Cluster Computing (2022) 25:1355–1372
https://doi.org/10.1007/s10586-022-03538-x

SSK-DDoS: distributed stream processing framework based classification system for DDoS attacks

Nilesh Vishwasrao Patil[1] · C. Rama Krishna[1] · Krishan Kumar[2]

Received: 3 June 2021 / Revised: 4 January 2022 / Accepted: 5 January 2022 / Published online: 17 January 2022

Abstract
Distributed denial of service (DDoS) is an immense threat for Internet based-applications and their resources. It immediately floods the victim system by transmitting a large number of network packets, and due to this, the victim system resources become unavailable for legitimate users. Therefore, this attack is claimed to be a dangerous attack for Internet-based applications and their resources. Several security approaches have been proposed in the literature to protect Internet-based applications from this type of threat. However, the frequency and strength of DDoS attacks are increasing day-by-day. Further, most of the traditional and distributed processing frameworks-based DDoS attack detection systems analyzed network flows in offline batch processing. Hence, they failed to classify network flows in real-time. This paper proposes a novel Spark Streaming and Kafka-based distributed classification system, named by SSK-DDoS, for classifying different types of DDoS attacks and legitimate network flows. This classification approach is implemented using a distributed Spark MLlib machine learning algorithms on a Hadoop cluster and deployed on the Spark streaming platform to classify streams in real-time. The incoming streams consume by Kafka's topic to perform preprocessing tasks such as extracting and formulating features for classifying them into seven groups: Benign, DDoS-DNS, DDoS-LDAP, DDoS-MSSQL, DDoS-NetBIOS, DDoS-UDP, and DDoS-SYN. Further, the SSK-DDoS classification system stores formulated features with their predicted class into the HDFS that will help to retrain the distributed classification approach using a new set of samples. The proposed SSK-DDoS classification system has been validated using the recent CICDDoS2019 dataset. The results show that the proposed SSK-DDoS efficiently classified network flows into seven classes and stored formulated features with the predicted value of each incoming network flow into HDFS.

Keywords DDoS attacks · Distributed stream processing frameworks · Big data · Apache Spark · Apache Kafka · Apache Hadoop · Spark MLlib machine learning

1 Introduction

Over the decade, companies have been running their services online for growing revenue and are open to users from anywhere-anytime. Further, in recent times, there is huge growth in Internet subscribers and connecting devices. However, this significant growth has come up with unsafe network routes with non-secure connecting devices. Therefore, attackers use this chance to compromise numerous nodes to form a botnet for performing DDoS attacks on the victim system.

1.1 DDoS attacks

A DDoS attack is the biggest threat to Internet-based applications and their resources [1, 2]. The motive of this attack is to overwhelm Internet-based services by transmitting a large amount of attack traffic [3, 4]. A typical example to perform the DDoS attack on the victim system is presented in Fig. 1. In this, a master took control of various slaves with the help of handler programs. The handler is the inter-mediator program between master and

✉ Nilesh Vishwasrao Patil
nilesh.cse18@nitttrchd.ac.in

[1] Computer Science & Engineering, National Institute of Technical Teachers Training & Research, Chandigarh, Panjab University, Chandigarh, India

[2] University Institute of Engineering & Technology, Panjab University, Chandigarh, India

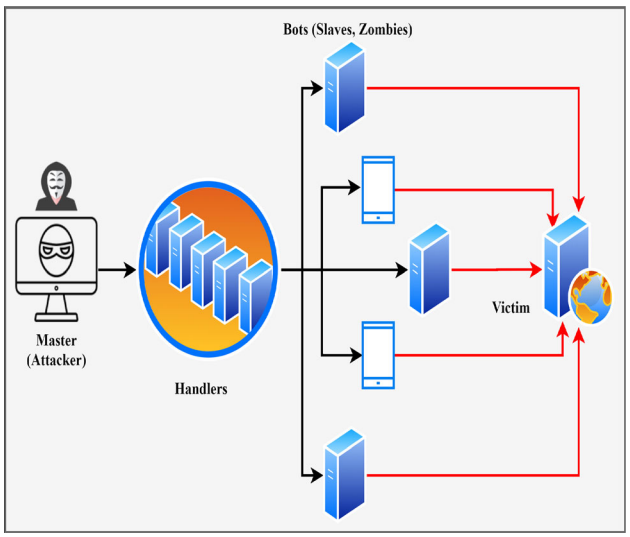

Fig. 1 A typical example of DDoS attack

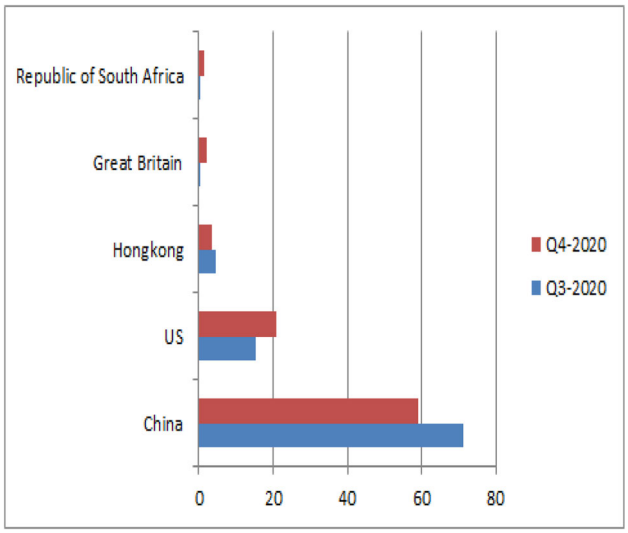

Fig. 2 Comparison of Q3-2020 and Q4-2020 country-wise statistics distribution of DDoS attacks [5, 6]

slave nodes that will help to perform a large-scale DDoS attack on victim-applications.

1.2 Summary of DDoS attack events

Each country has been struggling with the COVID-19 situation since Jan 2020. In this pandemic, peoples are working, shopping, enjoying, etc. in online mode. Therefore, attackers use this chance to compromise numerous nodes to form a botnet. The Q4-2020 DDoS attacks statistical report [5] is summarized as follows:

1. Most numbers of attacks experienced by countries: China (44%+), USA (23%+), and Hong Kong (7%+).
2. The highest number of attacks reported on Dec 31, 2020, i.e., 1349 incidents.
3. After exception in the last few quarters, once again Linux-based botnets used to launch every DDoS attack.
4. The majority of C&C servers located in the USA (36%+), Netherlands (19%+), and Germany (8%+).
5. Once again, most number of incidents observed on Thursday, and this trend dropped on Sunday.

The country-wise distribution of DDoS attacks incidents for Q3-2020 and Q4-2020 are given in Fig. 2. From this, we can conclude that both frequencies and the strength of attacks are increasing year-after-year. Further, the attack strength pattern shifted from "Gbps to Tbps". Therefore, one more challenge in front of researchers to systematically analyze such a large volume of traffic.

1.3 Challenges

In this big data world, the traditional framework-based DDoS attack detection approaches themselves become the victim while examining a massive number of packets. Therefore, there is a need to deploy the proposed approach on distributed stream processing framework (DSPF). The DSPF has the capability to handle (store and analyze) a large volume of data in real-time by employing multiple nodes. Further, data transfer between nodes, secure communication protocol, and metadata information is systematically managed by DSPF. The traditional and distributed processing frameworks (DPF) based DDoS attack detection systems are specially designed to examine flows in an offline mode. Therefore, this type of approach fails to analyze incoming streams in real-time. Additionally, most of the approaches have been tested on outdated datasets. Therefore, there is a need to design a distributed classification model using a recent dataset and deploy it on DSPF (such as the Spark Streaming platform).

1.4 Open-source technologies

In this section, we are going to summarize the open-source technologies that are required to design the proposed SSK-DDoS classification systems for DDoS attacks. We split-up this section into four sub-sections: Apache Hadoop, Spark Streaming, Apache Kafka, and CICFlowMeter.

A good DSPF must have the following features:

1. To analyze the streaming data such as network traffic flows as it receives and takes immediate action based on prediction.
2. To design real-time applications which have a loosely-coupled architecture. Therefore, multiple publishers and consumers can independently access the application without delay.

3. To have features like analyze data in a distributed manner, extremely low latency, reliability, scalable, fault-tolerant, etc.

1.4.1 Apache Hadoop

Apache Hadoop [7, 8] is one of the powerful DPF for storing and analyzing a large amount of data. It is specially designed to analyze a large amount of data using batch processing on a cluster of nodes. It consists of three major modules:

1. Hadoop Distributed File System (HDFS): It allows for storing a large amount of data on clusters of nodes called datanodes. The data is divided into multiple blocks and systematically stored on datanodes. Further, metadata information about each block is stored in namenode.
2. Yet Another Resource Navigator (YARN): This module is used to allocate resources for analyzing a large amount of data.
3. MapReduce: It is a programming model for analyzing a large amount of data in a distributed manner.

1.4.2 Apache Spark streaming

Apache Spark [9] is a large-scale data analytics engine. It provides a large data processing API. Spark Streaming is an extension of the core Spark API for developing real-time applications. The Apache Spark streaming platform is commonly used:

1. To design real-time applications for analyzing a large amount of data in real-time.
2. To immediately respond to the streaming data to take quick action without a delay.

Apache Spark consists of four essential components: Spark SQL, MLlib, GraphX, and Spark Streaming. It is possible to combine these four components to design a machine learning-based real-time application. Spark Machine Learning Library (MLlib) is a distributed in-memory machine learning library. It provides:

1. A way to design a model in a distributed manner.
2. Robust APIs.
3. High-scalability feature for the machine learning model when deployed on DPF/DSPF.
4. Support various programming languages: Python, Java, Scala, etc.

Several tools/techniques are available to design traditional and non-traditional machine learning models such as Python, Java, R, WEKA, etc. Further, few authors [10–13] have systematically discussed machine/deep learning methods and features selection. However, when we design a model using these techniques that will face the scalability issue when deployed on DPF/DSPF. The Spark MLlib machine learning library provides a way to design a distributed and in-memory machine learning model. This type of model is specially designed to deploy on DPF/DSPF (Hadoop, Kafka, Spark, etc.). Therefore, it is exciting to implement a distributed classification approach for DDoS attacks using the MLlib and deploy it on the Spark streaming platform.

1.4.3 Apache Kafka

Apache Kafka [14] is an open-source distributed and high-throughput publish-subscribe messaging system. It consists of six essential components: Brokers, Zookeeper, Topics, Partitions, Publishers, and Subscribers. The publishing/consuming feature of Kafka helps to provide a loosely-coupled architecture to real-time applications.

1.4.4 CICFlowMeter

CICFlowMeter [15] is an open-source network flow generator tool. It creates network flows in offline (from PCAP) and online (from network interfaces) mode. It creates 83 attributes and stores them in a CSV file from network traffic. An example of CICFlowMeter for collecting network packets using the network interface card and generating network flows from network packets is presented in Fig. 3.

1.5 Contributions

The significant contributions of this paper are listed in the following:

- Proposed a novel Spark Streaming and Kafka based classification system for DDoS attacks called SSK-DDoS.
- The SSK-DDoS is distributed and real-time classification approach built using distributed Spark MLlib machine learning algorithms on the Hadoop cluster and deployed on the Spark Streaming clusters to classify network flows in real-time.
- It stores formulated features of each network flow with predicted class in the HDFS to retrain the model using a new set of samples.
- Proposed SSK-DDoS classification system distributes the computational overhead i.e. preprocessing and classification tasks on network traffic between multiple nodes of Spark clusters.
- Proposed distributed SSK-DDoS runs in an automated style as incoming network flows published on Kafka

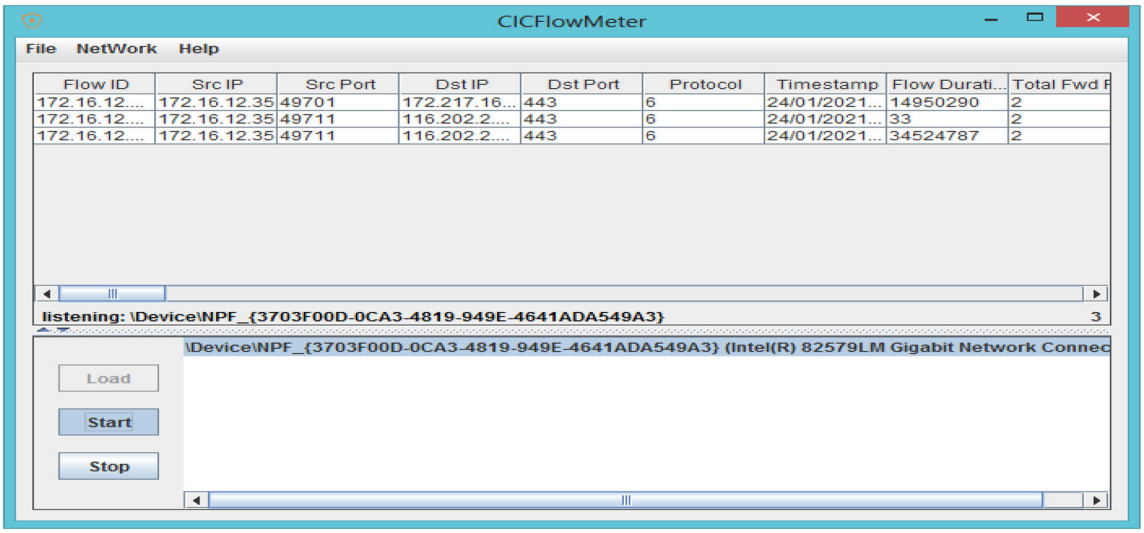

Fig. 3 CICFlowMeter: Capture incoming network traffic

topics, select essential variables, formulate features based on selected variables, perform classification job, and finally publish predictions on the Kafka topic to take action in real-time.
- Proposed SSK-DDoS classification approach is designed and validated using the recent CICDDoS2019 dataset.
- Proposed SSK-DDoS is a highly-scalable approach and provides loosely-coupled architecture.

Rest of the paper is organized as follows. A summary of related works presented in Sect. 2. Section 3 presents a novel distributed SSK-DDoS classification system for DDoS attacks. Section 4 provides testbed information of the classification approach. Results and analysis is presented in Sect. 5. Finally, Sect. 6 conclude the paper.

2 Related work

Numerous security approaches are available in the literature to protect the victim systems from different DDoS attacks. Patil et al. [16] have systematically classified DDoS attack detection approaches into two broad classes based on their deployment frameworks: traditional and DPF based detection approaches. In the literature [17–30], several authors systematically summarized traditional framework based approaches and few of the recent existing systems are [31–33]. However, few authors [16] specifically addressed DPF based approaches. The DPF (batch processing) and DSPF (real-time) themselves have distributed designs to store and analyze a massive volume of data on a cluster of nodes. In the literature, some authors [34–54], proposed DPF and DSPF based approaches. However, most of them are deployed on the DPF. Therefore, this type of detection approach efficiently analyzes a large number of packets and classifies them in a short time. However, they are not capable to classify network flows in real-time. This type of approach is useful for historical data analysis and retrain the distributed model. Therefore, if use-case demands to classify network flows in real-time then one need to deploy the proposed approach on DSPF (such as Spark Streaming platform).

We have drawn some inferences from the existing works related to DPF/DSPF. They are listed as follows:

- Most of the systems are designed and tested in an offline mode. Therefore, there is a need to deploy a classification model for DDoS attacks on DSPF such as Apache Spark Streaming that analyzes network traffic in real-time.
- Few researchers designed their classification model using shallow and deep learning algorithms. These models performed exceptionally well when we deployed on traditional frameworks. However, models will undergo the scalability issue when deployed on DPF/DSPF. Therefore, there is a need to implement a distributed model using distributed machine learning library that will provide a high scalability feature even models deployed on DPF/DSPF.
- Most of the DPF/DSPF based DDoS approaches efficiently analyzed a huge amount of network flows on a group of nodes by distributing the analysis task on multiple systems.
- Most of the existing DPF/DSPF based DDoS mechanisms employed a counter-based detection methodology for identifying the high-volume of attacks. Therefore, this type of system fails to recognize a low-volume of DDoS attacks.

– Most of the DPF/DSPF and traditional framework-based DDoS mechanisms are validated using outdated datasets. Few authors [55] designed there system using recent dataset. Therefore, there is a need for a new classification approach that can be validated using recent datasets, such as CICDDoS2019.

3 SSK-DDoS: Spark Streaming and Kafka based classification system for DDoS attacks

This section presents the functioning of the proposed SSK-DDoS classification system for DDoS attacks. The logical architecture of SSK-DDoS is given in Fig. 4.

The distributed SSK-DDoS classification system of DDoS attacks is consists of three Spark Streaming clusters: 'SC-1', 'SC-2', and 'SC-3'. Two Spark clusters 'SC-1' and 'SC-2' are deployed in the intermediate network i.e., at ISP-1 and ISP-2 respectively. The primary job of 'SC-1' and 'SC-2' clusters is to preprocess the incoming network traffic and pass it on to 'SC-3'. While the 'SC-3' cluster is deployed in the victim network and the job of this cluster is to classify flows into seven classes. The first step is producer agents (from ISP-1 and ISP-2) continuously publishing network flows generated by CICFlowMeter onto the "ssk_ddos_flow" topic. Both 'SC-1' and 'SC-2' clusters immediately consume flows from "ssk_ddos_-flow" topic. The second step is to extract essential variables from flows, formulate features using extracted variables, and publish them on "sss_ddos_features" topic. Then 'SC-3' cluster immediately consumes formulated features of each flow from "sss_ddos_features", classify them into seven classes, and publish predicted class on the "sss-ddos_prediction" topic to take action. Further, this system stores formulated features of each flow with predicted class into the HDFS that will help to retrain the distributed classification model of DDoS attacks using a new set of samples. Highlights of the proposed distributed SSK-DDoS classification system of DDoS attacks are as follows:

– Loosely-coupled architecture as it uses distributed publish-subscribe messaging system for communication
– Analyze network traffic flows in real-time using Spark Streaming API
– Distributed computational overhead between three clusters

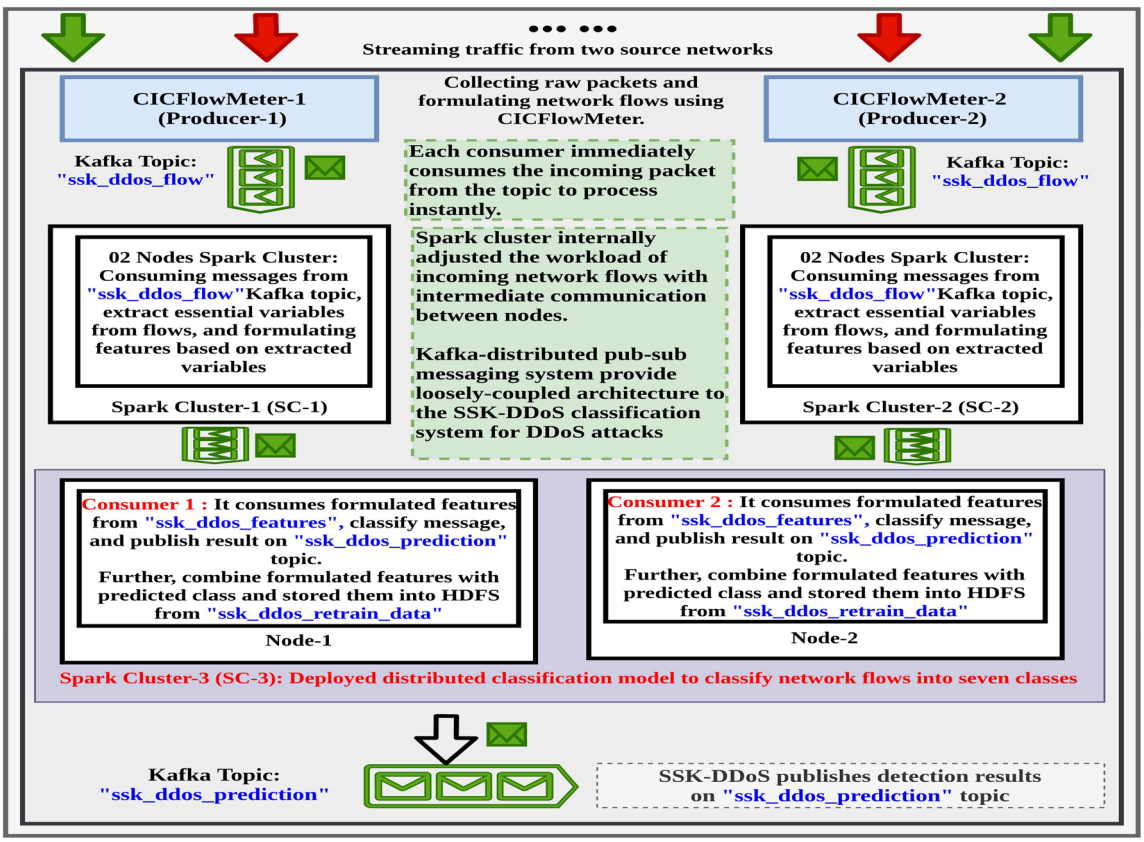

Fig. 4 Logical architecture of the proposed distributed SSK-DDoS classification system for DDoS attacks

– Stores formulated features of each flow with their predicted class into HDFS for retraining the existing classification model using a new set of samples

The detection approach of the proposed SSK-DDoS classification system splits into two parts: preprocessing and classification task.

3.1 Preprocessing task

The role of 'SC-1' and 'SC-2' clusters is to consume network traffic, generate network flows using CICFlowMeter, select significant variables, scale selected variable, formulate features using scaled variables, and finally publish it on the "ssk_ddos_features". Both 'SC-1' and 'SC-2' have a separate Kafka topic with the same name "ssk_ddos_features". We split this section into three subsections: create network flows, scaling variables, and formulating features.

3.1.1 Create network flows using CICFlowMeter

The CICFlowMeter generates network flows with 83 attributes from incoming traffic and puts flows in a CSV file. We employ producer agents to immediately pick up each entry from CSV and publish flows on the "ssk_ddos_flow" topic. The next task perform by 'SC-1' and 'SC-2' clusters is to select 23 significant variables from each flow. In [56], 24 significant variables are used to classify flows into different classes. However, in these 24 variables, two variables such as Fwd_Header_Length and Fwd_Header_Length.1 look like duplicate columns. Further, after generating network flows using the current version of CICFlowMeter, the Fwd_Header_Length.1 variable is removed from generated network flows. Therefore, we have selected 23 variables from the variable list of each network flows.

3.1.2 Scaling data values

The next job performed by both clusters is to scaling data values of twenty-three variables on the same scale. The scaling of data points can be adjusted with the help of the "MinMax" technique provided by the "sklearn.preprocessing". Therefore, after the scaling process, data point values lie between 0 and 1. The mathematical formula for the scaling is:

$$Norm\_Data_i = \frac{DataVal_i - min(DataVal)}{max(DataVal) - min(DataVal)} \quad (1)$$

3.1.3 Features formulation

Both 'SC-1' and 'SC-2' formulate ten features from 23 selected variables. It helps to enhance the accuracy and speed up the design process of the classification model. A summary of each feature is given in Table 1. After formulating features by 'SC-1' and 'SC-2' has been replicated to 'SC-3'.

3.2 Classification task

In this section, we present a distributed classification approach of the proposed SSK-DDoS for identifying various types of attacks: DDoS-DNS, DDoS-LDAP, DDoS-MSSQL, DDoS-NetBIOS, DDoS-UDP, and DDoS-SYN. The distributed classification approach is designed using the CICDDoS2019 dataset based on four distributed machine learning algorithms from Spark MLlib library: DecisionTreeClassifier (DTC), Naive Bayes (NB), Multinomial Logistic Regression (MLR), and Random Forest (RF). The Spark MLlib library provides an RF classifier algorithm for both binary and multiclass classification. It allows distributed designing of the model with millions or even billions of samples. The RF is an ensemble classifier that consists of multiple trees (classifiers), and each tree process is based different set of features. Gradient-Boosted Trees (GBT) is also an ensemble classifier and helps to improve accuracy. However, the Spark MLlib library provides this algorithm only for binary classification, and for this use-case, our classification approach has seven target classes. Therefore, this algorithm will not work for our use-case. We deployed an RF-based classification approach on the 'SC-3' for classifying flows into seven classes: Benign (One), DDoS_DNS (Two), DDoS_LDAP (Three), DDoS-MSSQL (Four), DDoS-NetBIOS (Five), DDoS-UDP (Six), and DDoS-SYN (Seven).

The primary objective of this classification approach is to classify network flows in real-time. We split the proposed classification approach into two parts: (i) Design process of a distributed classification model using distributed Spark MLlib library on the Hadoop cluster and (ii) After deployment of the classification model in 'SC-3' Spark Streaming cluster to classify network flows in real-time. The step-by-step workflow of the proposed classification model is presented in Figs. 5 (designing process) and 6 (after deployment process).

We divided this section into three sub-sections: details of the CICDDoS2019 dataset, designing and after deployment process of the classification model.

Table 1 Description of formulated features

Selected variables	Formulated features	Description
Max_Packet_Length		
Min_Packet_Length		
Fwd_Packet_Length_Min		
Fwd_Packet_Length_Max		
Average_Packet_Size	(1) Packet_statistics_data	Formulate new feature "Packet_statistics_data"
Packet Length Std		using nine packet relevant variables
Fwd_Packet_Length_Std		
Total_length_Fwd_Packets		
Fwd_packets		
Flow_IAT_Min		
Flow_IAT_Mean		
Flow_IAT_Max	(2) Flow_IAT_statistical_data	Formulate new feature "Flow_IAT_statistical_data"
Fwd_IAT_Total		using six IAT (Inter-Arrival Time) related attributes
Fwd_IAT_Mean		
Fwd_IAT_Max		
Pick up as selected one		
Min_seq_size_forward	(3) Min_seq_size_forward	Minimum segment size in the forward direction
Subflow_fwd_bytes	(4) Subflow_fwd_bytes	Avg. number of bytes in a sub-flow in the forward direction
Destination_port	(5) Destination_port	Port number which receives packets
ACK_flag_count	(6) ACK flag count	Number of packets with ACK
Init_win_bytes_forward	(7) Init_win_bytes_forward	Number of bytes initially in the window in the forward direction
Fwd_header_length	(8) Fwd_header_length	Header length of forwarded packets
Protocol	(9) Protocol	Protocol of the flow
Flow_duration	(10) Flow_duration	Duration of flow in microseconds

3.2.1 CICDDoS2019 dataset

The CICDDoS2019 [56] dataset is a collective project of the "Canadian Communications Security Establishment (CSE) and Canadian Institute for Cybersecurity (CIC)". It includes both benign and various types of DDoS attack scenarios. This dataset is available in both PCAP and CSV files i.e., raw packets and network flow with labeling, respectively. However, CSV files have several issues. Therefore, we generated network flows from PCAP files for various scenarios such as DDoS-UDP, DDoS-LDAP, DDoS-DNS, DDoS-SYN, DDoS-MSSQL, DDoS-NetBIOS, and Benign using the CICFlowMeter flow generator tool. The newly generated network flows contain 83 variables and one label column that we have to update as per the attack-wise schedule of PCAP files given on the dataset portal.

3.2.2 SSK-DDoS: design process

The step-by-step process to implement a distributed classification model for DDoS attacks using MLlib library is shown in Fig. 5. For designing this model, we assembled PCAP files of DDoS-UDP, DDoS-LDAP, DDoS-DNS, DDoS-SYN, DDoS-MSSQL, DDoS-NetBIOS, and Benign. The number of flows in each class is Benign: 56863, DDoS-DNS: 5071011, DDoS-LDAP: 2179930, DDoS-MSSQL: 4522492, DDoS-NetBIOS: 4093279, DDoS-UDP: 3134645, and DDoS-SYN: 1582289.

However, the number of flows in each class is highly-imbalanced which affects the accuracy of the classification model. We up-sampled some classes to 5071011. Therefore, the number of flows in the sample is 35 million+ and are stored in the HDFS. The next step is to implement a distributed classification model of DDoS attacks. We designed this classification model using Spark MLib machine learning-based algorithms: DTC, MLR, NB, and RF. Then deploy this model on the Spark Streaming cluster. The next task is to calculate performance evaluation metrics: precision, recall, and f1-score. The performance evaluation of these algorithms is discussed in Sect. 5. Finally, we save this model in the persistent storage for deploying in the 'SC-3' Spark Streaming cluster to analyze flows in real-time.

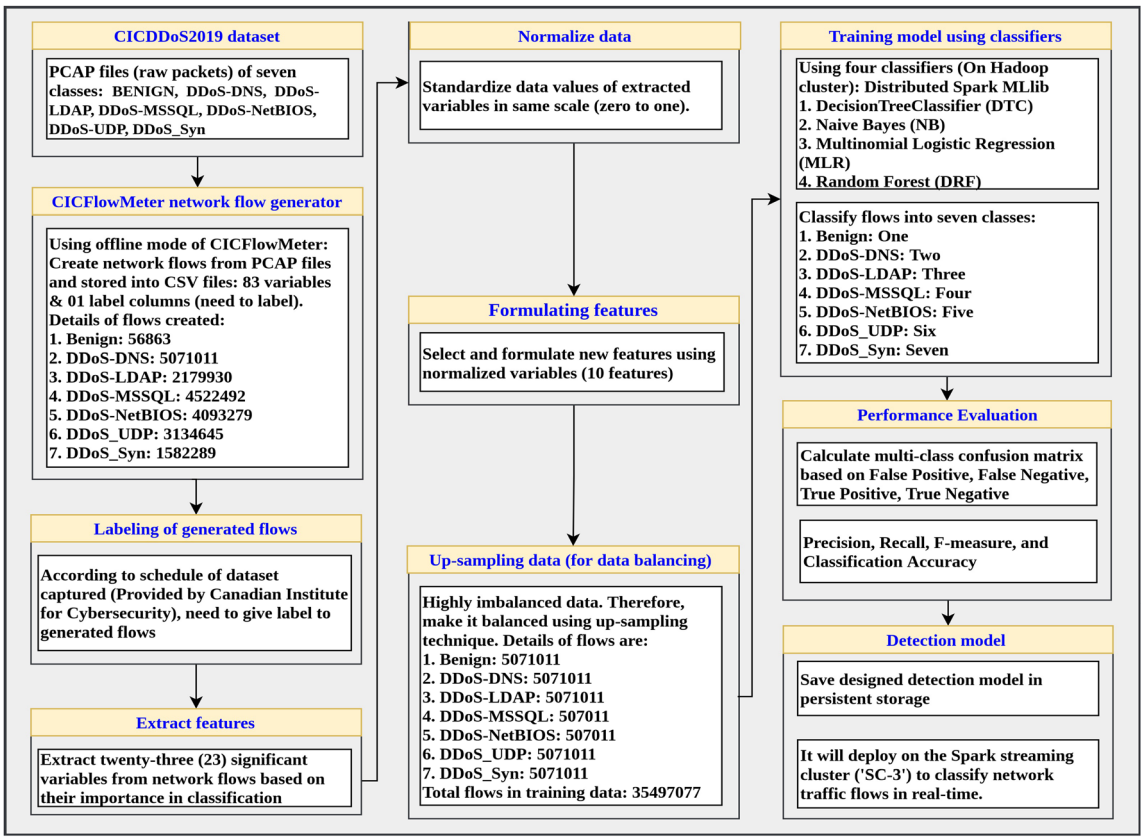

Fig. 5 SSK-DDoS classification model: design flow

3.2.3 K-DDoS: classification process in real-time (after deployment)

The second part of the classification approach is to classify incoming network traffic into seven classes. Figure 6 shows step-by-step process of the proposed classification approach after deploying in 'SC-3'. The CICFlowMeter generates network flows from incoming network traffic. Then, producer agents continuously publish created flows in the "ssk_ddos_flows". Both 'SC-1' and 'SC-2' immediately consume published flows and select twenty-three variables from the list of eighty-three variables. The next step is to scaling data values of variables, formulate features using scaled variables, and published them on the "ssk_ddos_features" by 'SC-1' and 'SC2'. The next step, distributed classification model immediately consumed messages from the "ssk_ddos_features", analyze and classify them into seven classes: DDoS-UDP, DDoS-LDAP, DDoS-DNS, DDoS-SYN, DDoS-MSSQL, DDoS-NetBIOS, and Benign. Finally, the proposed classification approach publishes the predicted class on the "ssk_ddos_prediction" topic to take immediate action on incoming network flows. Further, distributed SSK-DDoS classification system combines formulated features with the predicted result of each network flows and stores them in the HDFS with the help of the "ssk_ddos_retrain_data".

4 Experimental setup

In this section, we explore the experimental setup of the proposed distributed SSK-DDoS classification system for DDoS attacks. It is shown in Fig. 7. For the design and validation of the proposed SSK-DDoS, we consider two source networks, two ISPs in the intermediate network, and one victim network. Each ISP receives the network traffic from the source network, then generates network flows using CICFlowMeter from incoming traffic, selects essential variables, scales selected variables, formulate features using scaled variables, and replicates features in the 'SC-3'. The information about networks/clusters/nodes is given in the following:

- Two source networks: Legitimate and DDoS attack traffic traced towards victim network via ISPs.
- Two ISP networks: In each ISP network, deploy two nodes Spark Streaming cluster ('SC-1' and 'SC-2') for performing preprocessing task on incoming network traffic.

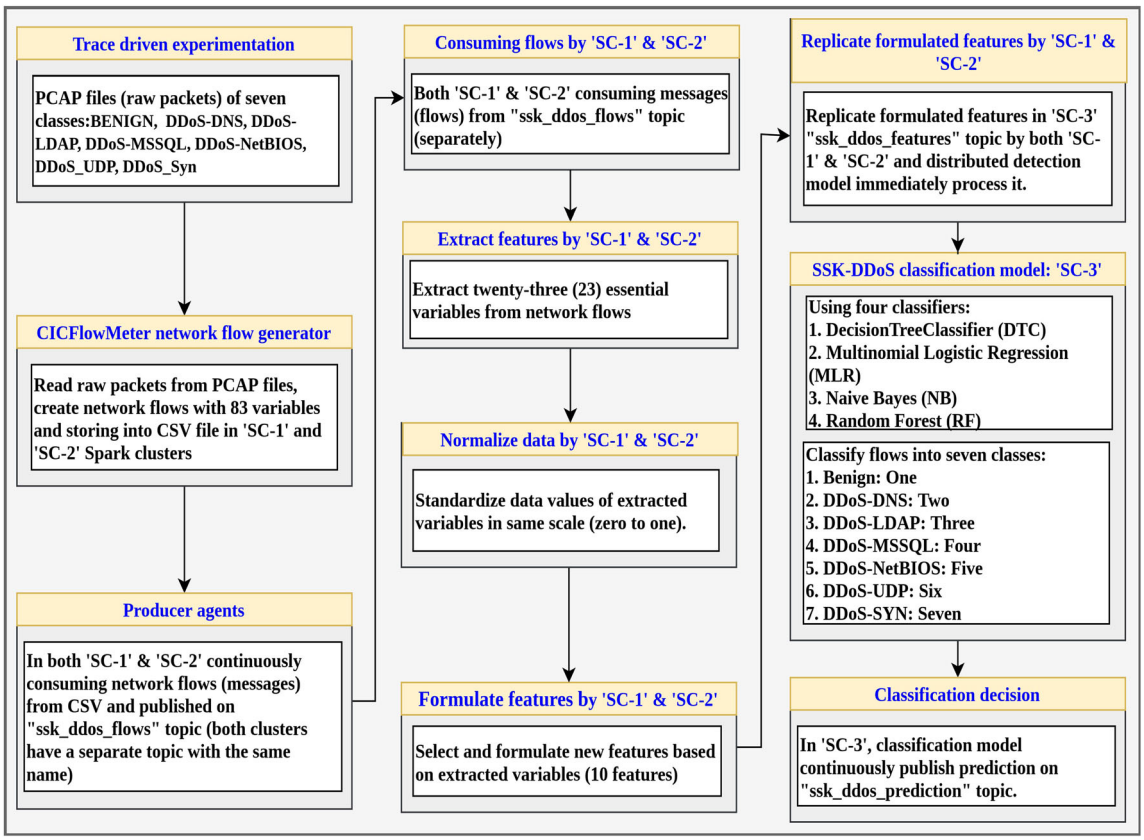

Fig. 6 SSK-DDoS classification model: after deployment flow

– Hadoop cluster: Deploy two nodes Hadoop cluster for storing formulated features with the predicted class of each network flow and retrain the existing model using a new set of samples.
– Spark Streaming cluster ('SC-3'): Implement two nodes Spark Streaming cluster 'SC-3' in the victim network to classify network flows in real-time.

Several Kafka topics have been created for publishing and consuming messages independently based on the distributed publish-subscribe messaging system. In 'SC-1' and 'SC-2' Spark Streaming clusters, 02 topics are created:

1. "ssk_ddos_flows": for publishing network flows created by CICFlowMeter.
2. "ssk_ddos_features": for publishing formulated features and replicated them to 'SC-3'.

Further, in the 'SC-3' Spark Streaming cluster, three Kafka topics are created:

1. "ssk_ddos_features": classification model immediately consumes features from this topic to classify flows in real-time.
2. "ssk_ddos_prediction": for publishing predicted class of the flows to take action.
3. "ssk_ddos_retrain_data": for publishing formulated features with predicted class of each flow to store in the HDFS.

5 Results and discussion

In this section, we evaluate the performance of our proposed SSK-DDoS classification system of DDoS attacks. The proposed SSK-DDoS classification system classifies network flows into seven classes.

We considered two cases for performance evaluation of the proposed SSK-DDoS classification system: case (I) While designing the classification model of DDoS attacks and case (II) After deployment of this classification model on DSPF i.e., Spark Streaming. For this, we measure three performance evaluation metrics for multi-class classification. The mathematical definition of these metrics for multi-class (in this use-case, seven target classes) classification: Precision (P_{m_class}), Recall (R_{m_class}), and F1-score ($F1S_{m\_class}$) are given in the following:

1. $P_{m\_class} = \frac{\sum_{i=1}^{n} \frac{TruePositive_i}{(TruePositive_i + FalsePositive_i)}}{n}$, *where n = number of classes (in this use-case, five classes)*

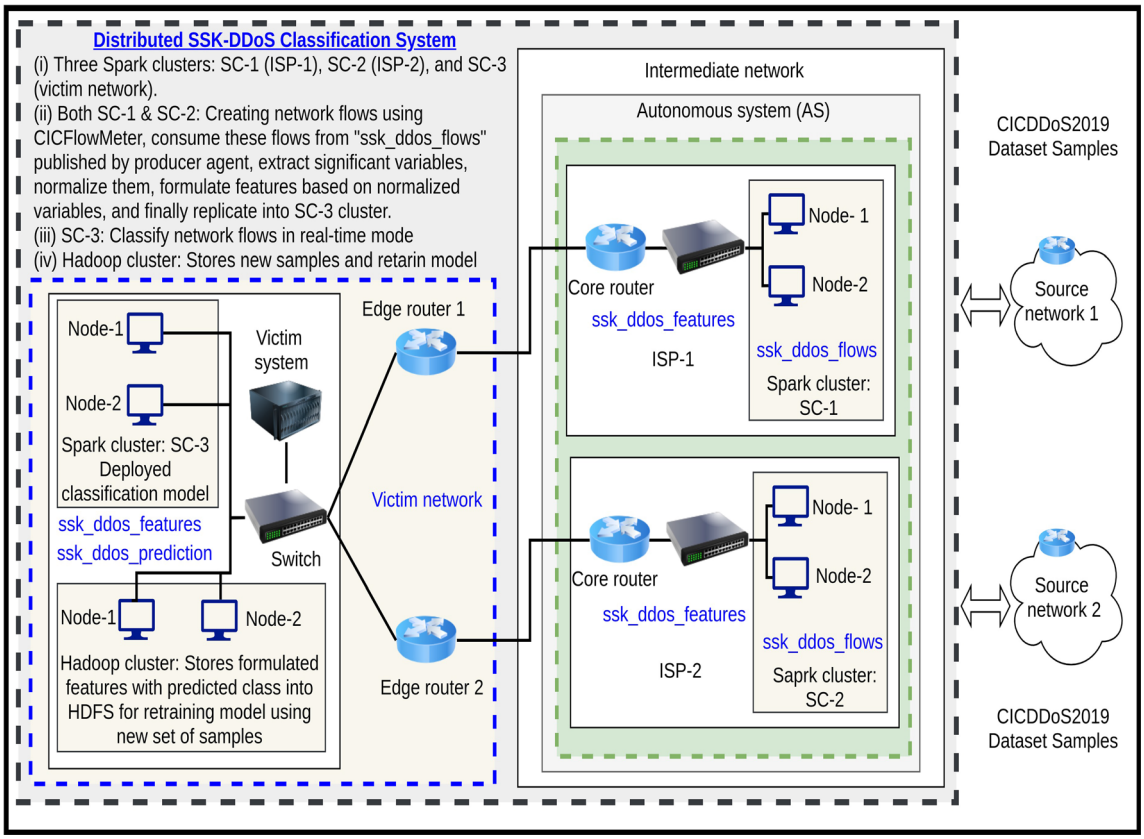

Fig. 7 Testbed for the proposed SSK-DDoS classification system for DDoS attacks

2. $R_{m\_class} = \frac{\sum_{i=1}^{n} \frac{TruePositive_i}{(TruePositive_i + FalseNegative_i)}}{n}$
3. $F1S_{m\_class} = \frac{2 * P_{m\_class} * R_{m\_class}}{(P_{m\_class} + R_{m\_class})}$

We designed and validated the proposed classification model using the CICDDoS2019 dataset. For evaluation of case-I, the description of class-wise network flows is given in Table 2. We designed this model using four Spark MLlib machine learning algorithms: DTC, MLR, NB, and RF. We visualized multiclass confusion matrices in Fig. 8 and evaluation metrics in Table 3. According to the accuracy, RF (89.05%) has given a better accuracy than the other three, i.e., MLR (43.28%) NB (69.39%) and DTC (87.61%). Further, we have tuned the number of trees ($T = 10, 20, 50$) parameter for the RF algorithm. We come across that RF gives better accuracy for $T = 50$ (89.05%) than $T = 10$ (87.89%) and $T = 10$ (87.91%).

For evaluation of the case-II, we examined six scenarios with different combinations of the CICDDoS2019 dataset classes. The description of each scenario is presented in

Table 2 Details of the CICDDoS2019 dataset for case-I

Traffic classes	No. of flows	Up-sampled/training flows	Testing flows	No. of flows correctly classified			
				RF	MLR	DTC	NB
Benign	56,863	5,071,011	1,672,499	1,672,499	1,377,530	1,672,274	536,291
DDoS-DNS	5,071,011	5,071,011	1,674,678	951,735	5	495,624	1261
DDoS-LDAP	2,179,930	5,071,011	1,673,339	1,219,812	0	1,524,730	1,593,745
DDoS-MSSQL	4,522,492	5,071,011	1,674,088	1,592,989	704,263	1,610,609	1,412,164
DDoS-NetBIOS	4,093,279	5,071,011	1,674,137	1,662,831	630,440	1,649,041	1,348,814
DDoS-UDP	3,134,645	5,071,011	1,673,164	1,659,647	698,027	1,639,431	862,807
DDoS-SYN	1,582,289	5,071,011	1,672,131	1,672,059	1,660,461	1,671,871	1,671,178
Total	20,640,509	35,497,077	11,714,036	10,431,569	5,070,726	10,263,580	7,426,260

DTC DecisionTreeClassifier, *MLR* multinomial logistic regression, *NB* naive Bayes, *RF* Random Forest

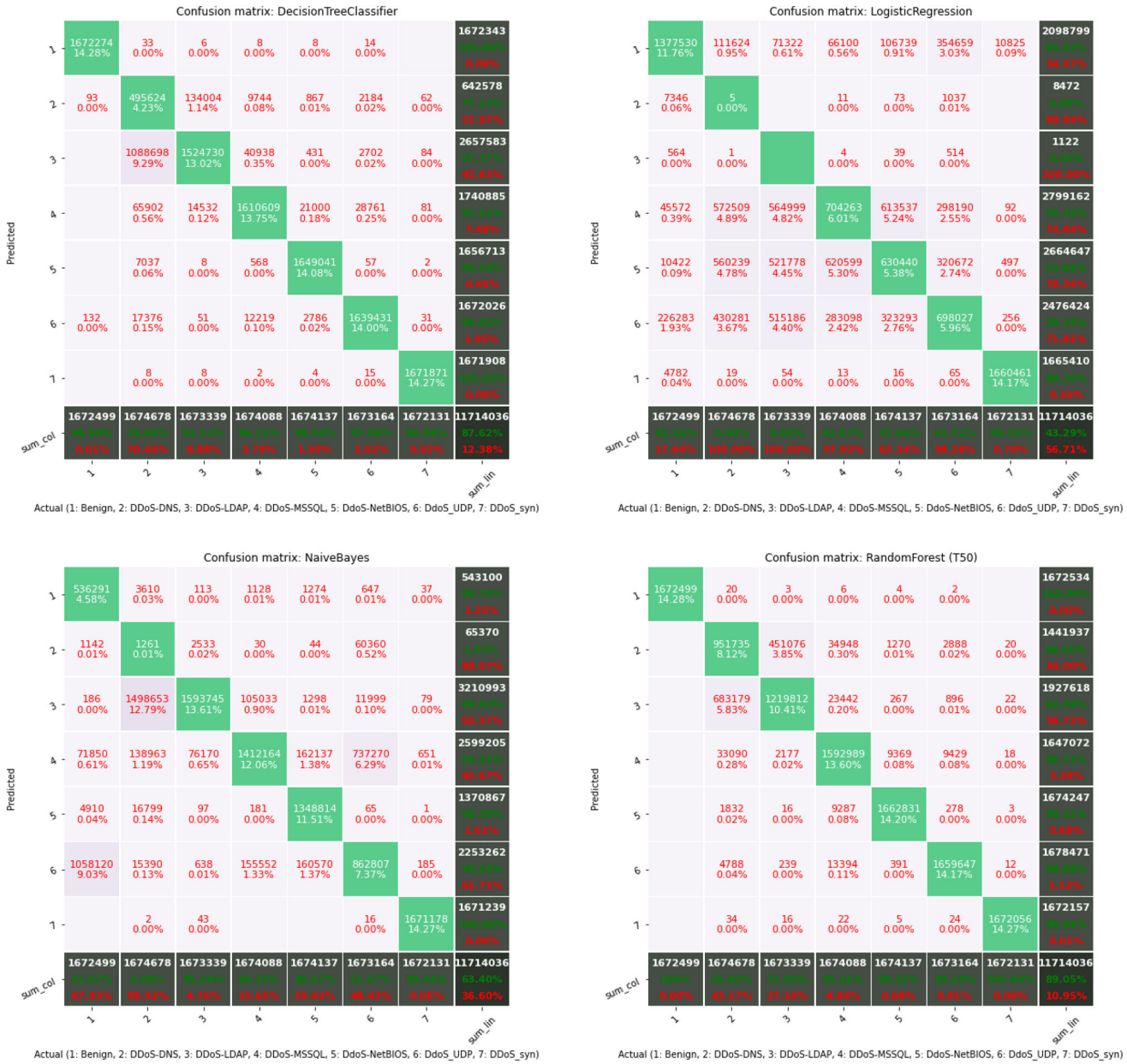

Fig. 8 Multi-class confusion matrices for Case-I (while designing a distributed model)

Table 4. After designing the classification model using various algorithms, the RF-based classification model ($T = 50$) has given better classification accuracy than MLR, NB, RF ($T = 10$), RF ($T = 20$), and DTC algorithms. Therefore, we deployed the RF-based classification model ($T = 50$) on the 'SC-3' Spark Streaming cluster in the production environment. The performance evaluation of these six scenarios is given in Table 5 and visualized their multi-class confusion matrices in Fig. 9.

From the performance evaluation of the proposed SSK-DDoS for case-II, the RF-based classification model ($T = 50$) provides a better accuracy such as scenario-I: 99.44%, scenario-II: 87.09%, scenario-III: 91.04%, scenario-IV: 99.17%, scenario-V: 92.17%, and scenario-VI: 94.42%. From this, we conclude that the proposed classification model gives 87%+ accuracy even attackers launch different types of attacks concurrently on the victim system.

Table 3 Performance of SSK-DDoS for Case-I (while designing a distributed model using MLlib)

Classifier	Metrics	Target classes (while designing a model)						
		Benign	DDoS-DNS	DDoS-LDAP	DDoS-MSSQL	DDoS-NetBIOS	DDoS-UDP	DDoS-SYN
RF	Precision	1.00	0.66	0.63	0.97	099	0.99	1.00
	Recall	1.00	0.57	0.73	0.95	0.99	0.99	1.00
	F-1 score	1.00	0.61	0.68	0.96	0.99	0.99	1.00
	Average classification accuracy: 89.05%							
MLR	Precision	0.66	0.00	0.00	0.25	0.24	0.28	1.00
	Recall	0.82	0.00	0.00	0.42	0.38	0.42	0.99
	F-1 score	0.73	0.00	0.00	0.31	0.29	0.34	1.00
	Average classification accuracy: 43.28%							
DTC	Precision	1.00	0.77	0.57	0.93	1.00	0.98	1.00
	Recall	1.00	0.30	0.91	0.96	0.99	0.98	1.00
	F-1 score	1.00	0.43	0.70	0.94	0.99	0.98	1.00
	Average classification accuracy: 87.61%							
NB	Precision	0.99	0.02	0.50	0.54	0.98	0.38	1.00
	Recall	0.32	0.00	0.95	0.84	0.81	0.52	1.00
	F-1 score	0.48	0.00	0.65	0.66	0.89	0.44	1.00
	Average classification accuracy: 63.39%							

5.1 Complexity analysis

In the case of the traditional framework-based DDoS attack detection mechanisms, each network flows is analyzed at a single point. Therefore, the time complexity of the system is $O(NNF)$, where NNF is the number of network flows analyzed by the system [63]. However, in the case of DPF/DSPF, the network flows analysis task is distributed between multiple nodes, and hence complexity is also distributed, say n (where n: no. of nodes). To measure the complexity of the proposed system, we assume each node equally examined network flows. Therefore, the complexity of DPF/DSPF is $O(\frac{NNF}{n})$. In this case, we have to measure one more parameter that is intermediate communication cost between nodes. Let us assume intermediate communication cost is $O(ICC)$. Therefore, the combined complexity cost (CCC) of the DPF/DSPF is $CCC = O(\frac{NNF}{n}) + O(ICC)$. However, DPF/DSPF is specially designed to analyze a large amount of data and hence $O(ICC)$ is negligible when we compared $O(NNF)$ with $O(ICC)$. Therefore the CCC of the DPF/DSPF-based DDoS attack detection system is $O(\frac{NNF}{n})$. It shows that the time complexity will go down as increasing nodes in the cluster.

5.2 Comparison with existing systems

In this section, we systematically compared of the proposed SSK-DDoS classification system of DDoS attacks with existing DPF and traditional framework based systems [34, 35, 37–39, 41–45, 47, 47–49, 57] in Tables 6 and 7.

Most of the DPF-based classification approaches [34, 35, 37–39, 44, 45, 47, 47, 48] of DDoS attacks and legitimate traffic are deployed on the Apache Hadoop framework. This type of approach efficiently handles a large number of flows on a cluster of nodes. However, Apache Hadoop is particularly employed to examine large data in offline mode. Therefore, this type of classification approach is not capable to classify network packets in real-time.

Few [41–43, 49, 57] authors have proposed Apache Spark-based classification approaches for DDoS attacks and legitimate traffic. This type of approach examines network flows in near to real-time. Further, these systems didn't provide an automated way to take action on incoming traffic flows. However, the proposed SSK-DDoS classification approach for DDoS attacks is not only designed on DPF (Using Spark MLlib machine learning library on Hadoop cluster) but also deployed on DSPF

Table 4 CICDDoS2019 dataset network flows details for Case-II (After deployment)

Scenario	Classifier (Deployed)	Classes	Predicting flows	Flows correctly predicted
Scenario-I	RF ($T = 50$)	Benign (1)	56,863	56,863
		DDoS-UDP (6)	3,134,645	3,108,124
		DDoS-SYN (7)	1,582,289	1,582,223
		Benign (1)	56,863	56,863
		DDoS-DNS (2)	5,071,011	3,268,262
		DDoS-LDAP (3)	2,179,930	1,588,788
Scenario-II	RF ($T = 50$)	DDoS-MSSQL (4)	4,522,492	4,304,647
		DDoS-NetBIOS (5)	4,093,279	4,065,444
		DDoS-UDP (6)	3,134,645	3,109,531
		DDoS-SYN (7)	1,582,289	1,582,223
		Benign (1)	56,863	56,863
Scenario-III	RF ($T = 50$)	DDoS-LDAP (3)	2,179,930	1,582,075
		DDoS-UDP (6)	3,134,645	3,109,528
		DDoS-SYN (7)	1,582,289	1,582,223
Scenario-IV	RF ($T = 50$)	Benign (1)	56,863	56,863
		DDoS-UDP (6)	3,134,645	3,108,124
		Benign (1)	56,863	56,863
		DDoS-LDAP (3)	2,179,930	1,582,075
Scenario-V	RF ($T = 50$)	DDoS-MSSQL (4)	4,522,492	4,304,051
		DDoS-UDP (6)	3,134,645	3,109,528
		DDoS-SYN (7)	1,582,289	1,582,223
		Benign (1)	56,863	56,863
		DDoS-LDAP (3)	2,179,930	1,582,075
Scenario-VI	RF ($T = 50$)	DDoS-MSSQL (4)	4,522,492	4,304,051
		DDoS-NetBIOS (5)	4,093,279	4,065,430
		DDoS-UDP (6)	3,134,645	3,109,528
		DDoS-SYN (7)	1,582,289	1,582,223

(Spark Streaming). Therefore, the proposed system provides a high-scalability feature. Further, we used Kafka's distributed pub-sub messaging system that will help to provide a loosely-coupled and automated-way to the proposed SSK-DDoS classification system for DDoS attacks.

Sharafaldin et al. [56] have generated a realistic dataset by considering various attack scenarios. Further, they have proposed a detection approach to classify different types of DDoS attacks. According to their performance evaluation, precision values for classifiers ID3, RF, NB, and LR is 0.78, 0.77, 0.41, and 0.25, respectively. While our RF-based classification model has given a better precision value (0.89).

Table 5 Performance of SSK-DDoS for Case-II (After deployment)

Scenarios	Metrics	Target classes (While designing a model)						
		Benign	DDoS-DNS	DDoS-LDAP	DDoS-MSSQL	DDoS-NetBIOS	DDoS-UDP	DDoS-SYN
Scenario-I	Precision	1.00	0.00	0.00	0.00	0.00	1.00	1.00
	Recall	1.00	0.00	0.00	0.00	0.00	0.99	1.00
	F-1 score	1.00	0.00	0.00	0.00	0.00	1.00	1.00
	Average classification accuracy: 99.44%							
Scenario-II	Precision	1.00	0.83	0.47	0.98	0.99	0.99	1.00
	Recall	1.00	0.64	0.73	0.95	0.99	0.99	1.00
	F-1 score	1.00	0.72	0.57	0.97	0.99	0.99	1.00
	Average classification accuracy: 87.09%							
Scenario-III	Precision	1.00	0.00	1.00	0.00	0.00	1.00	1.00
	Recall	1.00	0.00	0.73	0.00	0.00	0.99	1.00
	F-1 score	1.00	0.00	0.84	0.00	0.00	1.00	1.00
	Average classification accuracy: 91.04%							
Scenario-IV	Precision	1.00	0.00	0.00	0.00	0.00	1.00	0.00
	Recall	1.00	0.00	0.00	0.00	0.00	0.99	0.00
	F-1 score	1.00	0.00	0.00	0.00	0.00	1.00	0.00
	Average classification accuracy: 99.17%							
Scenario-V	Precision	1.00	0.00	0.96	1.00	0.00	0.99	1.00
	Recall	1.00	0.00	0.73	0.95	0.00	0.99	1.00
	F-1 score	1.00	0.00	0.83	0.97	0.00	0.99	1.00
	Average classification accuracy: 92.67%							
Scenario-VI	Precision	1.00	0.00	0.96	0.99	0.99	0.99	1.00
	Recall	1.00	0.00	0.73	0.95	0.99	0.99	1.00
	F-1 score	1.00	0.00	0.83	0.97	0.99	0.99	1.00
	Average classification accuracy: 94.42%							

Details of each scenario: Scenario-I Benign, DDoS-UDP, & DDoS-SYN, Scenario-II Benign, DDoS-DNS, DDoS-LDAP, DDoS-MSSQL,DDoS-NetBIOS, DDoS-UDP, & DDoS-SYN, Scenario-III Benign, DDoS-LDAP, DDoS-UDP, & DDoS-SYN, Scenario-IV Benign & DDoS-UDP, Scenario-V Benign, DDoS-LDAP, DDoS-MSSQL, DDoS-UDP, & DDoS-SYN, Scenario-V Benign, DDoS-LDAP, DDoS-MSSQL, DDoS-NetBIOS, DDoS-UDP, & DDoS-SYN

6 Conclusions

A distributed denial of service attack is one of the biggest threats to Internet-based services and their resources. It overwhelms victim resources in a short time by sending a large number of network packets. The traditional framework-based approaches themselves become a victim of attacks while classifying a massive amount of network flows. Further, most of the existing DPF-based classification systems for DDoS attacks were specially designed for offline mode and hence not capable to classify network flows in real-time.

This paper proposed Spark Streaming and Kafka-based distributed classification system for DDoS attacks, named by SSK-DDoS. This classification approach is designed using a distributed Spark MLlib machine learning library on a Hadoop cluster and deployed on the Spark streaming platform to classify the network traffic in real-time into seven classes: Benign, DDoS-DNS, DDoS-LDAP, DDoS-MSSQL, DDoS-NetBIOS, DDoS-UDP, and DDoS-SYN. Further, this system stored formulated features with the predicted class of each flow into the HDFS for retraining the existing distributed classification model using a new set of samples. The proposed SSK-DDoS classification system has been validated using the recent CICDDoS2019 dataset. The results show that the proposed SSK-DDoS detection system efficiently (89.05%) classified network traffic into seven classes.

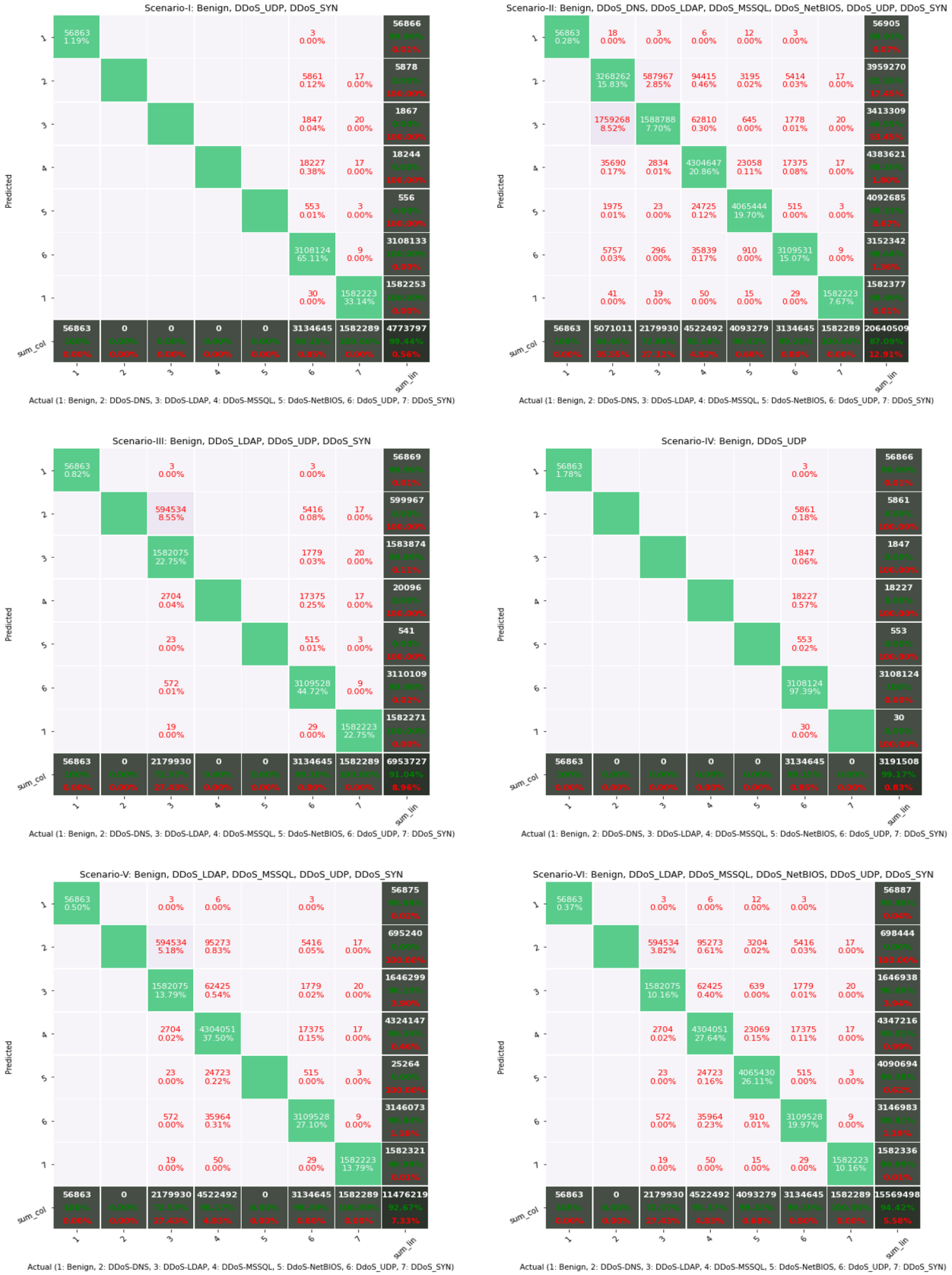

Fig. 9 Multi-class confusion matrices for Case-II (After deployment)

Table 6 Comparison of SSK-DDoS with existing DPF/DSPF-based approaches

System/ref. no.	Deployed on	Public dataset used	DDoS attacks	Retrain model	Real-time analysis	Real-time response	Deployment type
[34]	Hadoop	–	✔	✗	✗	✗	Victim-based
[35]	Hadoop	–	✔	✗	✗	✗	Victim-based
[37]	Hadoop	–	✔	✗	✗	✗	Victim-based
[57]	Spark	–	✔	✗	✔	✗	Victim-based
HADEC [38, 39]	Hadoop	–	✔	✗	✗	✗	Victim-based
[44]	Hadoop	–	✔	✗	✗	✗	Victim-based
[41, 42]	Spark	–	✔	✗	✗	✗	Victim-based
[45]	Hadoop	CAIDA	✔	✗	✗	✗	Victim-based
E-Had [46]	Hadoop	CAIDA, MIT Lincoln, FIFA98	✔	✗	✗	✗	Victim-based
[48]	Hadoop	–	✔	✗	✗	✗	Victim-based
[47]	Hadoop	CAIDA, MIT Lincoln	✔	✗	✗	✗	Victim-based
S-DDoS [49]	Spark,Hadoop	–	✔	✗	✔	✗	Victim-based
SAD-F [53]	Hadoop,Spark	UNSW- NB-15	✔	✗	✔	✗	Victim-based
SSK-DDoS (Proposed)	Spark Streaming Hadoop, Kafka	CICDDoS2019	✔	✔	✔	✔	Hybrid

Table 7 Comparison of SSK-DDoS with the traditional framework-based approaches

System	DDoS attacks	Retrain model	Real-time analysis	Real-time response	Handle massive data (traces)
[58]	✔	✗	✗	✗	✗
D-FAC [59]	✔	✗	✗	✗	✗
[2]	✔	✗	✗	✗	✗
[60]	✔	✗	✗	✗	✗
[61]	✔	✗	✗	✗	✗
TIDS [62]	✔	✗	✗	✗	✗
SSK-DDoS (proposed)	✔	✔	✔	✔	✔

Data availability Data available in a public (UNB-Canadian Institute for Cybersecurity, CICDDoS2019) repository that issues datasets with DOIs (https://www.unb.ca/cic/datasets/ddos-2019.html)

Declarations

Conflict of interest The authors declared that they have no conflict of interest.

References

1. Arivudainambi, D., Varun Kumar, K.A., Chakkaravarthy, S.S.: Lion IDS: a meta-heuristics approach to detect DDOS attacks against software-defined networks. Neural Comput. Appl. **31**(5), 1491–1501 (2019)
2. Gopi, R., Sathiyamoorthi, V., Selvakumar, S., Manikandan, R., Chatterjee, P., Jhanjhi, N., Luhach, A.K.: Enhanced method of ANN based model for detection of DDoS attacks on multimedia Internet of Things. Multimedia Tools Appl. (2021). https://doi.org/10.1007/s11042-021-10640-6

3. Behal, S., Kumar, K., Sachdeva, M.: D-FACE: an anomaly based distributed approach for early detection of DDoS attacks and flash events. J. Netw. Comput. Appl. **111**, 49–63 (2018)
4. Bhandari, A., Kumar, K., Sangal, A., Behal, S.: An anomaly based distributed detection system for DDoS attacks in Tier-2 ISP networks. J. Ambient Intell. Human. Comput. (2020). https://doi.org/10.1007/s12652-020-02208-3
5. Kaspersky: DoS attacks Q4-2020 (2021). https://securelist.com/ddos-attacks-in-q4-2020/100650/. Accessed 2 Mar 2021
6. Kaspersky: DDoS attacks Q3-2020 (2021). https://securelist.com/ddos-attacks-in-q3-2020/99171/. Accessed 2 Mar 2021
7. Apache Hadoop: https://hadoop.apache.org/. Accessed 10 Feb 2021
8. Bhardwaj, A., Singh, V.K., Narayan, Y.: Analyzing BigData with Hadoop cluster in HDInsight azure Cloud. In: Annual IEEE India Conference (INDICON), vol. 2015, pp. 1–5. IEEE (2015)
9. Apache Spark: https://spark.apache.org/. Accessed 10 Feb 2021
10. Chen, Y., He, F., Li, H., Zhang, D., Wu, Y.: A full migration BBO algorithm with enhanced population quality bounds for multimodal biomedical image registration. Appl. Soft Comput. **93**, 106335 (2020)
11. Quan, Q., He, F., Li, H.: A multi-phase blending method with incremental intensity for training detection networks. Vis. Comput. **37**(2), 245–259 (2021)
12. Zhang, S., He, F.: DRCDN: learning deep residual convolutional dehazing networks. Vis. Comput. **36**(9), 1797–1808 (2020)
13. Li, H., He, F., Chen, Y., Pan, Y.: MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution. Memetic Comput. **13**(1), 1–18 (2021)
14. Apache Kafka: https://kafka.apache.org/. Accessed 08 Feb 2021
15. Lashkari, A.H., Draper-Gil, G., Mamun, M.S.I., and Ghorbani, A.A.: Characterization of tor traffic using time based features. In: ICISSp, pp. 253–262 (2017)
16. Patil, N.V., RamaKrishna, C., Kumar, K.: Distributed frameworks for detecting distributed denial of service attacks: a comprehensive review, challenges and future directions. Concurr. Comput. Pract. Exp. **33**(10), e6197 (2021)
17. Mirkovic, J., Reiher, P.: A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Comput. Commun. Rev. **34**(2), 39–53 (2004)
18. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surveys Tutor. **15**(4), 2046–2069 (2013)
19. Manavi, M.T.: Defense mechanisms against distributed denial of service attacks: a survey. Comput. Electr. Eng. **72**, 26–38 (2018)
20. Peng, T., Leckie, C., Ramamohanarao, K.: Survey of network-based defense mechanisms countering the DoS or DDoS problems. ACM Comput. Surv. (CSUR) **39**(1), 3 (2007)
21. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**(1), 303–336 (2014)
22. Douligeris, C., Mitrokotsa, A.: DDoS attacks and defense mechanisms: classification and state-of-the-art. Comput. Netw. **44**(5), 643–666 (2004)
23. Hoque, N., Bhuyan, M.H., Baishya, R.C., Bhattacharyya, D.K., Kalita, J.K.: Network attacks: taxonomy, tools and systems. J. Netw. Comput. Appl. **40**, 307–324 (2014)
24. Lee, S.: Distributed denial of service: taxonomies of attacks, tools and countermeasures. In: Proceedings of the International Workshop on Security in Parallel and Distributed Systems, pp. 543–550 (2004)
25. Bhatia, S., Behal, S., Ahmed, I.: Distributed denial of service attacks and defense mechanisms: current landscape and future directions. In: Versatile Cybersecurity, pp. 55–97. Springer, Cham (2018)
26. Mahjabin, T., Xiao, Y., Sun, G., Jiang, W.: A survey of distributed denial-of-service attack, prevention, and mitigation techniques. Int. J. Distrib. Sensor Netw. **13**(12), 1550147717741463 (2017)
27. Behal, S., Kumar, K.: Characterization and comparison of DDoS attack tools and traffic generators: a review. IJ Netw. Security **19**(3), 383–393 (2017)
28. Elejla, O.E., Anbar, M., Belaton, B.: ICMPv6-based DoS and DDoS attacks defense mechanisms. IETE Tech. Rev. **34**(4), 390–407 (2017)
29. Fenil, E., Mohan Kumar, P.: Survey on DDoS defense mechanisms. Concurr. Comput. Pract. Exp. **32**(6), e5114 (2019)
30. Singh, J., Behal, S.: Detection and mitigation of DDoS attacks in SDN: a comprehensive review, research challenges and future directions. Comput. Sci. Rev. **37**, 100279 (2020)
31. Bouyeddou, B., Harrou, F., Kadri, B., Sun, Y.: Detecting network cyber-attacks using an integrated statistical approach. Clust. Comput. **24**(2), 1435–1453 (2021)
32. Maharaja, R., Iyer, P., Ye, Z.: A hybrid fog-cloud approach for securing the Internet of Things. Clust. Comput. **23**(2), 451–459 (2020)
33. Jyothsna, V., Prasad, K.M., Rajiv, K., Chandra, G.R.: Flow based anomaly intrusion detection system using ensemble classifier with feature impact scale. Clust. Comput. **24**(4), 1–18 (2021)
34. Lee, Y., Lee, Y.: Detecting DDoS attacks with Hadoop. In: Proceedings of the ACM CoNEXT Student Workshop, p. 7. ACM, New York (2011)
35. Khattak, R., Bano, S., Hussain, S., Anwar, Z.: DOFUR: DDoS Forensics Using MapReduce. In: Frontiers of Information Technology (FIT), vol. 2011, pp. 117–120. IEEE (2011)
36. Zhao, T., Lo, D.C.-T., Qian, K.: A neural-network based DDoS detection system using Hadoop and HBase. In: High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conference on Embedded Software and Systems (ICESS), pp. 1326–1331. IEEE (2015)
37. Dayama, R., Bhandare, A., Ganji, B., Narayankar, V.: Secured network from distributed DoS through Hadoop. Int. J. Comput. Appl. **118**(2), 20–22 (2015)
38. Hameed, S., Ali, U.: Efficacy of live DDoS detection with Hadoop. In: Network Operations and Management Symposium (NOMS), IEEE/IFIP, vol. 2016, pp. 488–494. IEEE (2016)
39. Hameed, S., Ali, U.: HADEC: a Hadoop based Live DDoS detection framework. EURASIP J. Inf. Security **2018**(1), 1–19 (2018)
40. Hsieh, C.-J., Chan, T.-Y.: Detection DDoS attacks based on neural-network using Apache Spark. In: 2016 International Conference on Applied System Innovation (ICASI), pp. 1–4. IEEE (2016)
41. Alsirhani, A., Sampalli, S., Bodorik, P.: DDoS attack detection system: utilizing classification algorithms with Apache Spark. In: 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–7. IEEE (2018)
42. Alsirhani, S., Sampalli, A., Bodorik, P.: DDoS detection system: utilizing gradient boosting algorithm and Apache Spark. In: 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), pp. 1–6. IEEE (2018)
43. Ahmad, S., Yasin, A., Shafi, Q.: DDoS attacks analysis in bigdata (Hadoop) environment. In: 2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 495–501. IEEE (2018)
44. Maheshwari, V., Bhatia, A., Kumar, K.: Faster detection and prediction of DDoS attacks using MapReduce and time series analysis. In: 2018 International Conference on Information Networking (ICOIN), pp. 556–561. IEEE (2018)

45. Chhabra, G.S., Singh, V., Singh, M.: Hadoop-based analytic framework for cyber forensics. Int. J. Commun. Syst. Wiley Online Library **31**(15), e3772 (2018)
46. Patil, N.V., Krishna, C.R., Kumar, K., Behal, S.: E-had: a distributed and collaborative detection framework for early detection of DDoS attacks. J. King Saud Univ. Comput. Inf. Sci. (2019). https://doi.org/10.1016/j.jksuci.2019.06.016
47. Patil, N.V., Krishna, C.R., Kumar, K., Behal, S.: Apache hadoop based distributed denial of service detection framework. In: Information, Communication and Computing Technology, pp. 25–35. Springer, Singapore (2019)
48. Sharma, A., Agrawal, C., Singh, A., Kumar, K.: Real-time DDoS detection based on entropy using Hadoop framework. In: Computer Engineering and Technology, pp. 297–305. Springer (2019)
49. Patil, N.V., Rama-Krishna, C., Kumar, K.: S-DDoS: Apache Spark based real-time DDoS detection system. J. Intell. Fuzzy Syst. **38**, 1–9 (2020)
50. Vani, Y.K., Ranjana, P.: Detection of distributed denial of service attack using DLMN algorithm in hadoop. J. Crit. Rev. **7**(11), 1011–1017 (2020)
51. Chen, L., Zhang, Y., Zhao, Q., Geng, G., Yan, Z.: Detection of dns ddos attacks with random forest algorithm on spark. Procedia Comput. Sci. **134**, 310–315 (2018)
52. Gumaste, S., Narayan, D., Shinde, S., Amit, K.: Detection of ddos attacks in openstack-based private cloud using apache spark. J. Telecommun. Inf. Technol. **4**, 62–71 (2020)
53. Ahmed, A., Hameed, S., Rafi, M., Mirza, Q.K.A.: An intelligent and time-efficient DDoS identification framework for real-time enterprise networks SAD-F: spark based anomaly detection framework. IEEE Access **8**, 219483–219502 (2020)
54. Jain, M., Kaur, G.: Distributed anomaly detection using concept drift detection based hybrid ensemble techniques in streamed network data. Clust. Comput. (2021). https://doi.org/10.1007/s10586-021-03249-9
55. Kshirsagar, D., Kumar, S.: A feature reduction based reflected and exploited DDoS attacks detection system. J. Ambient Intell. Human. Comput. (2021). https://doi.org/10.1007/s12652-021-02907-5
56. Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A.: Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1–8. IEEE (2019)
57. Han, D., Bi, K., Liu, H., Jia, J.: A DDoS attack detection system based on spark framework. Comput. Sci. Inf. Syst. **14**(3), 769–788 (2017)
58. Sree and Bhanu, S.M.S.: Detection of HTTP flooding attacks in cloud using fuzzy bat clustering. Neural Comput. Appl. (2019). https://doi.org/10.1007/S00521-019-04473-6
59. Behal, S., Kumar, K., Sachdeva, M.: D-FAC: a novel ϕ-divergence based distributed DDoS defense system. J. King Saud Univ. Comput. Inf. Sci. **33**(3), 291–303 (2018)
60. de Lima Filho, F.S., Silveira, F.A., de Medeiros Brito Junior, A., Vargas-Solar, G., Silveira, L.F.: Smart detection: an online approach for DoS/DDoS attack detection using machine learning. Security Commun. Netw. **2019**, 1574749 (2019)
61. Marvi, M., Arfeen, A., Uddin, R.: A generalized machine learning-based model for the detection of DDoS attacks. Int. J. Netw. Manag. **31**(6), e2152 (2020)
62. Joldzic, O., Djuric, Z., Vuletic, P.: A transparent and scalable anomaly-based DoS detection method. Comput. Netw. **104**, 27–42 (2016)
63. Brent, R.P., Zimmermann, P.: Modern Computer Arithmetic, vol. 18. Cambridge University Press, Cambridge (2010)

Nilesh Vishwasrao Patil has done Bachelor of Enginerring in Computer Engineering from NMU, Jalgaon in 2008. He finished his Masters in Computer Engineering from Savitribai Phule Pune University, Pune in 2015. He finished his Ph.D. from Department of Computer Science & Engineering, Panjab University, in 2021. His general research interests are in the areas of Information Security, Big Data Analytics and Computer Networks. He has published around 10+ research papers in different International Journals and Conferences of repute. He is reviewer of various journals of ACM and Springer.

C. Rama Krishna received B.Tech. from JNTU, Hyderabad, M.Tech., from Cochin University of Science & Technology, Cochin, and Ph.D. from IIT, Kharagpur. He is Senior Member, IEEE, USA. Since 1996, he is working as Professor with department of Computer Science and Engineering, National Institute of Technical Teachers Training and Research (NITTTR), Chandigarh. His areas of research interest include Computer Networks, Wireless Networks, Cryptography & Cyber Security, and Cloud Computing. To his credit, he has more than 100 research publications in referred International and National Journals and Conferences. He is reviewer of various journals of IEEE, ACM, Elsevier and Springer.

Krishan Kumar has done Bachelor of Technology in Computer Science and Engineering from National Institute of Technology, Hamirpur in 1995. He finished his Masters in Software Systems from BITS Pilani in 2001. He finished his Ph.D. from Department of Electronics and Computer Engineering at Indian Institute of Technology, Roorkee in 2008. His general research interests are in the areas of Information Security and Computer Networks. He has published around 200+ research papers in different International Journals and Conferences of repute. He is reviewer of various journals of IEEE, ACM, Elsevier and Springer.