Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 357–361
DOI: 10.15439/2021F114
ISSN 2300-5963 ACSIS, Vol. 25

Matrix profile for DDoS attacks detection

Faisal Alotaibi
Department of Computer Science
University of Liverpool

Email:Faisal.alotaibi@liverpool.ac.uk

Alexei Lisitsa
Department of Computer Science
University of Liverpool

Email: A.Lisitsa@liverpool.ac.uk

***Abstract*—Several previous studies have focused on Distributed Denial of Service (DDoS) attacks, which are a crucial problem in computer network security. In this paper we explore the applicability of a a time series method known as a matrix profile to the anomaly based DDoS attacks detection. The study thus examined how the matrix profile method performed in diverse situations related to DDoS attacks, as well as identifying those features that are most applicable in various scenarios. Based on reported empirical evaluation the matrix profile method is shown to be efficient against most of the considered types of DDoS attacks.**

I. Introduction

THE Internet has grown at an exponential rate since the 1960s [1], and 3 billion people now surf the Internet every day to access social media, banking, shopping, and other everyday services [1]. However, the Internet is not a safe zone, and privacy and information security are major causes for concern. Any system connected to the Internet is subject to security threats from hackers, viruses, or sniffers [2]. The most common approach to degrading the availability of a targeted service on the Internet is a Distributed Denial of Service (DDoS) attack. DDoS attacks can range from the misuse of application-level vulnerabilities to high-volume flooding on a network [3] , and they are undoubtedly one of the leading causes of concern for many companies, organisations, and institutions [1]. A DDoS attack may thus refer to any malicious coordinated attack against any form of online services, whether these are commercial websites, bank websites, or government websites. A DDoS attack is usually performed by a massive number of bots over a specified period, either flooding a network with high volumes of irrelevant data to create excess traffic or attacking a vulnerable application to render it useless [4]. Although it is often easy to probe service availability and decongest the network, the most significant challenge in assessing such attacks lies in differentiating between legitimate congestion and attacker-initiated congestion, however, as these may manifest in similar ways [5].

There are many types of DDoS attacks, though these can be summarised as follows:

- Value Based Attacks. Such attacks include i) User Datagram Protocol (UDP) floods, ii) Ping Floods, and iii) Spoofed-packet floods.
- Protocol Based Attacks. Such attacks include i) SYN Floods, ii) fragmented packet attacks, iii) “Ping of Death”, and iv) Smurf DDoS.
- Application Layer Attacks. Such attacks include i) low-and-slow attacks, ii) GET/POST floods, iii) attacks that target Apache, iv) attacks that target Windows, and vi) OpenBSD vulnerabilities.

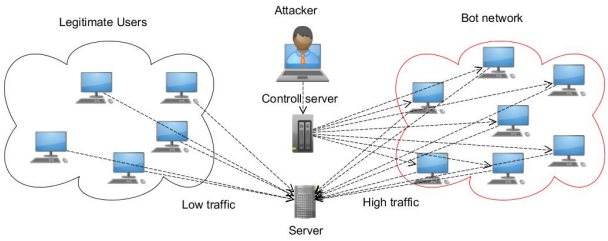

Fig. 1. DDoS attacks

As Cisco reports,The DDoS is predicted to become more harmful in the future and the world needs to develop appropriate solutions for the many scenarios that could arise. Over the next few years, all forms of DDoS attacks are likely to become more common, with predictions suggesting that the total number of DDoS attacks will double from the 7.9 million seen in 2018 to over 15 million by 2023.

Research on DDoS attacks detection and mitigation has proposed many efficient solutions [6]–[8]. Still due to unprecedented scale of the threat, a need for new highly scalable and precise solutions remains high. In this paper we present our initial study on the applicability of very powerful and promising approach in time series data mining, *matrix profile* [9], method for the detection of DDoS attacks. The paper is organized as follows. In the next section we present the basics of the matrix profile (MP) method, anomaly based detection using MP, dataset used in the experiments and data pre-processing. In the following section we discuss the details of the implementation. Section IV presents the results and discussion. Section V concludes the paper.

II. Matrix profile

The Matrix profile is a method, including a data structure and very efficient algorithms for computing *all-pairs-similarity-search* (or similarity join) for time series subsequences [10]. Since its invention, matrix profile has been shown to be a powerful method for solving various tasks in

time series data mining including motif discovery, classification and anomaly detection among others [9], [11], [12]. The idea of matrix profile is very natural. For a time series $T = t_0, \dots t_n$ and a positive integer m denoted by $T_{i,m}$ a subsequence $t_i, \dots, i_{i+m-1}$ of T. The matrix profile of T includes the following data: 1) distance profile which is a vector of distances between all pairs of subsequences $T_{i,m}$ and $T_{j,m}$ of T; 2) profile index which for every i stores j such that $T_{j,m}$ is the closest to $T_{i,m}$ among all m-subsequences ("a distance to the nearest neighbour"). While any concept of distance/metrics can be used in matrix profile, the standard euclidean distance between z-normalized values is a common choice [9], [11], [12]. The advantages of the matrix profile method include its support for very efficient and highly parallelizable algorithms for similarity join, the fact that it is domain agnostic, that fact that it offers precise solutions and requires only a single parameter (but can be expanded to multi-dimensional cases as well). Yet another crucial feature for many applications of matrix profile is that it supports *incremental* algorithms, so it can be applied for online processing.

Algorithm 1 Matrix profile

procedure MATRIX PROFILE(T, m)
  $n \leftarrow$ length of (T), $l \leftarrow$ *n-m+1*
  $\mu, \sigma \leftarrow$ *ComputeMeanStd(T,m)*
  $QT \leftarrow SlidingDotProduct(T[1:m], T)$
  $QT_{first} \leftarrow QT$
  $D \leftarrow CalculateDistanceProfile(QT, \mu, \sigma)$
  $P \leftarrow D$, $I \leftarrow ones$
  for $i = 2$ **to** l **do**
    for $j = l$ **downto** 2 **do**
      $QT[j] \leftarrow QT[j-1] - T[j-1] \cdot T[i-1] + T[j+m-l] \cdot T[j+m-l]$
    end for
    $QT[1] \leftarrow QT_{frist}[i]$
    $D \leftarrow CalculateDistanceProfile(QT,\ \mu,\ \sigma,\ i)$
    $P,\ I \leftarrow ElementWiseMin(P,\ I,\ D,\ i)$
  end for
  Return P, I
end procedure

Time series discords, that is subsequences with the large (maximal) distances to their nearest neighbours have already been proposed as novelty/anomaly detectors [9] and they can be easily identified using matrix profile data. Indeed, one has just to check the values of $\mid p(i) - i \mid$, where $p(i)$ is a profile index value for i. Thus, if we consider the metric used in matrix profile as a *similarity measure*, an *anomalous subsequence* is the one for which the most similar subsequence is found far away. Such an approach for anomaly detection has been considered already in [13], [14] for the medical domain.

A. Anomaly based detection with MP

In this study we investigate the applicability of matrix profile method in computer networks security domain, in particular, for anomaly based intrusion detection.

The outline of the proposed generic scheme for such a detection is as follows. We fix a time window W, subsequences length M and threshold value T.

- The traffic data is converted into time series;
- Matrix Profile method is applied to the subsequences of length M of the time series;
- If MP value for a given time window is greater than T then an anomaly is detected, if it is lower than T, no anomaly is detected and the traffic is considered as normal.

The implementation of such a scheme requires some choices to be made. The conversion of traffic data into time series can be done in various ways using different features of the data. Depending on the format of the data further processing might be needed as well. Finally the choices of the time window and threshold value have to be made.

We have focused on detection of DDoS attacks and have used for experiments a DDoS Evaluation Dataset (CIC-DDOS2019) obtained from the Canadian Institute for Cybersecurity[1] . This dataset is fully labelled, which helps in terms of measuring the performance matrix profile based on making comparisons with the times when attacks take place. We have conducted the experiments in offline scenario, while the case of online processing is a topic of our ongoing research.

B. Dataset CIC-DDoS2019

This dataset can be publicly accessed and includes two data formats: the first is pcap, while the second is CSV. While the pcap files include raw data recorded over two days, the CSV files include the information on network flows extracted using CICFlowMeter-V3. There are 80 variable features available. The features used in this work included Total Fwd Packets, Total Bwd Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length, Max, Fwd Packet Length, Min, 'Subflow Fwd Packets', Fwd Packet Length Mean, Fwd Packet Length Std, 'Bwd Avg Bulk Rate, and Bwd Packet Length Max.

Each CSV file contains a label for the flow that describe the flow type (normal or named for the nature of the attack); thus, for each type of attack, there is a separate CSV file.

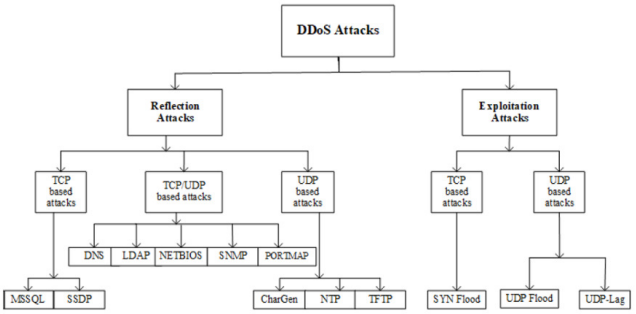

Fig. 2. Dataset for DDoS (https://www.unb.ca/cic/datasets/ddos-2019.html)

C. Data pre-processing

Fig. 3 shows the data sample used in this implementation. The data set file includes a time stamp indicating the time of

[1]https://www.unb.ca/cic/datasets/ddos-2019.html

[19]:

timestamp	Unnamed: 0	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd Packets	...	Active Std	Active Max	Ac
2018-12-01 10:51:39.813448	425	172.16.0.5-192.168.50.1-634-60495-17	172.16.0.5	634	192.168.50.1	60495	17	2018-12-01 10:51:39.813448	28415	97	...	0.0	0.0	
2018-12-01 10:51:39.820842	430	172.16.0.5-192.168.50.1-60495-634-17	192.168.50.1	634	172.16.0.5	60495	17	2018-12-01 10:51:39.820842	2	2	...	0.0	0.0	
2018-12-01 10:51:39.852499	1654	172.16.0.5-192.168.50.1-634-46391-17	172.16.0.5	634	192.168.50.1	46391	17	2018-12-01 10:51:39.852499	48549	200	...	0.0	0.0	
2018-12-01 10:51:39.890213	2927	172.16.0.5-192.168.50.1-634-11894-17	172.16.0.5	634	192.168.50.1	11894	17	2018-12-01 10:51:39.890213	48337	200	...	0.0	0.0	
2018-12-01 10:51:39.941151	694	172.16.0.5-192.168.50.1-634-27878-17	172.16.0.5	634	192.168.50.1	27878	17	2018-12-01 10:51:39.941151	32026	200	...	0.0	0.0	

Fig. 3. Dataset Before conversion

Out[25]:

timestamp	Total Length of Bwd Packets
2018-12-01 12:23:13	29918.0
2018-12-01 12:23:14	17832.0
2018-12-01 12:23:15	0.0
2018-12-01 12:23:16	0.0
2018-12-01 12:23:17	0.0

In [ ]:

Fig. 4. Dataset after conversion

the start of the flow. In order to use this data set in a matrix profile it has to be converted. The first step is to aggregate the data set for the traffic based on the time window: different flows have different times, and it is important to sum or group all the flows for each feature.

In this work, we have applied the following form of aggregation. For a chosen feature and for each time window we consider *all flows which start in that window* and aggregate the feature values across all these flows. For numerical features the summation is used as an aggregation operation.

As an example, Figure 4 shows the result of converting the data where `Total Length of Bwd Packets` is used as a feature. As most of the data in the data set were attacks, an additional normal traffic was also added to the data set; this was added thirty minutes before the attack start and after the attack end.

III. Implementation

Our implementation has proceeded by following steps.

1) Reading the dataset
2) Increasing normal traffic
3) Feature selection
4) Resampling traffic (time window aggregation)
5) Running data in matrix profile mode
6) Processing the output for each threshold
7) Measuring performance .
8) Repeating the experiment with different features and different attacks

This section offers details for each step in the implementation process. In this experiment, several attacks were assessed on both day 1 and day 2, as shown in table 2. The CSV file consists of different traffic flows including labels, with two types of labels (normal or attack). Most of the flows in the csv file as downloaded were attacks. Normal traffic in this dataset is labelled (BEIGN), while attacks are each named after the specific type of attack. As attacks dominated the traffic in the dataset, an increase in normal traffic was required before beginning the experiment for the following reasons:

1) The matrix profile works to identify anomalies, which must thus anomaly be unusual events; a dataset where attacks seem to be the norm is thus inappropriate for attack detection.
2) In real network scenarios, the normal traffic should dominate the anomalous traffic rather than the other way around.

TABLE I
RESULTS FOR EXPERIMENT THRESHOLD 0.5 FOR DAY 1

Distributed denial of service attacks				
Day	Attack	Threshold	Features	Accuracy
1	DrDNS	0.5	All	66%
1	LDAP	0.5	Fwd Packet Length Std	86%
1	MSSQL	0.5	Fwd Packet Length Std	80%
1	NETBIOS	0.5	Fwd Packet Length Std	82%
1	NTP	0.5	All	38%
1	SNMP	0.5	All	72%
1	SSDP	0.5	Total Length of Bwd Packets	82%
1	UDP	0.5	Fwd IAT MEAN	69%
1	SYN	0.5	Fwd packets Length Std	93%
1	TFTP	0.5	All	93%
1	UDPLag	0.5	All	69%

TABLE II
RESULTS FOR EXPERIMENT THRESHOLD 1 FOR DAY 1

Distributed Denial of service attacks				
Day	Attack name	threshold value	features	Accuracy
1	DrDNS	1	All features	66%
1	LDAP	1	Fwd IAT min	88%
1	MSSQL	1	Syn flag count	86%
1	NETBIOS	1	Bwd IAT Std	82%
1	NTP	1	All features	38%
1	SNMP	1	All features	72%
1	SSDP	1	Bwd IAT Max	81%
1	UDP	1	Fwd IAT Std	68%
1	SYN	1	All features	93%
1	TFTP	1	All features	82%
1	UDPLag	1	All features	69%

TABLE III
RESULTS FOR EXPERIMENT THRESHOLD 2 FOR DAY 1

Distributed Denial of service attacks				
Day	Attack name	threshold value	features	Accuracy
1	DNS	2	All features	66%
1	LDAP	2	All features	86%
1	MSSQL	2	All features	80%
1	NETBIOS	2	All features	82%
1	NTP	2	All features	38%
1	SNMP	2	All features	72%
1	SSDP	2	All features	81%
1	UDP	2	All features	68%
1	SYN	2	All features	93%
1	TFTP	2	All features	82%
1	UDPLag	2	All features	69%

Normal traffic was thus increased to make it the most common. As the attack duration in each case was around 10 to 15 minutes, similar steps were followed in each case: to increase the normal traffic, all the normal traffic available in a given dataset was duplicated multiple times; the resulting new normal traffic block was then inserted 30 minutes before the attack began, with a random time function to change the distribution within that 30 minutes. This was repeated for the 30 minutes after the attack ended in each case. This created datasets dominated by normal traffic.

The next step was to select features one by one, as the matrix profile accepts only one dimension. It was thus necessary to run the experiment for each feature separately. Re-sampling of the traffic to deliver time window aggregation was required after feature selection, based on the time window required.

The time window used in the experiments was one second. Further choices were 1) the length of the subsequences used in Matrix Profile set to M=10; 2) threshold MP values tested were 0.5, 1, 2.

The performance of the detection procedure was measured in terms of detection precision using labelled data in the dataset as the source of ground truth. The detection event is considered as *true positive* if the anomaly in a time window was detected and there was at least one flow starting in that window which is labelled as an attack.

In this study, a Python 3 library from the Matrix Profile Foundation was used to perform Matrxi Profile computations.[2]

IV. RESULTS AND DISCUSSION

We conducted the experiments and created confusion matrices for each combination of an attack, chosen features and chosen threshold values. The results were then assessed against the following criteria.

1) Success: Where the confusion matrix accuracy for each threshold value in each feature is over 70%, it is considered to represent a successful detection. This occurred for LDAP, MSSQL, NETBIOS, SSDP, SYN and TFTP in the day one results.
2) Struggling: Where the confusion matrix accuracy for each threshold value in each feature is 50% to 70% inclusive, the detection cannot be considered good. This occurred for the portmap attack, where detection showed 57% accuracy.
3) Failure: When the confusion matrix accuracy for each threshold value in each feature is 50% or lower, this is considered as a failure of detection, as seen in the day one attack NTP, which had an accuracy of only 38%.

After all the experiments were completed, the best result for each attack at each threshold value was recorded. As seen in the tables:

- The average of all accuracy result in threshold value 0.5 is 64.68%
- The average of all accuracy result in threshold value 1.0 is 67.84%
- The average of all accuracy result in threshold value 2.0 is 74.53%

Different threshold values produce different accuracy results. Consequently, based on our experiment we suggest that the optimal threshold value to be 2.0.

Finally, we notice, based on the literature review that thus far no study have used any unsupervised learning method with this dataset. However, there have been some works that used supervised learning exemplified in [15]. Their method successfully achieved an accuracy of 99%. While this result is typical in supervised learning, our work is different. First of all, we use unsupervised processing/learning based on matrix profile. Second, Matrix profile only accepts one-dimensional data . Further to that our pre-processing of the data is done to increase normal traffic in the dataset which simulates how

[2]https://pypi.org/project/matrixprofile/

TABLE VI
RESULTS FOR EXPERIMENT THRESHOLD 2 DAY 2

Distributed Denial of service attacks				
attack day	Attack name	threshold value	features	Accuracy
2	LDAP	0.5	All features	74%
2	MSSQL	0.5	All features	82%
2	NETBIOS	0.5	All features	89%
2	portmap	0.5	All features	57%

TABLE IV
RESULTS FOR EXPERIMENT THRESHOLD 0.5 FOR DAY 2

Distributed Denial of service attacks				
attack day	Attack name	threshold value	features	Accuracy
2	LDAP	0.5	All features	74%
2	MSSQL	0.5	All features	82%
2	NETBIOS	0.5	All features	89%
2	portmap	0.5	All features	57%

TABLE V
RESULTS FOR EXPERIMENT THRESHOLD 1 DAY 2

Distributed Denial of service attacks				
attack day	Attack name	threshold value	features	Accuracy
2	LDAP	0.5	All features	74%
2	MSSQL	0.5	All features	82%
2	NETBIOS	0.5	All features	89%
2	portmap	0.5	All features	57%

DDoS attacks normally happen in the network. Also we give more details of the detection of different types of DDoS attack while other studies treated all DDoS attack as one group.

V. CONCLUSION

This study aimed to examine the advantages of using a Matrix Profile algorithm to address network security problems with DDoS attacks. The results of this initial study showed that this method is effective against multiple specific types of DDoS attacks. The next step will be to develop a module to allow the Matrix Profile method to be used online and the resulting performance to be assessed. Broader classes of settings should be explored and the detection of wider classes of attacks should be considered.

REFERENCES

[1] Dhruba Kumar Bhattacharyya and Jugal Kumar Kalita. Ddos attacks evolution, detection, prevention, reaction, and tolerance. 2016.

[2] Zhuo Lin. Internet security and firewall [j]. *Journal of Changsha University*, 15:32–35, 2001.

[3] Susan J Harrington. Why people copy software and create computer viruses. *Information Resources Management Journal (IRMJ)*, 2(3):28–38, 1989.

[4] Neelam Dayal and Shashank Srivastava. Analyzing behavior of DDoS attacks to identify DDoS detection features in SDN. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 274–281. IEEE, jan 2017.

[5] Shibo Luo, Jun Wu, Jianhua Li, and Bei Pei. A Defense Mechanism for Distributed Denial of Service Attack in Software-Defined Networks. *9th International Conference on Frontier of Computer Science and Technology (FCST 2015)*, pages 325–329, 2015.

[6] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, and Muttukrishnan Rajarajan. A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, 36(1):42–57, 2013.

[7] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1–31, 2009.

[8] Dhruba Kumar Bhattacharyya. *DDoS attacks: evolution, detection, prevention, reaction, and tolerance*. Chapman and Hall/CRC, 2019.

[9] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. Ieee, 2016.

[10] Shaghayegh Gharghabi, Shima Imani, Anthony Bagnall, Amirali Darvishzadeh, and Eamonn Keogh. Matrix profile xii: Mpdist: a novel time series distance measure to allow data mining in more challenging scenarios. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 965–970. IEEE, 2018.

[11] Dieter De Paepe, Sander Vanden Hautte, Bram Steenwinckel, Filip De Turck, Femke Ongenae, Olivier Janssens, and Sofie Van Hoecke. A generalized matrix profile framework with support for contextual series analysis. *Eng. Appl. Artif. Intell.*, 90(C), April 2020.

[12] Frank Madrid, Shima Imani, Ryan Mercer, Zachary Zimmerman, Nader Shakibay, and Eamonn Keogh. Matrix profile xx: Finding and visualizing time series motifs of all lengths using the matrix profile. In *2019 IEEE International Conference on Big Knowledge (ICBK)*, pages 175–182. IEEE, 2019.

[13] Haemwaan Sivaraks and Chotirat Ratanamahatana. Robust and accurate anomaly detection in ecg artifacts using time series motif discovery. *Computational and mathematical methods in medicine*, 2015:453214, 01 2015.

[14] Rutuja Wankhedkar and Sanjay Kumar Jain. Motif discovery and anomaly detection in an ecg using matrix profile. In *Progress in Advanced Computing and Intelligent Engineering*, pages 88–95. Springer, 2021.

[15] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. 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)*, pages 391–396. IEEE, 2020.