Kufa Journal of Engineering
Vol. 16, No. 2, April 2025, P.P. 344 -366
Article history: Received 16 July 2024, last revised 25 November 2024, accepted 25 November 2024

ENHANCING DDOS ATTACK CLASSIFICATION THROUGH SDN AND MACHINE LEARNING: A FEATURE RANKING ANALYSIS

Kawthar Rasoul ALesawi [1] and Aymen Hasan Alawadi [2]

[1] Department of Computer Science, Faculty of Education, University of Kufa, Iraq - Najaf , kawtharr.alesawi@student.uokufa.edu.iq

[2] Department of Computer Science, Faculty of Education, University of Kufa, Iraq – Najaf, aymen@uokufa.edu.iq

https://doi.org/10.30572/2018/KJE/160221

ABSTRACT

Due to the growing dependence of digital services on the Internet, Distributed Denial of Service (DDoS) attacks are a common threat that can cause significant disruptions to online operations and financial losses. Machine learning (ML) offers a promising way for early DDoS attack detection due to its ability to analyze large datasets and identify patterns. However, adding too many features to the ML might reduce its effectiveness in identifying the attacks provided by central network paradigms such as the Software-Defined Network (SDN). In this research, we investigate the effectiveness of the ML methods such as (Random Forest (RF), Naive Base (NB), and K-Nearest Neighbor's (KNN)) combining SDN to enhance the classification of DDoS attacks. We leverage three diverse datasets: DDoS attack SDN, CICDDoS2019, and SDN-DDOS-TCP-SYN dataset. By leveraging cross-feature selection and feature ranking techniques, such as information gain, gain ratio, and Gini importance, we could identify the most relevant network features for DDoS attacks. We reduced the feature up to 5 effective features without compromising the classification accuracy. The experimental results show that the proposed models achieved an accuracy of 100% for both Random Forest (RF) and K-Nearest Neighbor (KNN), and 99.8% for Naive Bayes (NB). Due to their high accuracy and lower complexity, KNN and NB outperform ML algorithms in this study..

KEYWORDS

SDN; DDOS; Machine Learning; Classification; Feature Ranking.

1. INTRODUCTION

Today's networks are more exposed to a range of cyber-attacks due to the expansion of online services and growing reliance on the Internet. DDoS attacks are among the most prevalent and disruptive online attacks. Such an assault can flood a target system, service, or network with traffic, killing it so that authorized users cannot access it.

In the networking sector, SDN has grown significantly. By enabling centralized network management and device programming, SDN technology enhances network performance and administration. By separating the data plane from the control plane, an SDN controller can configure the control plane of a network device. Three levels make up SDN architecture: the control layer, which implements the controller, the application layer, which implements networking applications, and the data plane layer, which houses networking devices like switches and hosts. The open-flow protocol facilitates secure communication between the controller and switches. Upon arrival of a new packet at the switch, it searches its tables for a match. If found, the packet follows the table's instructions for forwarding; otherwise, it's forwarded to the controller for processing. In the case of a DDoS attack with spoofed addresses, packets are sent to the controller, potentially overwhelming it, and causing it to become unreachable to legitimate traffic, thus disrupting the SDN architecture. SDN presents an essential management framework for high-demand network applications, such as Worldwide Interoperability for Microwave Access (WiMAX) (Laassiri, Moughit and Idboufker, 2018). WiMAX real-time applications, such as video streaming requires low latency and high throughput (Barznji and Ameen, 2021). Such principles apply to SDN-enabled networks, where maintaining Quality of Service (QoS) during an active DDoS attack necessitates a dynamic and responsive network design.

ML-based methodologies utilize various techniques to classify anomalies within a networked environment. These models can be founded on statistical and mathematical principles, as well as on both supervised and unsupervised algorithms. Such algorithms take into account various network features and traffic characteristics to detect the presence of anomalies. Any system that is built to detect any anomalies in the network catches the traffic on it and extracts some kind of information from them, and the approach that uses ML models is trying to catch the pattern of normal and abnormal traffic on the network, without the need to know the pattern itself (Santos et al., 2020). Collecting a wider range of network features might enhance anomaly detection. In the SDN environment, however, it can also lead to more complex models that are harder to interpret and manage. Performance optimization and understanding each application's specific demands can enable network designers to make strategic choices that improve overall

network efficiency. This approach ensures that network resources are efficiently utilized, minimizing latency and maximizing throughput, essential for applications requiring consistent, high-speed data transmission (O. Hasan, 2022). In this paper, we propose the following contributions:

1- Maintaining high accuracy in detecting DDoS attacks using ML models in SDN environment.

2- A feature reduction metho named cross-feature selection using several feature ranking methods.

3- Examining the complexity of the used ML detection algorithms on various related DDoS datasets and identifying less complex ones in terms of training and testing.

2. RELATED WORK

In previous publications, ML algorithms have been used to identify DDoS activities. (Hosseini and Azizi, 2019) introduced the client side containing three steps. The first step involved the data collection of the client system, followed by feature extraction based on forward feature selection for each algorithm, and finally, the divergence test. Consequently, if the divergence exceeds a threshold, the attack will be detected; otherwise, the data will be processed to the proxy side. On the proxy side, they employed NB, RF, DT, MLP, and K-NN to achieve better results. Different attacks exhibited specific behaviors, and due to the selection of different features for each algorithm, the system demonstrated improved performance in detecting attacks and a greater ability to distinguish new attack types. The paper lacks evaluation using real-world datasets. The authors have only used two synthetic datasets for the evaluation, one containing DDoS attack traffic mixed with normal traffic, and another with more modern attack types. However, the paper does not mention the size or representativeness of these datasets, raising concerns about the generalizability of the results.

(Polat and Polat, 2020) utilized an existing dataset to create a new dataset employing feature selection methods, as presented. This study investigated the simplicity of feature selection approaches and how they are easy to interpret and train. ML models such as SVM, NB, ANN, and KNN, were employed in the classification. The KNN classifiers and wrapper features detected DDoS attacks with high accuracy (98.3%). However, this study did not provide sufficient information about the origin or source of the dataset, which questions its validity and generalizability.

(Dong and Sarem, 2020) introduced new DDoS detection algorithms: DDADA (DDoS Detection Algorithm based on the Degree of Attack) and DAMDL (DDoS Detection Algorithm

based on ML) to effectively identify DDoS attacks in SDN environments. Algorithms with selected DDoS features like flow length, flow duration, flow size, and attack degree, have been proposed to enhance detection effectiveness in ML-based DDoS classification. In the ROC analysis, DDAML demonstrated promising performance with superb accuracy, and its AUC value was approximately 91%. However, this study has some limitations related to the datasets used. (Tonkal et al., 2021) presented an SDN method based on ML capabilities to identify DDoS attacks. The approach focused on classification and found the SDN traffic as either normal or indicative of an attack using ML algorithms enhanced with Neighborhood Component Analysis (NCA). The experimental results indicated that the DT algorithm outperformed others, achieving a 100% classification accuracy rate. However, to evaluate their work, they used just one dataset in the experimental results.

(Nadeem et al., 2022) proposed a DDoS detection model based on SDN environment, showing that the RF classifier achieves 99.97% accuracy. The paper highlighted the importance of optimal feature selection for accurate classification and SDN controller performance. However, it lacks detailed dataset information, essential for understanding the generalizability and real-world applicability of the proposed approach.

(Mansoor et al., 2023) proposed another SDN approach using the Deep Learning (DL). The proposed approach evaluated using standard evaluation metrics, including false positive rate and detection accuracy, with the help of a benchmark dataset. According to the findings, the recommended approach detects DDoS attacks with average detection accuracy, average precision, average FPR, and average F1-measure of 94.186%, 92.146%, 8.114%, and 94.276%, respectively.

(Ali, Chong and Manickam, 2023) investigated the effectiveness of DL and other ML algorithms in identifying and mitigating anomalies, particularly DDoS attacks. Several classification methods were compared, including Support Vector Machines (SVM), K-Nearest Neighbors (KNNs), Decision Trees (DTs), Multi-Layer Perceptron (MLPs), and Convolutional Neural Networks (CNNs). The CNN model achieved a training accuracy of 97.808%, but its prediction accuracy was only 90.08%. On the other hand, the SVM model exhibited the highest prediction accuracy of 95.5%.

(Najar and Manohar Naik, 2024) introduced an SDN-based model combined with DL to detect and mitigate DDoS attacks. The detection model was designed using Balanced Random Sampling (BRS) and CNNs. The mitigation techniques included filtering, rate-limiting, and iptables rules to block spoofed IP addresses. They also set up a rate-limiting monitoring system for blocked IP addresses to make sure that legitimate traffic is served promptly. The proposed

model achieved 99.99% accuracy in binary classification and 98.64% in multiclassification. This is in contrast to a method with the problem of a huge number of considered features: 58 features, which need more computational resources to process and analyze a larger set of data. Moreover, the larger number of features may require more sophisticated techniques for feature selection and dimensionality reduction to prevent overfitting and improve the model's generalizability.

This research (Han et al., 2024) presented a new feature selection methodology, designated as XRDI, to enhance Intrusion Detection Systems (IDS) by filtering out unnecessary features to improve detection speed and accuracy. The methodology was implemented across several ML algorithms such as Decision Trees (DT), RF, Support Vector Machines (SVM), and Logistic Regression (LR) using benchmark datasets (InSDN, CICIDS2017, CICIDS2018). The method was tested on an SDN simulation platform. The results showed that XRDI significantly reduced detection time while maintaining high accuracy, meeting modern network demands for speed and reliability. The study shows that more research needs to be done on how well XRDI can handle new attacks in different network environments.

Our research focuses on classifying DDoS attacks using ML and SDN paradigm. Our motivation in this study has been to work on datasets obtained from SDN network platforms. by using three datasets in our study. We use the KNN, RF, and NB algorithms to detect the attacks due to their accurate classification and less complexity. This research explores the integration of ML techniques and SDN to enhance the detection capabilities DDoS attacks. To improve the performance of our detection system, we implement cross-feature selection and feature ranking algorithms aimed at reducing features while maintaining or even enhancing classification accuracy. Feature reduction is crucial in DDoS attack detection, especially in SDN environments and real-time processing demands. By leveraging the ranking algorithm, we were able to identify the most informative features, thereby streamlining the input data fed into our ML algorithms.

3. METHODOLOGY

To identify DDoS attacks, we use various ML-related techniques. The steps taken to achieve the results are illustrated into a flowchart Fig. 1. Below is a simplified flowchart demonstrating the process of detecting DDoS attacks using the evaluated datasets:

- Starting by obtaining the dataset containing network traffic information.
- The feature ranking algorithm is the next step used to determine the most relevant features that will highly impact the classification accuracy for the utilized datasets.

- The next step involves the cross-feature selection in which we highlight the intersected sets of features resulting from ranked features.
- Selecting an ensemble of ML algorithms tailored to classify this dataset effectively and derive results.
- Ending the flowchart, by choosing the best-trained method with the selected features.

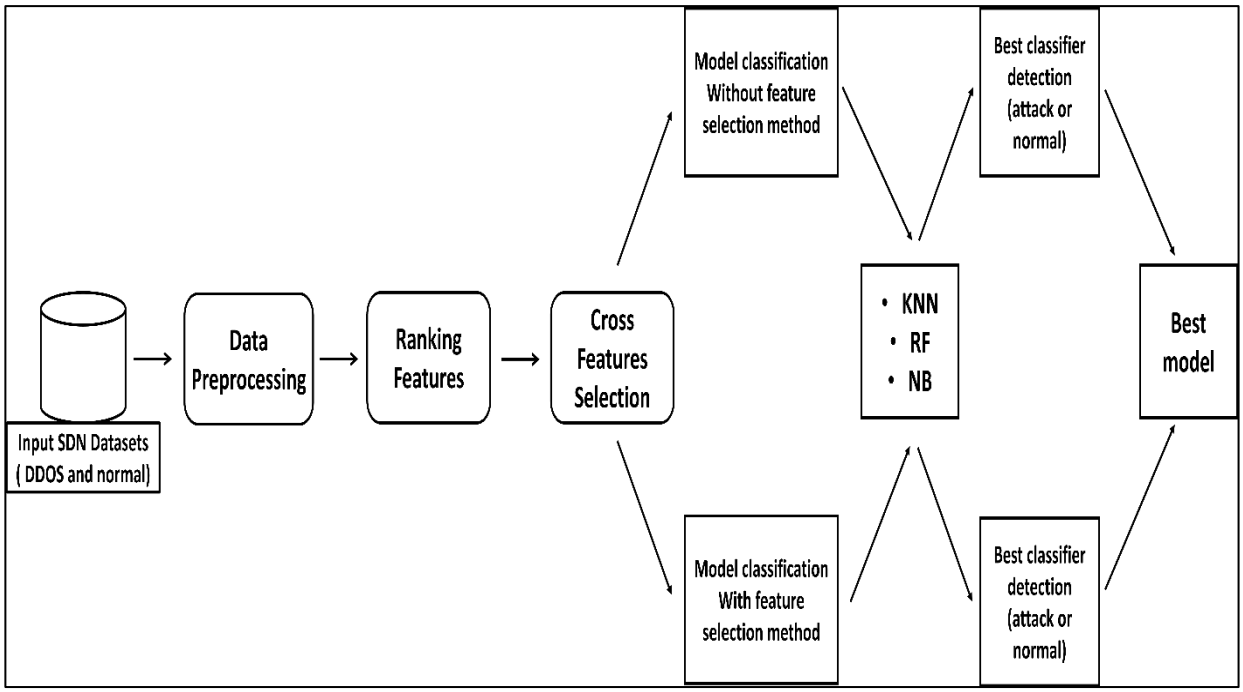

Fig. 1. The steps of the proposed model.

The features and classes of the public datasets used in this study will be discussed in the following sections:

1. Datasets: we utilized the "DDoS attack SDN dataset" comprising 23 features and 104,345 instances (Nisha Ahuja Singal, Gaurav Mukhopadhyay, 2020). This dataset encompasses both normal and attack instances of Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP) traffic. Within the dataset's 23 features, some are derived from switches, while others are computed. For more reliable results, we further utilized the "CICDDoS2019 Dataset "The Canadian Institute for Cybersecurity developed a DDoS attack dataset in 2019 (Talukder, Md Alamin; Uddin, 2023). CICDDoS2019 delivers regular and up-to-date common DDoS attacks close to real-world data PCAPs. It also provides the findings of a network monitoring performed using CICFlowMeter-V3 that extracted 80 features apart from labeled flows. Additionally, we leveraged another dataset, the "SDN-DDOS-TCP-SYN DATASET" created by Gupta et al. (2021) (Gupta, Karan; Sharma, Shivam; Kumar, 2021), which consists of 28 columns and 10,47,500 rows and captures the network traffic during a TCP-SYN Flood attack scenario.

2. Data Preprocessing: this is a crucial step in the data analysis process. It involves preparing raw data and transforming it into a format suitable for analytical models and ML algorithms. This process includes several key steps:

- **Data Cleaning:** This process involves identifying and removing errors, inconsistencies, or irrelevant data from the datasets, such as duplicate data, standardizing values, and handling outliers, to ensure that the data used in the model is accurate and reliable. This process was used to examine the dataset, and it was confirmed that it was free of these issues.
- **Handling Missing Values:** Missing data can affect the accuracy of the model. Therefore, it is essential to deal with it properly.
- **Dimensionality Reduction:** It is a technique utilized in data analysis and ML to reduce the number of input features or variables in a dataset while retaining as much important information as possible. This was performed on the datasets via a reduction of the number of features (cross-feature selection). The objective was to achieve high-speed execution, high accuracy, and low computational complexity on target datasets.
- **Data balancing:** Regarding the utilized datasets, the "DDoS attack SDN dataset" exhibited a balanced feature distribution, with 63,561 instances of normal data and 40,784 instances of attacks. The "CICDDoS2019 dataset" contains 97,831 instances of normal data and 333,540 instances of labeled attacks. The "sdn-ddos-tcp-syn-dataset" was the most extensive, with 1,421,835 instances of normal data and 115,115 instances of attacks.

These steps enhance data quality and increase the accuracy of analytical and predictive models.

3. Ranking of features: Feature ranking algorithms play a crucial role in ML workflows by identifying the most informative features within a dataset. The impact of a ranking algorithm on a dataset depends on the specific algorithm being used and the context of the dataset where feature ranking algorithms contribute significantly to the success of ML tasks by improving model performance, reducing computational complexity, and enhancing interpretability. Below is a brief discussion of the used features' ranking algorithms:

- **Gini Importance (GI):** (also known as Gini Decrease in Impurity) the GI method used mainly to rank features based on their importance in classification. This importance was determined by calculating how much each feature reduced the Gini impurity when used to split the data in a decision tree. Features that lead to a greater impurity reduction are ranked higher because they provide more valuable information for distinguishing between classes. The GI is calculated as follows in equation (1) (Orange Data Mining, 2024):

$$Gini = 1 - \sum_{i=1}^{2} P_k^2 \quad (1)$$

Where, P_k represents the probability of a sample belonging to class k within a node, show in equation 2:

$$P_k = \frac{\text{Number of samples in class } k}{\text{Total number of samples}} \quad (2)$$

- **Information gain (IG):** is an ML technique that evaluates feature relevance by measuring the reduction in entropy when a dataset is split based on a feature (Xing, Jordan and Karp, 2001). The IG for a given attribute *A* concerning a dataset *D* is calculated as follows in equation (3) (Orange Data Mining, 2024):

$$IG(D, A) = H(D) - H(D/A) \quad (3)$$

Where:

- H(D/A) is the conditional entropy of D given A.

- H(D) is the entropy of the dataset D.

Entropy *H(D)* is the measures of impurity, disorder or uncertainty in a bunch of examples. If we have a set of *i* different values , then we can calculate the entropy using this formula (4) (Orange Data Mining, 2024):

$$H(D) = -\sum_{j=1}^{2} p_j \, log_2(pj) \quad (4)$$

where, *pj* is the probability of getting the *j* value when randomly selecting one from the set, show in equation 5:

$$pj = \frac{\text{Number of samples in class } j}{\text{Total number of samples in the dataset}} \quad (5)$$

- **Information Gain Ratio (IGR):** Information Gain indicates a particular attribute's ability to reduce uncertainty or entropy when used to split the dataset. An attribute with a higher information gain is considered more valuable for decision-making in the construction of a decision tree and signifies greater relevance for task differentiation. Information gain ratio ranking refines this method by normalizing information gain to account for potential bias towards features with many values. The higher the information gain, the better the discrimination between classes in the dataset. Predictions can be made more accurately by separating data into more homogeneous subsets. The formula for Information Gain is in equation (6) (Orange Data Mining, 2024):

$$IG(D, A) = H(D) - H_A(D) \quad (6)$$

Generally, $H_A(D)$ has to be as small as possible to maximize IG. This means that the subsets created by splitting the dataset based on the values of attribute A should be as pure as possible (i.e., containing instances of a single class).

4. Feature selection: Selecting the right features is critical to the performance of the ML algorithms. ML-based feature selection provides the following advantages for DDoS detection in an SDN environment:

- **Enhanced accuracy**: Feature selection can lower noise and redundancy in the data by concentrating on the most pertinent and instructive elements, resulting in more accurate ML algorithms.
- **Decreased computational complexity:** ML algorithms become more effective and appropriate for real-time SDN contexts when dealing with fewer features, leading to faster training, and testing times. This is essential to reducing the effect of DDoS attacks before they inflict substantial harm.

5. ML algorithms: Detecting and mitigating DDoS attacks in SDN environments involves leveraging ML techniques to identify abnormal network behavior and applying intelligent mitigation strategies. Choosing more than one ML method is commonly used to detect DDoS attacks for several reasons including robustness, redundancy, adaptability, and false positive reduction.

Overall, using multiple ML techniques for DDoS detection improves the defense's overall system's effectiveness, adaptability, and resilience. However, below are the utilized approaches in our paper:

- **K-Nearest Neighbor (KNN):** the KNN algorithm is a simple supervised ML algorithm that uses the concept of "feature similarity" to classify a given data sample. By determining a sample's identity based on its neighbors and how far away it is from them, KNN can effectively determine the class of a data sample. The value of the KNN algorithm's *k* parameter can have an impact on its performance, and selecting a *k* value that is too small or too large can lead to overfitting or incorrect categorization of the sample case. However, it might not perform well with high-dimensional data or when the training set is very large due to its computational cost. The K- KNN algorithm is calculated as described in equation (7) (Ali, Chong and Manickam, 2023).

$$d_{xy} = \sqrt{\sum_{t=1}^{2} (x_t - y_t)^2} \qquad (7)$$

where:

- *x* and y are two points .
- x_t and y_t are components of *x* and *y*.

- **Random Forest (RF):** This ML technique is used and developed for handling classification and regression tasks. The algorithm falls under the category of ensemble learning—a process of combining many weak learners to obtain a stronger, more robust predictive model. The RF comprises an assemblage of decision trees, each developed on a randomly selected subset of the training dataset. By contrast, at each node of this tree, a random sample of features is considered in terms of splitting. Such a methodology tends to decrease overfitting and, therefore, generalize better (Sadhwani *et al.*, 2023).
- **Naive Bayes (NB):** NB classification is a fundamental probabilistic classifier widely used in ML for text classification tasks due to its simplicity, efficiency, and effectiveness. It is based on Bayes' theorem, a statistical method for calculating the probability of an event, given some evidence. NB classifiers assume independence between features, meaning the presence or absence of one feature does not influence the presence or absence of another, given the class label. This assumption, while not always strictly true, often works well in practice and is calculated as follows in equation (8) (Deepa, 2019).

$$P(c|x) = P(x|c)P(c) / P(x) \quad (8)$$

where:

- P(c|x) is the posterior probability of class (c) based on a set of predictors (x).
- P(x|c) is the joint probability of the class and the predictors.
- P(c) is the prior probability of the class.
- P(x) is the joint probability of the predictors.

These ML algorithms are selected due to their adaptability to the current task and their aptness for real-time implementation, where achieving both speed and accuracy is essential.

6. Metrics for Classification Models: ML models are evaluated in this section according to their accuracy:

- **Accuracy (CA):** The proportion of correct predictions out of the total predictions made.

4. RESULT AND DISCUSSION

In the first step of the experiment, the SDN records were directly classified by the ML methods without any preliminary feature ranking. The classification was performed with the help of the Orange Data Mining software (Orange Data Mining, 2024), , which is a free and open-source tool including data visualization, ML, and data mining functionalities. Orange has been exploited in several fields of knowledge, such as agriculture (Piñeiro et al., 2020), biomedical data mining (Urbanowicz et al., 2018), and mental health studies (Aristovnik et al., 2021). In

our analysis, 10-fold cross-validation with 10-fold to carry out the classification process using the KNN, RF, and NB algorithms. Results were visualized using the evaluate widget, which is an important component for presenting important performance metrics such as accuracy score. We conduct our evaluation on various comprehensive datasets to ensure robust and unbiased assessment of the performance of selected features and to provide more context to the obtained results. Specifically, we consider 23 features of the "DDoS attack SDN dataset," the "CICDDoS2019 Dataset" with a total of 84 features, and the "SDN-DDOS-TCP-SYN Dataset" with 28 features under our evaluation umbrella; it will guarantee the comprehensiveness of the evaluation. In Table 7, we will present our obtained results. Our experiments were conducted using a PC equipped with an Intel Core i5-10310U CPU, 16.0 GB of RAM, and a 64-bit Windows 11 operating system. Note that we refer to DB1 as (DDoS attack SDN dataset), DB2 (CICDDoS2019 Dataset), and DB3 (SDN-DDOS-TCP-SYN Dataset).

4.1. Cross-Features Selection

Cross-feature selection identifies features that have the highest predictive power and are most relevant to the target task. This process removes irrelevant or redundant information, which may introduce noise and potentially degrade model performance. In this way, the model can focus on the most informative features to better understand the underlying pattern and relationship of the data. To identify the most relevant network features for effective DDoS attack detection, we used the three algorithms mentioned earlier in the ranking techniques: Information Gain (IG), Gain Ratio (GR), and Gini Importance (GI). The feature selection process was performed in two iterations. During the first iteration, feature ranking methods were applied to the entire feature set, resulted in 14 important features were found to be most informative for the classification of DDoS, as shown in Tables 1, 2, and 3. In the second iteration, we apply cross features selection to find out the intersecting features resulting from all three ranking algorithms. Then, we extract the most relevant features for DDoS attack detection to be used for ML training and testing. The proposed cross-feature selection approach improves model accuracy as well as a decrease in the computation complexity. The number of intersecting features also varied across the three datasets, which reflects the differing importance of features within each dataset.

Table 1. Description of studied features in DB1.

NO.	Feature	Description
1	pktrate	Number of packets sent per second.
2	byteperflow	Byte count during a single flow.
3	pktperflow	Packet count during a single flow.
4	bytecount	Total number of bytes.
5	pktcount	Total number of packets.

NO.	Feature	Description
6	rx_kbps	Data receiving rate.
7	flows	Number of flow entries in the switch.
8	tx_bytes	Bytes are transferred from the switch port.
9	tot_dur	Total duration of flows.
10	dur	Duration in seconds.
11	Protocol	Protocol type used.
12	dt	Date and time.
13	src	Source IP address.
14	dst	Destination IP address.

Table 2. Description of studied features in DB2.

NO.	Feature	Description
1	Label	Indicates whether the traffic flow is benign or associated with a specific type of DDoS attack.
2	Bwd Packets/s	The number of packets sent in the backward direction per second.
3	Init Fwd Win Bytes	The initial window size in bytes in the forward direction.
4	Down/Up Ratio	The ratio of bytes downloaded to bytes uploaded.
5	Bwd Packet Length Max	The maximum length of packets in the backward direction.
6	Bwd Packet Length	The average length of packets in the backward direction.
7	Avg Bwd Segment Size	The average size of segments in the backward direction.
8	Total Backward Packets	The total number of packets sent in the backward direction.
9	Subflow Bwd Packets	The number of packets in the backward sub flow.
10	Bwd IAT Mean	The average time between packets arriving in the backward direction.
11	Bwd IAT Max	The maximum time between packets arriving in the backward direction.
12	Bwd IAT Total	The sum of all the inter-arrival times between packets in the backward direction.
13	Bwd IAT Min	The minimum time between packets arriving in the backward direction.
14	Bwd Header Length	The total length of headers in the packets sent in the backward direction.

Table 3. Description of studied features in DB3.

NO.	Feature	Description
1	tcp.analysis.push_bytes_sent	The number of bytes sent with the TCP PSH (Push) flag set.
2	frame.number	The unique identifier assigned to each frame captured in the dataset.
3	ip.len	The total length of the IP packet, including the header and data.
4	ip.id	The unique identifier assigned to each IP packet.
5	ip.flags	Flags used in the IP header to control or identify fragments.
6	tcp.len	The length of the TCP segment, including the header and data.

NO.	Feature	Description
7	tcp.nxtseq	The sequence number of the next expected byte in the TCP stream.
8	tcp.flags	Control flags in the TCP header used to manage the state of a connection.
9	tcp.window_size_value	The size of the TCP window is specified by the sender, which controls the flow of data.
10	tcp.window_size	The effective window size can be scaled by the window size scaling factor.
11	frame.len	The total length of the frame, including all headers and data.
12	tcp.stream	A unique identifier for each TCP connection (stream) in the capture.
13	eth.src	The MAC address of the source device in the Ethernet frame.
14	tcp.seq	The sequence number of the first byte of data in this TCP segment.

4.2. Cross-Feature Selection Result

The cross-feature selection process involves two steps: (a) feature ranking and (b) cross-feature selection, which are shown in the figures Fig.2, Fig.3, and Fig.4. By leveraging multiple feature selection algorithms, the most important features are extracted for identifying DDoS attacks on the SDN controller. This approach reduces the number of features while maintaining detection

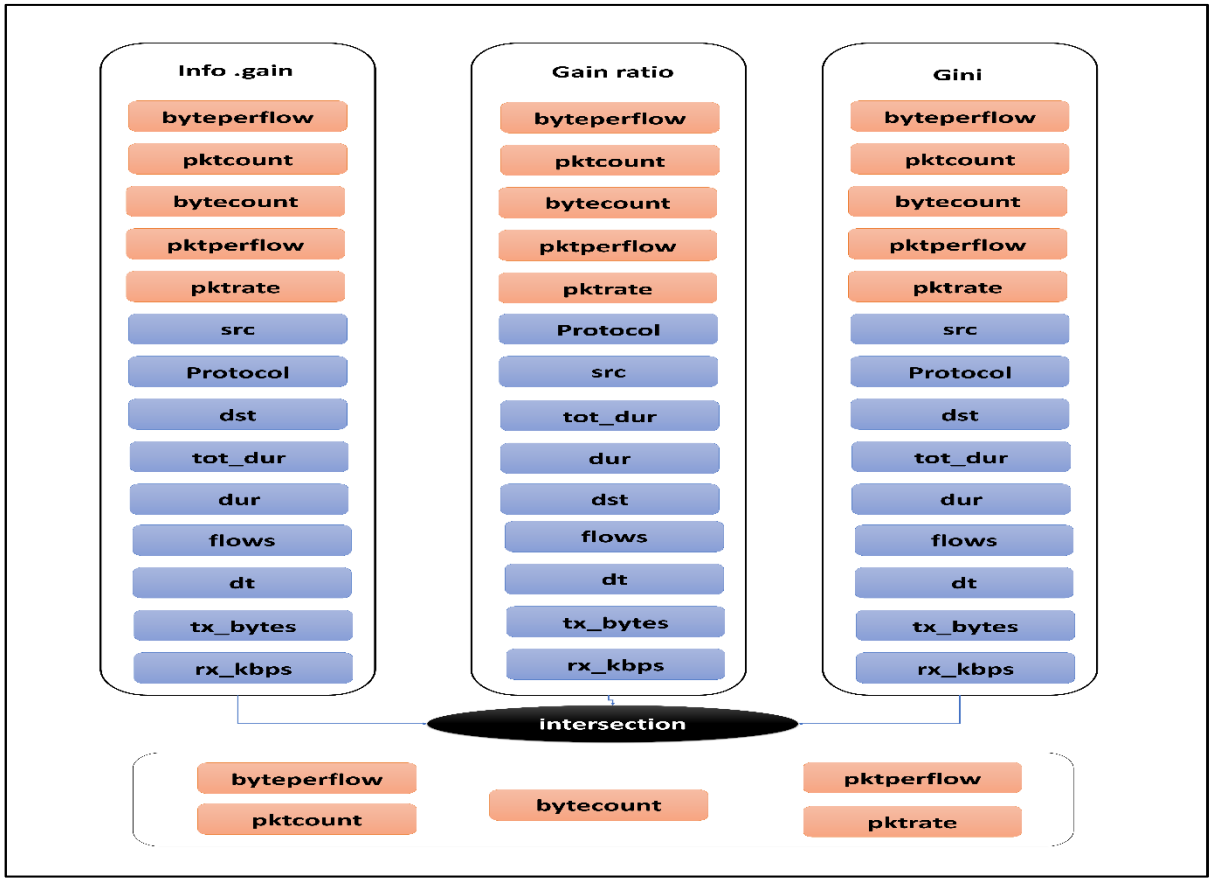

Fig. 2. Cross-Feature Selection Process for DB1.

accuracy. These features are essential for analyzing network traffic and identifying DDoS attacks, providing valuable insights into the behavior of network flows under normal and attack conditions. This approach resulted in varying numbers of features across datasets, ultimately leading to the highest accuracy in data detection and classification.

In Fig. 2 defines the methodology for the cross-feature selection process. This process of the 14 features is discerned by the triad of ranking algorithms applied to the DB1 dataset. The convergence of these features yields the quintet of 5 premier features pertinent to DDoS detection.

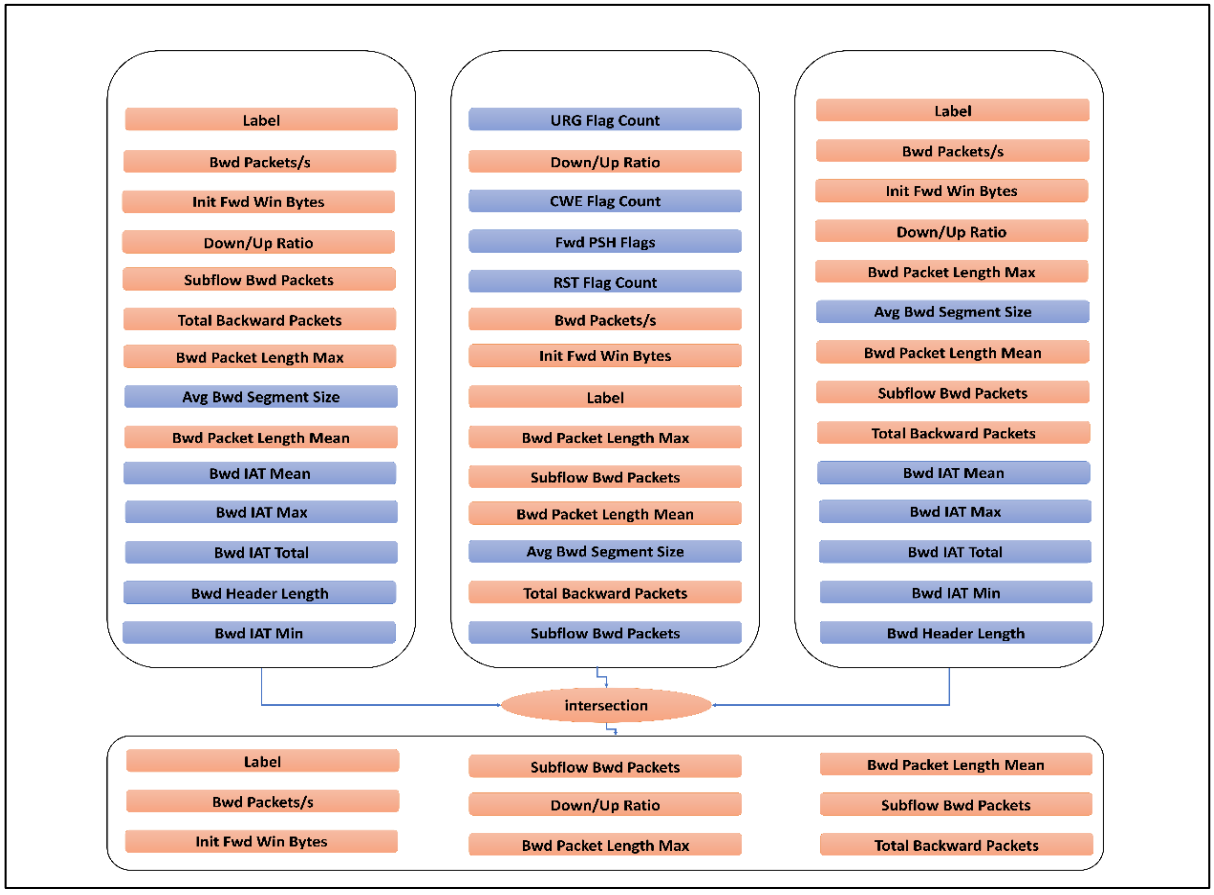

Fig. 3. Cross-Feature Selection Process for DB2.

Fig. 3 shows the cross-feature selection methodology, which amalgamates the 14 most relevant features identified by the three ranking algorithms within the DB2 dataset. The intersection of these features yields the 9 most prominent features pertinent to DDoS detection.

Fig. 4 illustrates the methodology employed in cross-feature selection. It amalgamates the foremost 14 features discerned by the three ranking algorithms within the DB3 dataset, establishing the leading 12 attributes for DDoS detection.

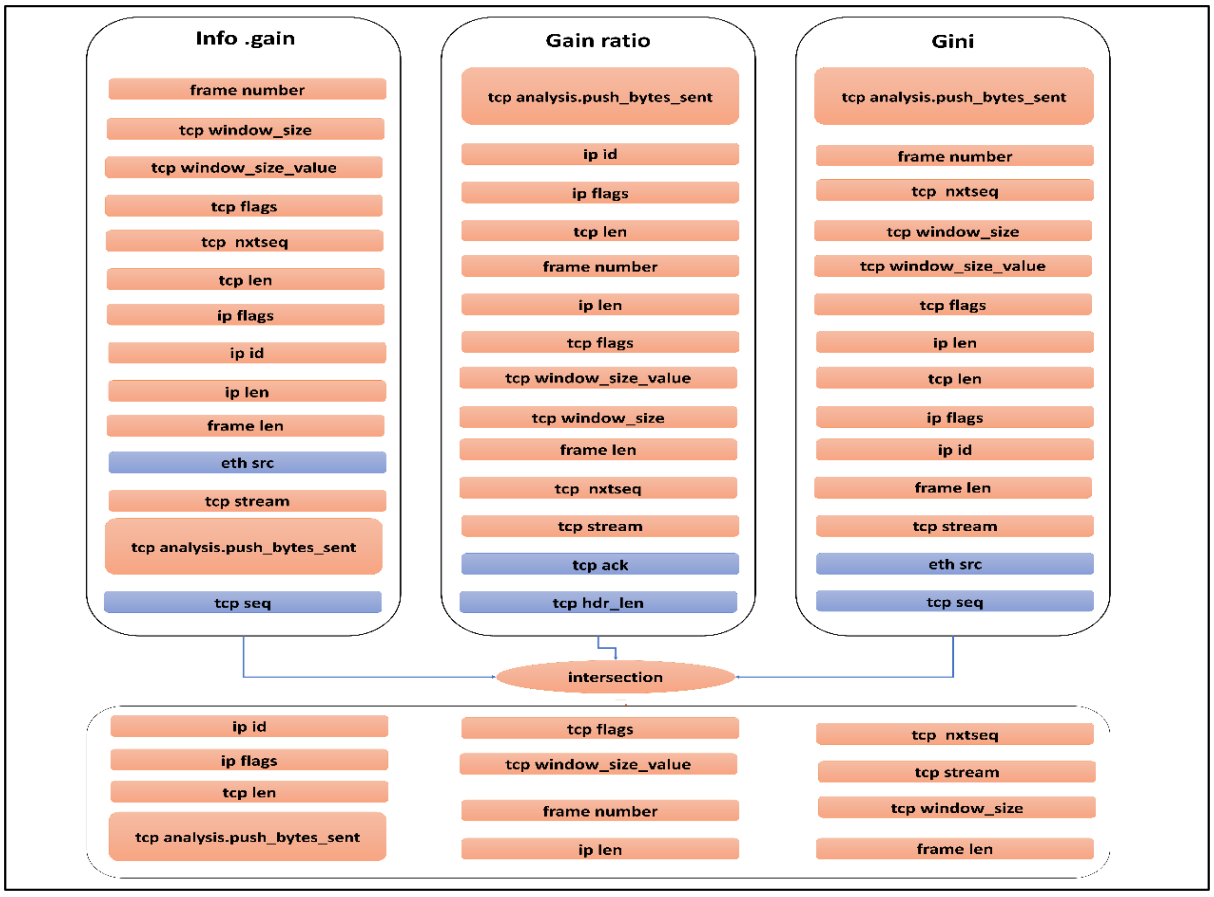

Fig. 4. Cross-Feature Selection Process for DB3.

4.3. Classification Results

A combination of NB, KNN, and RF algorithms, in conjunction with a robust cross-feature selection methodology, ensures an efficient and accurate detection mechanism for DDoS attacks. Focusing on the most informative features enables better understanding and identification of network anomalies, ultimately enhancing the overall security of the SDN controller. Here is a summary of the tested parameters for each ML algorithm across the three datasets shown in tables Tables 4,5 and 6:

Table 4. Summary of tested parameter of RF algorithm for 3 Dataset.

Tested parameter	Values
Cross-validation strategy	Group k-fold
Cross-validation split	10 folds
Number of Trees in the Forest	10
Number of Features	23, 80, and 28 features
Number of Features per Split	14,12,9 and 5
Minimum Samples per Split	5

Table 5. summary of tested parameter of KNN algorithm for 3 Dataset.

Tested parameter	Values
Cross-validation strategy	Group k-fold
Cross-validation split	10 folds
Distance Metric	Euclidean
Number of Features	23, 84, and 28 features
Number of Features per Split	14,12,9 and 5
Number of Neighbours	5

Table 6. Summary of tested parameter of NB algorithm for 3 Dataset.

Tested parameter	Values
Cross-validation strategy	Group k-fold
Cross-validation split	10 folds
Number of Trees in the Forest	10
Number of Features	23, 84, and 28 features
Number of Features per Split	14,12,9 and ٥

We present the results of the evaluated ML approaches and achieve the accuracy (CA) obtained through the dataset on all features. It is important to know that across all datasets in the first phase, as depicted below :

- DB1 (RF – 100% , KNN – 89.8% , NB – 83%).
- DB2 (RF – 100% , KNN – 99.6% , NB – 97.5%).
- DB2 (RF – 100% , KNN – 99.8% , NB – 88.4%).

For 14 features in the second phase:

- DB1 (RF – 100% , KNN – 91.4% , NB – 83.5%).
- DB2 (RF – 99.5% , KNN – 98.5% , NB – 79.8%).
- DB2 (RF – 100% , KNN – 100% , NB – 88.4%).

The balance between model complexity and classification performance is critical for effective online DDoS detection in an SDN environment, and our feature classification process enabled us to successfully achieve this balance. The accuracy rate achieved by implementing the ML model, True Positive Rates (TPR), and False Positive Rates (FPR) are illustrated in Table 7.

- **True Positive Rate (TPR**): TPR measures the proportion of actual positive cases that are correctly identified by the model. Mathematically, it is defined in equation (9):

$$TPR = True\ Positives / True\ Positives + False\ Negatives \quad (9)$$

- **False Positive Rate (FPR):** FPR measures the proportion of actual negative cases that are incorrectly classified as positive by the model. Mathematically, it is defined in equation (10):

$$FPR = False\ Positives / False\ Positives + True\ Negatives \quad (10)$$

Table 7. Classification results of ML model with cross-feature selection method with 3 datasets with cross-validation accuracy estimation (10-fold).

Dataset	Model	Training time	Testing time	CA	TPR	FPR
DB 1	RF	13.487	0.273	100 %	0.999	0.0004
5-features	KNN	3.762	4.668	99.5 %	1.0003	0.004
	NB	2.152	0.108	82.5 %	0.830	0.178
DB 2	RF	43.962	1.717	100 %	1	0
9-features	KNN	20.236	603.312	99.9 %	0.997	0.001
	NB	14.917	0.552	99.8 %	1	0.002
DB 3	RF	194.028	7.649	100 %	0.999	0.000003
12-features	KNN	41.772	26502.162	100 %	0.996	0.000002
	NB	76.789	3.166	94.1 %	1	0.063

Table 7 presents the classification results using several ML algorithms on the tested dataset as can be seen from the results the RF, KNN and NB algorithms achieved the highest accuracy of (RF= 100 % , NB = 99.8 % , KNN=100 %). The classification results using RF, KNN, and NB on three datasets showed high TPR and low FPR, indicating excellent accuracy with minimal false positives. Some cases had lower TPR and higher FPR, highlighting a trade-off between detection accuracy and false positives. By thoroughly testing the dataset and experimenting with various ML algorithms, coupled with a careful process of feature classification and dimensionality reduction, the proposed approach achieved a significant reduction in both training and testing times. This is evidenced by the results presented in Table 7. Furthermore, the RF algorithm was able to successfully detect 100% of the DDoS attacks in the DB1 and DB2 datasets. This exceptional result can be attributed to the comprehensive nature of the datasets and the inherent strengths of the RF algorithm in dealing with complex, non-linear relationships within high-dimensional data. However, many studies such as (Tonkal *et al.*, 2021) with 23 features and (Sadhwani *et al.*, 2023) with 15 features have proven this achievement, confirming the effectiveness of the algorithm in effectively detecting DDoS attacks. As with any ML model, it is crucial to undertake a thorough experimentation process when considering feature reduction techniques for DDoS detection systems. This involves several key steps to ensure that the chosen feature set optimally balances efficiency and detection accuracy (experimentation with different feature sets, cross-validation, evaluation of the impact on model performance, consideration of real-world constraints, and iterative optimization). The evaluated ML algorithms in data classification involve factors such as computational complexity, model training time, and interpretability. On the other hand, NB works well with smaller datasets and is particularly effective if the features are conditionally independent in the given class. However, it assumes independence among features, which may not hold in practice and can potentially reduce accuracy. RF is highly effective with large

datasets, robust to overfitting, and capable of handling various feature types, offering high accuracy and good performance. However, it is computationally intensive during training, requires significant memory, and can be slow with extensive datasets, while KNN is simple and effective for smaller datasets and scenarios where real-time prediction speed is not critical, but it can be computationally expensive during prediction, especially with large datasets.

Complexity varies based on data size, dimensionality, and application requirements in real-time scenarios. However, the KNN algorithm has been effectively utilized in a range of studies (Hosseini and Azizi, 2019; Dong and Sarem, 2020 and Tonkal et al., 2021). In our test, we observed that KNN typically involves lengthy training and testing phases. Therefore, we conclude that although it is easy to implement and can yield satisfactory outcomes, the substantial computational complexity associated with KNN is a key factor to consider when selecting an appropriate classification algorithm.

4.4. Feature Reduction and Classifier Evaluation

To enhance analysis and model performance, we focused on refining the dataset by streamlining the feature set. Initially comprising 23 features in the " DDOS attack SDN Dataset ", this dataset underwent rigorous reduction processes, resulting in a notable decrease of 39%, culminating in a condensed set of 14 features, and for 5 features is 78%. These carefully chosen features make up the core of our experiment in DDoS classification.

In Table 8, we conduct a comparison between our current study and some of the recent works. The comparison involved contrasting the employed methods, assessing the number of tools utilized, the number of features used, the diversity of the evaluated databases, and the strategies employed for feature reduction. Our comparison demonstrates a significant reduction in the number of features while maintaining high classification accuracy levels. In this table, we refer to ML accuracy as (ML-CA), dataset as (DS), Feature number as (F-N), and Feature selection as (F-S).

4.5. Complexity Analysis

In Table 9 we will examine the time complexity (both training (Buczak and Guven, 2016) and testing) besides the space complexity of the employed ML algorithms in the context of reduced classification features.

Where N is the number of training samples, k is the number of required neighbors, f is the number of reduced features, M is the number of trees in the RF algorithms, and C is the number of classes in the NB algorithm.

Table 8. The comparison of the related studies.

Works	Method		ML-CA	DS	F-N	F-S
	ML	DL				
(Tonkal et al., 2021)	✓	×	KNN= 97.7 % DT= 99.1 % ANN= 96.2 % SVM= 81.4 %	DDOS attack SDN dataset	8	NCA
(Mansoor et al., 2023)	×	✓	RNN=94.18%	DDOS attack SDN dataset	5	• IGR • Chi-square
(Ma et al., 2023)	✓	×	RF= 99.99 %	cicddos2019 dataset	24	• Filter • Wrapper • Embedded
(Samaan and Jeiad, 2023)	✓	×	DT=93.62% RF=93.6 % LR=93.39 % GBT=93.66 %	SDN-DDOS-TCP-SYN dataset	٥	ANOVA F-test
Our study	✓	×	RF= ١٠٠ % NB = 99.8 % KNN=100 %	1-DDOS attack SDN dataset 2- cicddos2019 3-SDN-DDOS-TCP-SYN	DB1=5 DB2= 9 DB3= 12	• IG • IGR • GI

Table 9. The complexity of the used ML methods.

Algorithm	Training time	Testing time	Space
KNN	O(N log k)	O(N * f)	O(N * f)
RF	O(N × log(N) × M)	O(M * log(N))	O(N * M * log(N))
NB	O(N * f)	O(f)	O(C * f)

The training time of KNN is relatively low (*O(N)*) when it only involves storing the training data and if the *k* smallest distances are needed. However, the prediction time (testing) is high due to the need to calculate distances between the tested instance and all of the training instances. As for the required space to implement the classification model, it only requires storing the entire training dataset with *f* features per sample. In the case of RF, the complexity of the training, prediction time, and space complexity is high and proportional to the number of training samples (*N*), the number of trees (*M*), and the depth of the trees (*log(N)*). Therefore, the model will not be affected by the reduced number of features. On the other hand, the prediction time of the NB algorithm is highly affected by the selected features, since it calculates the probability of each class based on the input features and selects the class with the highest probability. The aforementioned statements have been experimentally proven in Table 7 concerning the training time and testing time for all evaluated algorithms. We found that NB has a reasonable time and space complexity, making them a suitable choice for your DDoS

detection problem. However, RF is often a strong candidate for classification due to its high performance and effectiveness with large datasets. While it may have higher computational requirements during the training phase.

Furthermore, we conducted a comparative analysis with another study that used Neighborhood Component Analysis (NCA) and appeared in (Tonkal *et al.*, 2021). We note that NCA is more computationally intensive than simpler classification algorithms because it involves an optimization process to learn the transformation matrix. While classification algorithms provide a fast and intuitive way to evaluate feature importance, NCA potentially identifies more complex and multivariate relationships between features and the target variable, which may lead to more robust and accurate feature selection for particular tasks. Despite this distinction, our approach has achieved more promising results.

Based on the analysis and result findings, the conclusive recommendation for action is delineated as follows: NB has reasonable time and space complexity, rendering it an appropriate selection for addressing the DDoS detection challenge. It is particularly advocated when temporal and resource allocation efficiency is paramount. The RF algorithm frequently exhibits exceptional performance and efficacy when applied to extensive datasets, positioning it as a formidable contender for classification endeavors. Although it may necessitate elevated computational resources during the training phase, it compensates with enhanced accuracy and effectiveness in detection capabilities. Ultimately, the appropriateness of NB, RF, or KNN is profoundly influenced by the requirements and constraints inherent in the real-time classification task.

5. CONCLUSION

In this study, we leverage ML algorithms to classify DDoS attack traffic from normal traffic on different datasets obtained from the SDN environment with minimal feature extraction. Relying on several features in an ML model may reduce the effectiveness and increase the complexity of the real-time detection model. Therefore, to minimize the execution time and computational complexity in real-time applications, we employ various feature ranking methods, including (IG, IGR, and GI), and then we use cross-feature selection of the resulting feature ranking process for only 14 features to identify the most relevant network features. Then, apply ML algorithms such as (Random Forest (RF), Naive Rule (NB), and K-Nearest Neighbors (KNN)) to classify DDoS attacks. For the used datasets, we used various public datasets such as (DDoS attack SDN, CICDDoS2019, and SDN-DDOS-TCP-SYN dataset). We reduced the number of the most relevant features to 5, 9, and 12 for each dataset, respectively. Using the reduced

features, we achieved significant results in classifying DDoS attacks using less complex ML models (NB and KNN), making the model ready for real-time implementation. We found that such ML methods achieved significant results in terms of classification accuracy, and less computational complexity. The results indicate that RF achieved 100%, NB achieved 99.8%, and KNN achieved 100%. This demonstrates that feature reduction and correct algorithms can help achieve the best results for finding attacks in each dataset. In future studies, we plan to implement the proposed model for real-time DDoS classification on SDN controllers, leveraging the insights gained from this work to enhance the robustness and efficiency of DDoS attack detection in the SDN environment.

6. REFERENCES

Ali, T.E., Chong, Y.W. and Manickam, S. (2023) 'Comparison of ML/DL Approaches for Detecting DDoS Attacks in SDN', Applied Sciences (Switzerland), 13(5). Available at: https://doi.org/10.3390/app13053033.

Aristovnik, A. et al. (2021) 'Impacts of the Covid-19 Pandemic on Life of Higher Education Students: Global Survey Dataset from the First Wave', Data in Brief, 39(January), pp. 1–34. Available at: https://doi.org/10.1016/j.dib.2021.107659.

Barznji, A.O. and Ameen, J.J.H. (2021) 'Wi-Max Network Simulation for Salahaddin University New Campus', Kufa Journal of Engineering, 12(4), pp. 1–13. Available at: https://doi.org/10.30572/2018/kje/120401.

Buczak, A.L. and Guven, E. (2016) 'A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection', IEEE Communications Surveys and Tutorials, 18(2), pp. 1153–1176. Available at: https://doi.org/10.1109/COMST.2015.2494502.

Deepa, V. (2019) 'Design of Ensemble Learning Methods for DDoS Detection in SDN Environment', 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), pp. 1–6.

Dong, S. and Sarem, M. (2020) 'DDoS Attack Detection Method Based on Improved KNN with the Degree of DDoS Attack in Software-Defined Networks', IEEE Access, 8, pp. 5039–5048. Available at: https://doi.org/10.1109/ACCESS.2019.2963077.

Gupta, Karan; Sharma, Shivam; Kumar, S. (2021) No Title. Available at: https://doi.org/10.17632/8nb4cdwc9h.1.

Han, D. et al. (2024) ‘Traffic Feature Selection and Distributed Denial of Service Attack Detection in Software-Defined Networks Based on Machine Learning’, Sensors, 24(13). Available at: https://doi.org/10.3390/s24134344.

Hosseini, S. and Azizi, M. (2019) ‘The hybrid technique for DDoS detection with supervised learning algorithms’, Computer Networks, 158, pp. 35–45. Available at: https://doi.org/10.1016/j.comnet.2019.04.027.

Laassiri, F., Moughit, M. and Idboufker, N. (2018) ‘An Improvement of Performance in 4G LTE Using Software Defined Network’, Colloquium in Information Science and Technology, CIST, 2018-Octob(12), pp. 508–513. Available at: https://doi.org/10.1109/CIST.2018.8596517.

Ma, R. et al. (2023) ‘Real-Time Detection of DDoS Attacks Based on Random Forest in SDN’, Applied Sciences (Switzerland), 13(13). Available at: https://doi.org/10.3390/app13137872.

Mansoor, A. et al. (2023) ‘Deep Learning-Based Approach for Detecting DDoS Attack on Software-Defined Networking Controller’, pp. 1–21.

Nadeem, M.W. et al. (2022) ‘Ddos detection in sdn usingmachine learning techniques’, Computers, Materials and Continua, 71(1), pp. 771–789. Available at: https://doi.org/10.32604/cmc.2022.021669.

Najar, A.A. and Manohar Naik, S. (2024) ‘Cyber-Secure SDN: A CNN-Based Approach for Efficient Detection and Mitigation of DDoS attacks’, Computers and Security, 139, p. 103716. Available at: https://doi.org/10.1016/j.cose.2024.103716.

Nisha Ahuja Singal, Gaurav Mukhopadhyay, D. (2020) Dataset”, “DDOS attack SDN, 27 Sep 2020. Available at: https://doi.org/10.17632/jxpfjc64kr.1.

O. Hasan, A. (2022) ‘Application Based performance monitoring heavy data transmission of Local Area Network’, Kufa Journal of Engineering, 13(3), pp. 14–40. Available at: https://doi.org/10.30572/2018/kje/130302.

Orange Data Mining (2024) 2024.

Piñeiro, V. et al. (2020) ‘A scoping review on incentives for adoption of sustainable agricultural practices and their outcomes’, Nature Sustainability, 3(10), pp. 809–820. Available at: https://doi.org/10.1038/s41893-020-00617-y.

Polat, H. and Polat, O. (2020) ‘Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models.pdf’, Mdpi [Preprint].

Sadhwani, S. et al. (2023) ‘A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques’, Applied Sciences (Switzerland), 13(17). Available at: https://doi.org/10.3390/app13179937.

Samaan, S.S. and Jeiad, H.A. (2023) ‘Feature-based real-time distributed denial of service detection in SDN using machine learning and Spark’, Bulletin of Electrical Engineering and Informatics, 12(4), pp. 2302–2312. Available at: https://doi.org/10.11591/eei.v12i4.4711.

Santos, R. et al. (2020) ‘Machine learning algorithms to detect DDoS attacks in SDN’, Concurrency and Computation: Practice and Experience, 32(16), pp. 1–14. Available at: https://doi.org/10.1002/cpe.5402.

Talukder, Md Alamin; Uddin, M.A. (2023) “CIC-DDoS2019 Dataset”, 3 March 2023 | Version 1. Available at: https://doi.org/10.17632/ssnc74xm6r.1.

Tonkal, Ö. et al. (2021) ‘Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking’, Electronics, 10(11), p. 1227. Available at: https://doi.org/10.3390/electronics10111227.

Urbanowicz, R.J. et al. (2018) ‘Relief-based feature selection: Introduction and review’, Journal of Biomedical Informatics, 85, pp. 189–203. Available at: https://doi.org/10.1016/j.jbi.2018.07.014.

Xing, E.P., Jordan, M.I. and Karp, R.M. (2001) ‘Feature selection for high-dimensional genomic microarray data’, Proceedings of the 18th International Conference on Machine Learning, pp. 601–608.