attacks-the-latest-cloud-computing-security-issues.

[3] M. Henriquez, "The top 10 data breaches of 2020," 3 December 2020. [Online]. Available: https://www.securitymagazine.com/articles/94076-the-top-10-data-breaches-of-2020.

[4] CrowdStrike, "CrowdStrike 2024 Global Threat Report," 2024. [Online]. Available: https://www.crowdstrike.com/global-threat-report/.

[5] A. A. Shaikh, "Attacks on cloud computing and its countermeasures," in *2016 International conference on signal processing, communication, power and embedded system (SCOPES)*, 2016.

[6] P. Mishra, E. S. Pilli, V. Varadhara and U. Tupakula, "Efficient approaches for intrusion detection in cloud environment," in *2016 international conference on computing, communication and automation (ICCCA)*, 2016.

[7] M. M. Sakr, M. A. Tawfeeq and A. B. El-Sisi, "Network intrusion detection system based PSO-SVM for cloud

computing," *International Journal of Computer Network and Information Security*, vol. 14, 2019.

[8] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of network and computer applications*, vol. 36, pp. 42-57, 2013.

[9] N. Keegan, S.-Y. Ji, A. Chaudhary, C. Concolato, B. Yu and D. H. Jeong, "A survey of cloud-based network intrusion detection analysis," *Human-centric Computing and Information Sciences*, vol. 6, pp. 1-16, 2016.

[10] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373-440, 2020.

[11] J. Han, J. Pei and H. Tong, Data mining: concepts and techniques, Morgan kaufmann, 2022.

[12] S. S. S. Sindhu, S. Geetha and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Systems with applications*, pp. 129-141, 2012.

[13] D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman and R. M. Mohammad, "Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, 2023.

[14] N. A. Azeez, O. J. Asuzu, . S. Misra, A. Adewumi, R. Ahuja and R. Maskeliunas, "Comparative Evaluation of Machine Learning Algorithms for Network Intrusion Detection Using Weka," in *Towards Extensible and Adaptable Methods in Computing*, Springer, 2018, p. 195–208.

[15] T. Nathiya and G. Suseendran, "An effective way of cloud intrusion detection system using decision tree, support vector machine and Na{\"\i}ve bayes algorithm," *International Journal of Recent Technology and Engineering*, vol. 7, pp. 38-42, 2018.

[16] K. Peng, V. C. Leung, L. Zheng, S. Wang, C. Huang and T. Lin, "Intrusion detection system based on decision tree over big data in fog environment," *Wireless Communications and Mobile Computing*, 2018.

[17] V. Kanimozhi and T. P. Jacob, "Artificial Intelligence outflanks all other machine learning classifiers in Network Intrusion Detection System on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," *ICT Express*, vol. 7, no. 3, pp. 366-370, 2021.

[18] Q. R. S. Fitni and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems," in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 2020.

[19] G. Rathod, V. Sabnis and J. K. Jain, "Intrusion Detection System (IDS) in Cloud Computing using Machine Learning Algorithms: A Comparative Study," *Grenze International Journal of Engineering and Technology (GIJET)*, vol. 10, no. 1, 2024.

[20] W. Feng, Q. Zhang, G. Hu and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127-140, 2014.

[21] J. Kevric, S. Jukic and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, vol. 28, pp. 1051-1058, 2017.

[22] A. Ahmim, M. Derdour and M. A. Ferrag, "An intrusion detection system based on combining probability predictions of a tree of classifiers," *International Journal of Communication Systems*, vol. 31, 2018.

[23] L. Muttappanavar and P. S. Challagidad, "Intrusion Detection On Cloud Using Hybrid Machine Learning Techniques," International Journal of Computer Engineering and Applications, vol. XII, 2018.

[24] I. Aljamal, A. Tekeoğlu, K. Bekiroglu and S. Sengupta, "Hybrid intrusion detection system using machine learning techniques in cloud computing environments," in *2019 IEEE 17th international conference on software engineering research, management and applications (SERA)*, 2019.

[25] J. Sharma, C. Giri, O.-C. Granmo and M. Goodwin, "Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation," *EURASIP Journal on Information Security*, pp. 1-16, 2019.

[26] Y. Zhou, G. Cheng, S. Jiang and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer networks*, vol. 174, 2020.

[27] S. T. Ikram and A. K. Cherukuri, "Improving accuracy of intrusion detection model using PCA and optimized SVM," *Journal of computing and information technology*, vol. 24, no. 2, pp. 133-148, 2016.

[28] W. L. Al-Yaseen, "Improving intrusion detection system by developing feature selection model based on firefly algorithm and support vector machine," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 534-540, 2019.

[29] M. H. Abdulraheem and N. B. Ibraheem, "A detailed analysis of new intrusion detection dataset," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 17, 2019.

[30] I. Sharafaldin, A. Gharib, A. H. Lashkari and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," *Software Networking*, pp. 177-200, 2018.

[31] S. Lata and D. Singh, "Intrusion detection system in cloud environment: Literature survey & future research directions," *International Journal of Information Management Data Insights*, vol. 2, no. 2, 2022.

[32] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108-116, 2018.

[33] J. Brownlee, "Types of classification tasks in machine learning," *Machine Learning Mastery*, vol. 1, 2020.

[34] "A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)," [Online]. Available: https://registry.opendata.aws/cse-cic-ids2018. [Accessed July 2024].

[35] J. Brownlee, "How to choose a feature selection method for machine learning," *Machine Learning Mastery*, vol. 10, pp. 1-7, 2019.