2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)

Feasibility Evaluation of Compact Flow Features for Real-time DDoS Attacks Classifications

Muhammad Fajar Sidiq
Department of Informatics
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia
fajar@ittelkom-pwt.ac.id

Nanda Iryani
Department of Informatics
Institut Teknologi Telkom Purwokerto
Purwokerto, Indonesia
nanda@ittelkom-pwt.ac.id

Akbari Indra Basuki
Research Center for Data and Information Sciences
National Research and Innovation Agency
Bandung, Indonesia
akba002@brin.go.id

Arief Indriarto Haris
Research Center for Data and Information Sciences
National Research and Innovation Agency
Bandung, Indonesia
arie046@brin.go.id

Rd Angga Ferianda
Research Center for Data and Information Sciences
National Research and Innovation Agency
Bandung, Indonesia
rdan002@brin.go.id

***Abstract*—According to the research trend, training the distributed denial of services (DDoS) attacks classifier using network flow features will yield higher classification performances and efficiency than the per-packet-based approach. Nonetheless, the existing flow-based classifier uses bloated features and offline flow extraction that is not suitable for real-time DDoS protection. This study investigates the feasibility of compact flow features that can be directly extracted using a programmable switch for real-time DDoS attack classification. The proposed method considers only four flow features: IP protocols, packet counter, total byte counter, and the delta time of a network flow. The evaluation results on the CICDDoS2019 dataset showed a comparable classification performance to the works that use bloated features ($24 - 82$ features). The best result was achieved by the decision tree and the random forest classifier showing $\geq 89.5\%$ scores in accuracy, precision, recall, and F1 score. The proposed models can classify 10 out of 12 DDoS attacks correctly, failing only to discriminate between SSDP and UDP-based DDoS attacks. In addition, the trained classifier shows a better generalization ability by retaining similar performances on unseen 42.8 millions flow data while trained on ≤ 200 thousand flow data. At last, the proposed method is suitable for real-time application since it supports quick classification performance of up to 9.6 millions of flow inferring per second on the Decision Tree classifier.**

***Index Terms*—DDoS Attacks classification, Compact flow features, Software-defined networking, Real-time protection**

I. Introduction

Distributed denial of services (DDoS) attacks have a severe impact on internet infrastructure and cause multi-million dollars of losses to information technology businesses. The proper countermeasure for DDoS attacks is to scrub the attacks as soon as possible before they depleted the server and network resources. Massive works has been proposed to effectively counter DDoS attacks, from packet-based screening [1]–[11] to network flow level analysis [12]–[25]. The technique has evolved from simple threshold-based detection [6]–[8], [11] to entropy analysis [9], [10], and the use of machine learning [1], [5], [32], [33] and deep learning method [2]–[4], [34].

Despite its ability to mitigate DDoS attacks, most of the non-machine learning approach [6]–[11] are limited to detecting one kind of DDoS attack and unable to differentiate several types of attacks at once. This limits the defender's ability to provide a proper countermeasure to the attacks. The use of machine learning successfully addresses the classification problem as presented in [1]–[5]. Most works are trained and tested using the CICDDoS2019 dataset due to its vast coverage of modern DDoS attacks.

Nevertheless, the machine learning approach has a drawback in its workflow that makes it not suitable for real-time protection. The works use flow features extracted in an offline fashion using the CICFlowMeter tool [26]. Consequently, the classification cannot run in a real-time manner, since we must first log the packet and extract the flow features before the machine learning model can infer them accordingly.

This study aims to solve the aforementioned problem by proposing compact flow features that are extractable by networking devices, thus it can be inferred directly by the trained classifier in a real-time fashion. The compact feature preference is to compensate for the limited processing ability of network devices without incurring significant latency in packet forwarding. We propose four flow features that are commonly extracted by network switches: IP protocol, packet counter, total byte counter, and flow delta time. We use the CICDDoS2019 dataset for the evaluation and the Scapy tool to parse the compact flow features. The study compared the classification performance of the classifier trained using compact flow features with existing works that use numerous features. At last, we present the feasibility analysis for real-time classification of DDoS attacks by observing the classifiers inferring time.

We structure the paper by first presenting the system design, covering the technical requirement of the compact flow features and the design of the evaluation testbed. In section III, we present the evaluation that covers three main objectives: classification performance, generalization ability, and real-time feasibility. Finally, section IV concludes the paper.

978-1-6654-6030-9/22/$31.00 ©2022 IEEE

II. Proposed Method

A. Compact flow features

The minimalist selection of flow features aims to minimize extraction time by the switches without significantly affecting forwarding latency. We select four flow features that are commonly extracted by the software-defined networking (SDN) switches: IP protocols, packet counter, byte counter, and delta time. In OpenFlow SDN [27], the controller can gather the features by sending *OFPFlowStatsReply* command to the OpenFlow switches. In P4 SDN [28], the switch can extract the features using a programmable parser and save them in the switch's internal registers. The classifier can collect the features using in-band telemetry [29], [30] or using a pooling mechanism to carry the stored flow features using a carrier packet.

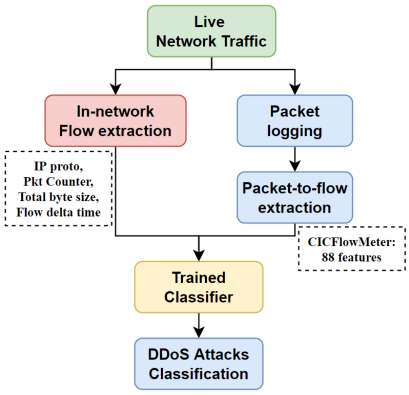


Fig. 1: Comparison of the proposed method (*left branch*) to the existing works (*right branch*)

The comparison between our proposed scheme to the existing work is shown in Figure 1. The right branch shows the existing works that use packet logging and offline flow parsing, thus prohibiting real-time flow classification. In real-time, the network device does not know whether a certain network packet belongs to a certain network flow. Consequently, the switch cannot directly drop the packet as the classifier only issues dropping commands at the flow level. The decision to drop the packet is only applicable after the packet has been logged and parsed to the flow data. This schema has two setbacks, first, it adds extra processing time for logging and flow parsing, and second, it needs extra storage to store the logged packet. Considering one flow data comprises multiple network packets, the space to store the logged packets is in different magnitudes than the flow data.

The SDN switches, both OpenFlow and P4 switches, can extract the flow information within the switch hardware. As a consequence, they know whether a certain packet belongs to a certain network flow or not. The switch can drop the packet directly if it belongs to the flow labeled as DDoS attacks. The remaining requirement to achieve real-time DDoS protection is to ensure that the classifier can infer the network flow as quickly as possible to handle massive concurrent flows during peak traffic.

TABLE I: Compact flow feature dataset

DDoS Attack	Number of flow data
DNS	14629
LDAP	2068826
MSSQL	3511790
NTP	1191398
NetBIOS	3935138
SNMP	4811793
SSDP	2583056
SYN	1715307
TFTP	19932309
UDP	3109787
UDP-lag	4128
WebDDoS	446

The final compact flow feature dataset is available on our GitHub page [31]. As a recap, the number for each DDoS attack is presented in table I.

B. Classifier model

Existing work uses a bloated number of flow features consisting of statistical features in addition to the basic flow data. This scheme requires an excellent classifier model and massive computation resources the train them. Several works [2]–[4], [34] intentionally use deep learning techniques to achieve better classification results. Despite the outstanding performances, the scheme is not practical for real-time DDoS filtering due to extensive computation time.

Our proposed scheme, in contrast, uses a compact number of 4 flow features to ease model training and quick classification to meet the real-time constraint. Considering the proposed flow data is quite minimalist, our scheme can be applied to lightweight machine learning models such as Random forests, Support vector machines, Decision trees, and others alike. These classifier models can be trained in a short time and can infer the flow in the sub-seconds band to meet the real-time properties. Based on the preliminary test, we limit the classifiers in our evaluation to four: XGBoost, Random Forest, Decision Tree, and K-nearest Neighbor (KKN).

C. Evaluation method

For evaluation, we use CICDDoS2019 [19] as the dataset considering two reasons. First, it has a wide coverage of modern DDoS attacks providing 12 kinds of them, either reflective or flooding ones. Second, it has been used in numerous works, thus it eases the comparison effort of our proposed scheme against similar works.

Due to limited testbed infrastructures, we opt for the emulation approach by mimicking in-network flow extraction using Scapy. The process commences by parsing the PCAP file into a flow-based session of TCP or UDP protocols. Next, we label the flow data based on the timing information [cicddospaper] to determine the type of DDoS attacks. Finally, we store only the four compact flow features and their label as the final compact dataset. We can reduce the total size of the dataset from ≈ 152 GB of per-packet data into ≈ 0.99 GB of compact flow data. Compared to CICFlowMeter generated flow data (≈ 20.7 GB), our compact flow features is $\approx 5\%$ of the fraction.

The evaluation phase aims to measure three main objectives as follows.

- Classification performances for DDoS attacks
  The first objective is to evaluate the classification performance of the four chosen classifiers (XGBoost, Random Forest, Decision Tree, and K-nearest neighbor) to determine the most minimalist training dataset. The minimalist classifier will be able to infer the flow data faster while preserving its classification performance. We use accuracy, precision, recall, and F1-score as the evaluation parameter.
- Generalizability of the Classifier
  This evaluation is to test the overfitting possibility of the training dataset. We run the trained classifier on all of the generated flow datasets. If the classifier is overfitted to the training data, the classifier performance will degrade significantly. A slight degradation of the performances indicates that the trained classifier is indeed able to classify DDoS attacks or generalizable for unseen flow data.
- Feasibility for real-time classification
  The last objective is to assess the feasibility of real-time protection by measuring resource utilization and the inferring time of the classifiers. The number of flow classifications per second determines the real-time applicability of the classifier.

III. Evaluation

A. Classification performances

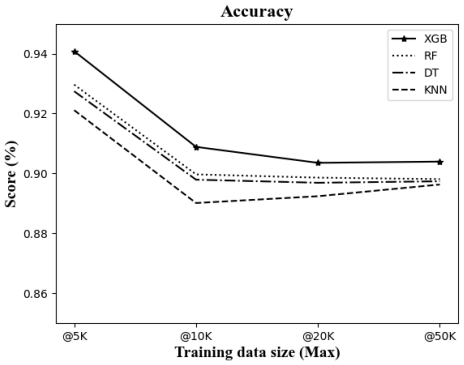
We evaluate the impact of using compact flow features by varying the size of training data. A small dataset usually produces better accuracy and precision due to the overfitting of the training data. Consequently, the trained model cannot be generalized to unseen flow data. Meanwhile, a bigger dataset will require more computation resources to train the model. It will slow down the inferring process and is not suitable for real-time classification.

Figure 2 shows the classification performances over four different sizes of training data. The training data is generated by randomly sampling the flow dataset based on the label information. A 5K training dataset means that we pick up 5000 flow data from the main dataset. The results show that with the exception of the KNN classifier, the classification performance is convergent at approximately $\geq 90\%$ score using 20 thousand samples per-DDoS attack.

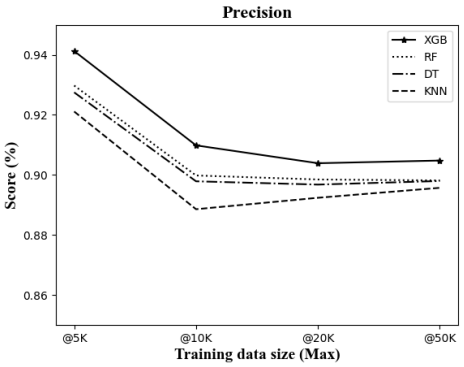
We compared our proposed scheme that uses a small number of flow features with existing work that uses tens of flow features [18], [19], [32]–[34]. Table II shows the comparison results. Even though our proposed method has a lower score in accuracy, precision, recall, and F1 scores, it can classify more DDoS attack types than the existing works. Our proposed methods only fail to differentiate between SSDP DDoS and UDP DDoS attacks.

B. Generalization performances

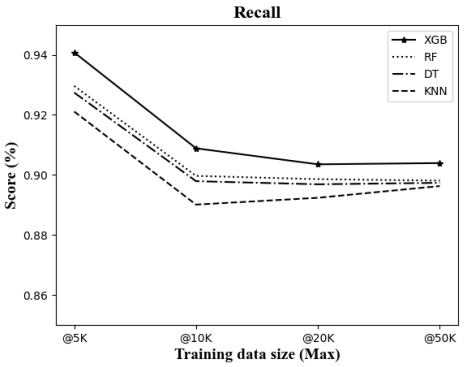
The next step after training the model is to test whether the model can be generalized to classify the unseen flow data. Figure 3 shows the confusion matrices for each classifier tested on the generated dataset. The compact dataset comprises 12 DDoS attacks with a total of ≈ 42.8 millions of flow data.



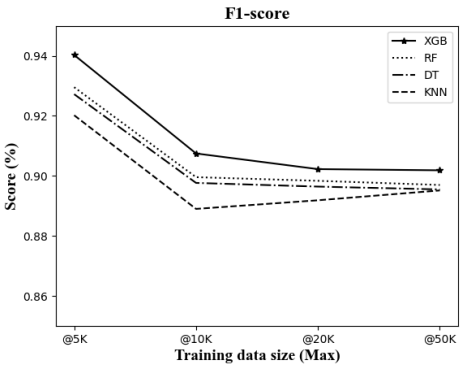
(a) Accuracy comparison



(b) Precision comparison

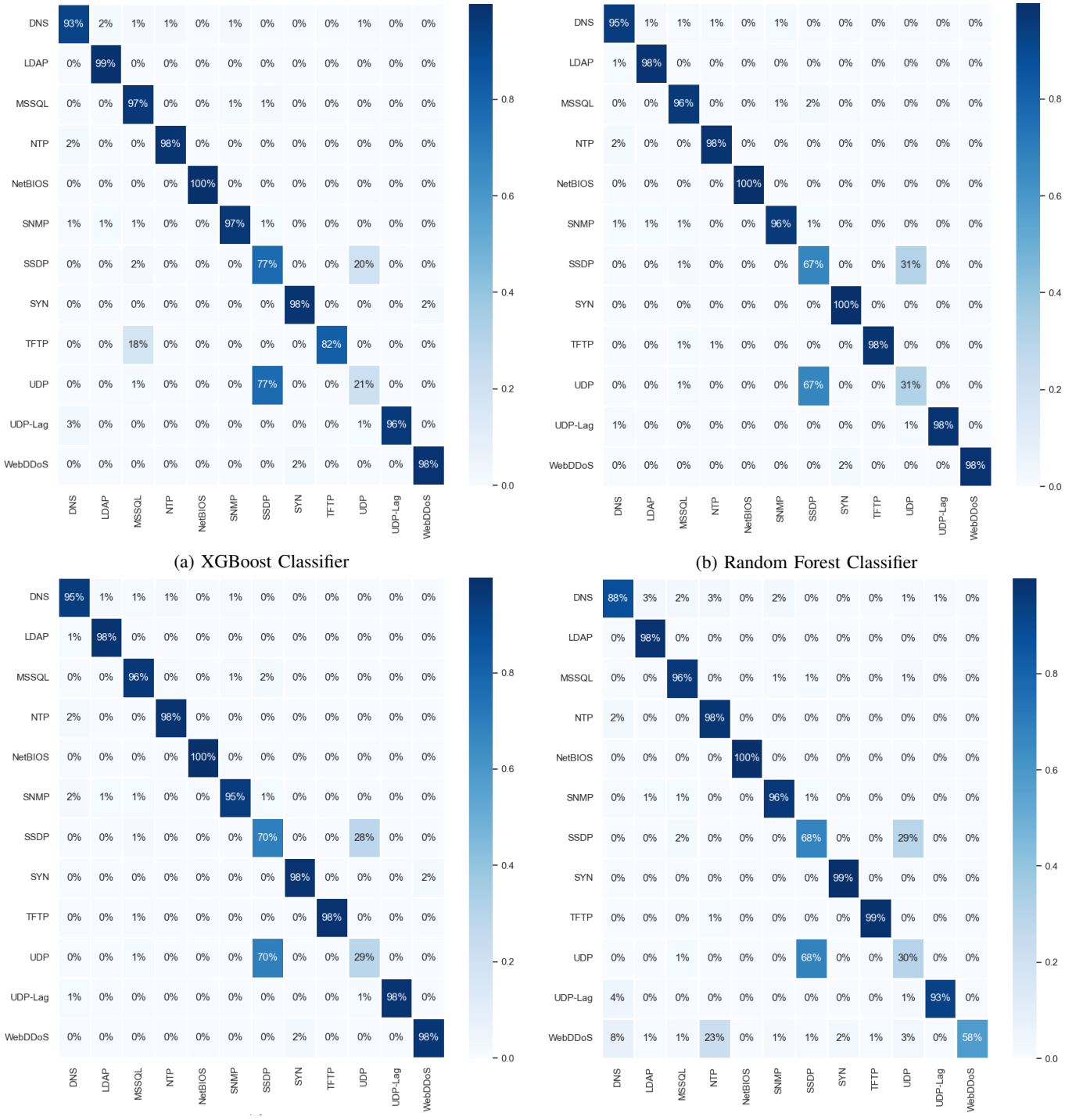


(c) Recall comparison



(d) F1-score comparison

Fig. 2: Classification performance on different sizes of training data



(a) XGBoost Classifier

(b) Random Forest Classifier

(c) Decision Tree Classifier

(d) Gaussian Naive Bayes Classifier

Fig. 3: Confusion matrices over unseen flow data ($\approx$ 42 million of network flow data)

TABLE II: Comparison of classification performances

Paper	#Flow features	#Attack types	Classifier	Accuracy	Precision	Recall	F1-score
[19]	80	12	ID3	-	0.78	0.65	0.69
			RF	-	0.77	0.56	0.62
			Naive Bayes	-	0.41	0.11	0.05
			Logistic regression	-	0.25	0.02	0.04
[32]	24	12	MLP	-	0.8519	0.7651	0.7544
	82	12	MLP	-	0.9116	0.7941	0.7939
[33]	24	3	Naive bayes	-	0.790	0.004	0.008
			SVM	-	0.988	0.459	0.627
			Decision Tree	-	0.997	0.704	0.825
			Logistic regression	-	0.25	0.02	0.04
[34]	78	5	Auto encoder + MLP	0.9834	0.9791	0.9848	0.9818
[18]	25	7	Naive bayes	0.9625392	0.96	0.96	0.96
Proposed	4	10/12	XGBoost	0.90354	0.90390	90354	0.90227
			RF	0.89859	0.89849	0.89859	0.8983
			Decision Tree	0.89689	0.89679	0.89689	0.8964
			KNN 0.89239	0.89239	0.89239	0.8918	

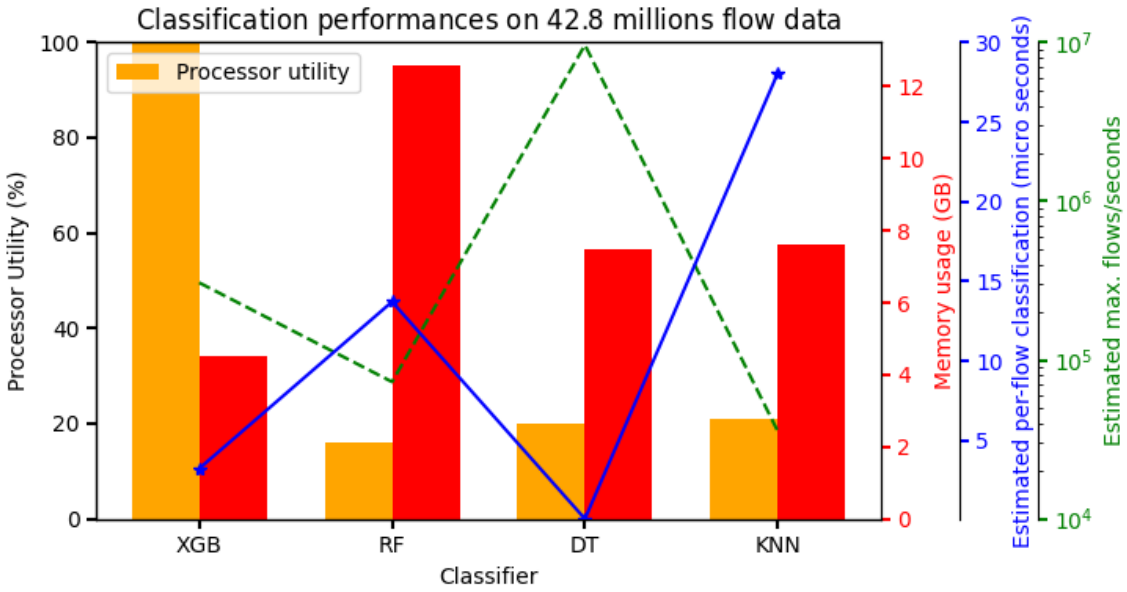


Fig. 4: Estimated classification performance and the resources usage

The result shows that there is a slight improvement over the training result. This phenomenon is due to the duplicated flow data that occurs at different time spans. DDoS attacks are usually launched by compromised hosts or farm servers that run a similar or the same DDoS script. As consequence, the generated flow data might yield similar patterns.

Overall, the performance is consistent with the training dataset, having a disability to discriminate between SSDP and UDP DDoS attacks. The best classification result was obtained by Random forest and Decision tree classifier with accuracy of $\geq 95\%$. The XGBoost and KNN classifiers have worse discrimination performance than the training dataset. The KNN classifier cannot correctly infer web DDoS attacks while the XGBoost cannot infer TFTP attacks.

C. Real-time feasibility

Acquiring quick flow classification is not the only requirement for real-time applicability. The computational resource used to infer is crucial to guarantee the scalability of flow classification during high-load traffic. Figure 4 shows the summary of processor utility, memory usage, and flow classification speed for each selected classifier. The test is run on a laptop machine with an Intel i7 10750H and 32 GB of RAM.

XGBoost has better processor utilization compared to other classifiers implemented in Sklearn library. Consequently, it placed second in the classification speed test. Meanwhile, the KNN classifier has the worst classification speed among the selected classifiers.

The decision tree classifier achieves the fastest classification speed within the sub-microsecond band per-flow data. As a result, for every second, decision trees can infer $\approx 9.6 millions$ flow labeling. The resource consumption for the decision tree classifier is also among the lowest ones, requiring only $\leq 8GB$ of memory.

IV. Conclusion

This study shows that the proposed compact flow features are applicable for real-time DDoS attack classification. First, it

has a classification performance comparable to existing work that uses bloated flow features, reaching $\geq 89\%$ score in accuracy, precision, recall, and F1-score. The trained classifier is also generalizable for unseen flow data without significant degradation. The decision tree classifier shows the best result. It has an estimated classification capacity of ≈ 9.6 million of flow data per second.

REFERENCES

[1] Perez-Diaz, J. A., Valdovinos, I. A., Choo, K. K. R., & Zhu, D. (2020). "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning." IEEE Access, 8, 155859-155872.

[2] Benzaïd, C., Boukhalfa, M., & Taleb, T. (2020, May). "Robust self-protection against application-layer (D) DoS attacks in SDN environment." In 2020 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE.

[3] Chen, S., Shen, C., Yu, D., Wu, Y., & Wu, C. (2021). "Intelligent DDoS Detection in Botnet Combined with Packet-Level Features under SDN." In 2021 International Symposium on Networks, Computers and Communications (ISNCC) (pp. 1-6). IEEE.

[4] Yungaicela-Naula, N. M., Vargas-Rosales, C., Perez-Diaz, J. A., & Carrera, D. F. (2022). "A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning. Journal of Network and Computer Applications," 103444.

[5] Musumeci, F., Ionata, V., Paolucci, F., Cugini, F.,& Tornatore, M. (2020, June). "Machine-learning-assisted DDoS attack detection with P4 language." In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

[6] Friday, K., Kfoury, E., Bou-Harb, E., & Crichigno, J. (2020, June). "Towards a unified in-network DDoS detection and mitigation strategy." In 2020 6th IEEE Conference on Network Softwarization (NetSoft) (pp. 218-226). IEEE.

[7] MAHRACH, S., & HAQIQ, A. (2020). "DDoS flooding attack mitigation in software defined networks." International Journal of Advanced Computer Science and Applications, 11(1).

[8] Simsek, G., Bostan, H., Sarica, A. K., Sarikaya, E., Keles, A., et al. (2019, August). "Dropppp: a P4 approach to mitigating dos attacks in SDN. In International Workshop on Information Security Applications (pp. 55-66)." Springer, Cham.

[9] Ujjan, R. M. A., Pervez, Z., Dahal, K., Khan, W. A., Khattak, A. M., & Hayat, B. (2021). "Entropy based features distribution for anti-ddos model in sdn." Sustainability, 13(3), 1522.

[10] da Silveira Ilha, A., Lapolli, A. C., Marques, J. A., & Gaspary, L. P. (2020). "Euclid: A fully in-network, P4-based approach for real-time DDoS attack detection and mitigation." IEEE Transactions on Network and Service Management, 18(3), 3121-3139.

[11] Ding, D., Savi, M., Pederzolli, F., Campanella, M., & Siracusa, D. (2021). "In-network volumetric DDoS victim identification using programmable commodity switches. IEEE Transactions on Network and Service Management," 18(2), 1191-1202.

[12] Salahuddin, M. A., Bari, M. F., Alameddine, H. A., Pourahmadi, V., & Boutaba, R. (2020, November). "Time-based anomaly detection using autoencoder." In 2020 16th International Conference on Network and Service Management (CNSM) (pp. 1-9). IEEE.

[13] Abreu Maranhão, J. P., Carvalho Lustosa da Costa, J. P., Pignaton de Freitas, E., Javidi, E., & Timóteo de Sousa Júnior, R. (2020). "Error-robust distributed denial of service attack detection based on an average common feature extraction technique." Sensors, 20(20), 5845.

[14] Can, D. C., Le, H. Q., & Ha, Q. T. (2021, April). "Detection of distributed denial of service attacks using automatic feature selection with enhancement for imbalance dataset." In Asian Conference on Intelligent Information and Database Systems (pp. 386-398). Springer, Cham.

[15] Elsayed, M. S., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2020, August). "Ddosnet: A deep-learning model for detecting network attacks." In 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM) (pp. 391-396). IEEE.

[16] Najafimehr, M., Zarifzadeh, S., & Mostafavi, S. (2022). "A hybrid machine learning approach for detecting unprecedented DDoS attacks." The Journal of Supercomputing, 78(6), 8106-8136.

[17] Ortet Lopes, I., Zou, D., Ruambo, F. A., Akbar, S., & Yuan, B. (2021). "Towards effective detection of recent DDoS attacks: A deep learning approach." Security and Communication Networks, 2021.

[18] Gohil, M., & Kumar, S. (2020, December). "Evaluation of classification algorithms for distributed denial of service attack detection." In 2020 IEEE Third International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (pp. 138-141). IEEE.

[19] Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy." In 2019 International Carnahan Conference on Security Technology (ICCST) (pp. 1-8). IEEE.

[20] Carvalho, R. N., Costa, L. R., Bordim, J. L., & Alchieri, E. A. (2021, November). "Detecting DDoS Attacks on SDN Data Plane with Machine Learning." In 2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW) (pp. 138-144). IEEE.

[21] Macías, S. G., Gaspary, L. P., & Botero, J. F. (2020). "ORACLE: Collaboration of Data and Control Planes to Detect DDoS Attacks." arXiv preprint arXiv:2009.10798.

[22] Dimolianis, M., Pavlidis, A., & Maglaris, V. (2020, February). "A multi-feature DDoS detection schema on P4 network hardware." In 2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN) (pp. 1-6). IEEE.

[23] Rebecchi, F., Boite, J., Nardin, P. A., Bouet, M., & Conan, V. (2019). "DDoS protection with stateful software-defined networking". International Journal of Network Management, 29(1), e2042.

[24] Hill, J., Aloserij, M., & Grosso, P. (2018, November). "Tracking network flows with P4." In 2018 IEEE/ACM Innovating the Network for Data-Intensive Science (INDIS) (pp. 23-32). IEEE.

[25] Tavares, K., & Ferreto, T. (2019, May). "DDoS on Sketch: Spoofed DDoS attack defense with programmable data plans using sketches in SDN." In Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (pp. 805-819). SBC.

[26] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017, February). "Characterization of tor traffic using time based features." In ICISSp (pp. 253-262).

[27] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., et al. (2008). "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM computer communication review, 38(2), 69-74.

[28] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., et al. (2014). "P4: Programming protocol-independent packet processors." ACM SIGCOMM Computer Communication Review, 44(3), 87-95.

[29] Kim, C., Sivaraman, A., Katta, N., Bas, A., Dixit, A., & Wobker, L. J. (2015, August). "In-band network telemetry via programmable dataplanes." In ACM SIGCOMM (Vol. 15).

[30] Tan, L., Su, W., Zhang, W., Lv, J., Zhang, Z., et al. (2021). "In-band network telemetry: A survey." Computer Networks, 186, 107763.

[31] DSRGBRIN, "Flow-Condensing". Available at https://github.com/DSRGBRIN/Flow-Condensing (2022/09/08)

[32] Can, D. C., Le, H. Q., & Ha, Q. T. (2021, April). "Detection of distributed denial of service attacks using automatic feature selection with enhancement for imbalance dataset." In Asian Conference on Intelligent Information and Database Systems (pp. 386-398). Springer, Cham.

[33] Vuong, T. H., Thi, C. V. N., & Ha, Q. T. (2021, April). "N-tier machine learning-based architecture for DDoS attack detection." In Asian Conference on Intelligent Information and Database Systems (pp. 375-385). Springer, Cham.

[34] Wei, Y., Jang-Jaccard, J., Sabrina, F., Singh, A., Xu, W., & Camtepe, S. (2021). "Ae-mlp: A hybrid deep learning approach for ddos detection and classification." IEEE Access, 9, 146810-146821.