International Journal of Engineering & Technology, 7 (2.7) (2018) 203-208

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper



Traffic flow features as metrics (TFFM): detection of application layer level DDOS attack scope of IOT traffic flows

Kalathiripi Rambabu[1*], N. Venkatram[2]

[1]Research scholar, Department of ECE, KL University, Guntur District- 522 502, Andrapradesh, India
[2] Professor, Department of ECSE, KL University, Guntur District- 522 502, Andrapradesh, India
**Email: rambabukala@gmail.com*

Abstract

The phenomenal and continuous growth of diversified IOT (Internet of Things) dependent networks has open for security and connectivity challenges. This is due to the nature of IOT devices, loosely coupled behavior of internetworking, and heterogenic structure of the networks. These factors are highly vulnerable to traffic flow based DDOS (distributed-denial of services) attacks. The botnets such as "mirae" noticed in recent past exploits the IoT devises and tune them to flood the traffic flow such that the target network exhaust to response to benevolent requests. Hence the contribution of this manuscript proposed a novel learning-based model that learns from the traffic flow features defined to distinguish the DDOS attack prone traffic flows and benevolent traffic flows. The performance analysis was done empirically by using the synthesized traffic flows that are high in volume and source of attacks. The values obtained for statistical metrics are evincing the significance and robustness of the proposed model

Keywords: *IoT, DDOS, Botnet, DDoSFlowgen, MEC Shield, Preventive Measures, Traffic Flow, Mirae.*

1 Introduction

IoT equipment's usage is continuous and proliferating worldwide due to remote accessibility, and low costs. Moreover, these devices are incorporated with internet access, enabling users to operate them from remote places. A few areas incorporating IoT devices include thermostats, internet televisions, closed-circuit monitors, vehicle systems (GPS), smart devices, etc. Due to the wide use of IoT devices, the shift from conventional computers involves serious intrusion scope challenges. For example, DDoS intrusion observed in 2016 has been a serious intrusion, affecting the Dyn server. This resulted in shut down of most applications including Twitter, Spotify, Netflix, NYT [1]. Some of the factors contributing to the launch of this attack include- Rapid increase in the use of IoT equipment worldwide not backed up with an adequate network infrastructure such as unavailability of strict safety norms. Gartner estimates the IoT devices employed worldwide will exceed 26 billion with other research organizations evaluating the number above 100 billion [2]. In addition, sudden growth in devices and traffic quantum results in storage space limitations and network equipment. This leads to overload on in- line ADSs, which fail to check each data message in detail [3]. Accordingly, packet-sampling and flow-sampling techniques are engaged to lower the amount of information to be processed by the ADSs, which can bypass malware packets from the compromised nodes to the destination/ source nodes. Further, the malicious 'Mirae' that constantly monitors the internet for IoT equipment guarded through manufacturer's default login credentials. For instance, the entire volume of 68 usernames and passwords can access through the 'Mirae' program code [4].

1.1 Problem formulation

Data transmission over the conventional network systems largely depends on the routing-table lookup routines to forward messages from source nodes to corresponding end nodes. As mentioned earlier, strict storage space and computational challenges, in-line ADSs are unable to check each message in depth. Accordingly, the end network's firewall mechanism or ADS should check for the incoming messages in depth and ignore any malware messages or flows. Any DDoS intrusion requires the considerable volume of bandwidth at the compromised IoT equipment that attempts to induce large quantum of malware messages into the transmission system. Further, upon successful arrival of these malware messages or flows at the end network, they crowd the firewall mechanism, rendering it inaccessible to genuine messages. Accordingly, the network framework must be enabled to check traffic flows on the route and deny services to malware message prior to being crowded at the destination firewalls. This ensures two primary advantages including reduction of network crowding and ensuring that the compromised IoT equipment do not amplify the effect on the victim node.

2 Related Research

Several research works are put forward for defending against DDoS attacks over the internet. A few of the researchers also emphasized on categorizing DDoS intrusions and defending strategies into diverse groups [5], [6], [7]. But very few works have been put forward for handling DDoS attack issue in the IoT environment despite the fact that DDoS can pose serious challenges in the IoT context.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intrusion detection techniques are engaged to track the inflowing traffic with three metrics primarily being assessed for tracking. The parameters are IP-address, traffic growth rate and similarity of the traffic. The traffic growth degree metrics are observed to be efficient in detecting congestion but can also result in high FPRs. For instance, a flash-crowd can be a result of genuine user requirements but is believed to be malware issue [8]. The remaining two metrics ensure that the system precisely differentiates malware traffic instances from the flash-crowd instances. The similarity between different messages in the DDoS intrusion is typically much more significant than the flash-crowd instances. This is because the intrusion traffic often originates from a single botnet depicting similarity regarding source IP-address. Further, even if the intruding source-IP is spread globally, the bots put forward similar source-code information and accordingly, checking for similarity among message content will be much more than that of flash-crowd instance [9].

On similar lines, for DDoS intrusion detection, a few authors have presented mining historical intrusion attributes for message sampling. In addition, constant monitoring of the intruding root can enable the server to block the intrusion traffic. This defense mechanism has been observed to pose superior performance against DDoS attacks [7].

LA- model has been put forward as an effective tool against DDoS intrusion. The model functions by establishing the data message sampling degree from the system [10]. Adaptive learning element has been suggested in their earlier paper [7] as an effective defensive strategy in WMN environment.

The study in [11] put forward a different technique, which backs the sink-node through a novel sensor-node. The recently include node is termed as redundant-path and stores a part of the sink-node functions. Accordingly, the possibility of sink-node being down is decreased. Further, the entire subnetwork life is extended. The model is also a cost-effective due to the incorporation of the single node for each subnetwork.

The authors in [12] presented defensive mechanism specific to IoT destination network, addressing the transmission security issues. A prevention strategy against DDoS intrusion has been presented through pre-monitoring and restraining from the intrusion. The defense strategy is supported by various motivations. This mechanism efficiently assists the operating nodes within the IoT transmission system to differentiate malware requests from genuine requests and accordingly separate processing is engaged.

The study in [13] presented a novel research- MECshield, intended to safeguard heterogeneous IoT through smart DDoS- Intrusion-Behaviour Learning. The model uses MEC model to establish localized smart-filters at the boundaries of related source-network or destination-network. A centralized controlling mechanism is incorporated to coordinate among different filters via policy deliveries. These policies are created by intrusion-behaviour insights observed from the analyzer within the controller. These are created prior to being delivered to the related filers present at the intrusion source location. Every filter is equipped with a SOM element [14] and is trained in parallel through local traffic patterns within the policy guidelines. The trained self-organizing maps efficiently identify malware attacks by mapping the attributes of traffic into their map and assess if the traffic is characterized as DDoS.

The concepts of all these studies account for the portion of framing specific rules for the development of an efficient defense strategy including appending resources, IP-address base filtering, preventive blocking and peer-layering transmission. Despite these efforts, robust, effective, and highly structured mechanisms must be developed to operate in the IoT context.

The ideas behind these research works cover part of designing rules of thumb for a defending system in IoT network including adding resources, filtering by ID address, proactive block, and peer layering communication. However, more structured, efficient, and acute approach needs to add into this scope. In addition, the parameters considered to diversify the DDOS attack prone traffic from benevolent traffic flow such as source, repetitive requests, an unrelated sequence of requests are ease to counter by botnets like Mirai [15]. Hence, the contributions of this manuscript endeavor to define novel traffic flow features to train the DDOS attack to detect and defence model that proposed for IOT networks.

3 Model Definition: Drift Detection in Traffic Flow Streams for DDoS Attack Defence

The proposed model is performing the DDOS attack detection and defence for IOT in two phases, and they are detection by training from past traffic, and defence by classifying the present traffic. The training phase notifies the transactions carried in target network as web sessions, and each of these web session is a record labeled as attack prone (positive) or benevolent (negative). Further these records bipartite into two sets denoted as S_{+ve}, S_{-ve}, which is based on their label, such that all positive records (attack prone) records are one set S_{+ve}, and the records labeled as negative (benevolent) as other set S_{-ve}. Further, the depicted model assesses the values related to proposed flow features projected as training metrics for both positive and negative record sets. The following section explores the proposed flow features as training metrics and their significance

3.1 Traffic Flow Features

The buffered traffic flows of the request streams in IOT network can detect as DDOS attack prone (positive) or benevolent (negative) through the similarity observed between the values obtained for the proposed traffic flow features from training corpus and buffered traffic flow of the live IOT network request stream. The traffic flow features proposed as DDOS detection metrics are, (i) "Session Instigation Time Variance" that indicates the difference between the instigation times of sessions, which initiated in sequence. (ii) "Session Termination Time Variance" that indicates the difference between termination times of sessions, which initiated in sequence. (iii) "Request Instigation Time Variance" indicates the gap between the instigation times of the two requests those initiated in sequence. (iv) "Request Termination Time Variance" indicates the gap between the termination times of the two requests in sequence. (v) "Request Sequence" indicates the logical order of the requests initiated in sequence. (vi) "Session Source Sequence" indicates the order of sources those initiated the sessions depicted in sequence. (vii) "Bandwidth per Session" indicates the aggregate bandwidth required to complete the requests depicted in target session.

3.2 Feature Extraction

The vectors $stiv_+, stiv_-$ denote the Session Instigation Time Variances observed for the attack-prone sessions S_{+ve} and benevolent sessions S_{-ve} respectively. Similarly, the vectors $sttv_+, sttv_-$ denotes the Session Termination time variances observed for the sessions listed as attack prone S_{+ve} and benevolent S_{-ve} in respective order The session instigation time variances, and session termination time variances of respective sessions of positive and negative sets S_{+ve}, S_{-ve} are depicted as follows

For each session $\{s_i \exists s_i \in S_{+ve} \wedge i = 1,2,...,|S_{+ve}|-1\}$ Begin

$sitv_+ \leftarrow sit(s_{i+1}) - sit(s_i)$ // moves the variance between session instigation times $sit(s_{i+1}), sit(s_i)$ observed for the sessions s_{i+1}, s_i

$sttv_+ \leftarrow stt(s_{i+1}) - stt(s_i)$ //moves the absolute variance between session termination times $stt(s_{i+1}), stt(s_i)$ observed for the sessions s_{i+1}, s_i

End

Similarly, the session instigation time variances and session termination time variances are also depicting for the benevolent sessions S_{-ve} , which is as follows:

For each session $\{s_i \exists s_i \in S_{-ve} \wedge i = 1,2,...,|S_{-ve}|-1\}$ Begin

$sitv_- \leftarrow sit(s_{i+1}) - sit(s_i)$ (Eq 1)

// (Eq 1) moves the difference between session instigation times $sit(s_{i+1}), sit(s_i)$ observed for the sessions s_{i+1}, s_i

$sttv_- \leftarrow stt(s_{i+1}) - stt(s_i)$ (Eq 2)

// (Eq 2) moves the absolute difference between session termination times $stt(s_{i+1}), stt(s_i)$ observed for the sessions s_{i+1}, s_i

End

Further, the Request Instigation Time variances are recorded as vectors $ritv_+, ritv_-$ for corresponding attack prone sessions S_{+ve} , and benevolent sessions S_{-ve} . Similarly, the Request Termination Variances of attack prone sessions S_{+ve} and benevolent sessions S_{-ve} are also recorded as a respective order of vectors $rttv_+, rttv_-$, which is as follows:

Let the R_{+ve} be the set contains all the requests related to the sessions S_{+ve} in ascending order of their instigation time. Similarly, Let the R_{-ve} be the set contains all the requests related to the benevolent sessions S_{-ve} in ascending order of their instigation time.

For each request $\{r_i \exists r_i \in R_{+ve} \wedge i = 1,2,...,|R_{+ve}|-1\}$ Begin

$ritv_+ \leftarrow rit(r_{i+1}) - rit(r_i)$ (Eq 3)

// (Eq 3) moves the variance between request instigation times $rit(r_{i+1}), rit(r_i)$ observed for the requests r_{i+1}, r_i

$rttv_+ \leftarrow rtt(r_{i+1}) - rtt(r_i)$ (Eq 4)

// (Eq 4) moves the absolute variance between request termination times $rtt(r_{i+1}), rtt(r_i)$ observed for the requests r_{i+1}, r_i

End

Similarly, the request instigation time variances and request termination time variances are also depicting for the benevolent requests R_{-ve} , which is as follows:

For each request $\{r_i \exists r_i \in R_{-ve} \wedge i = 1,2,...,|R_{-ve}|-1\}$ Begin

$ritv_- \leftarrow rit(s_{i+1}) - rit(s_i)$ (Eq 5)

// (Eq 5) moves the variance between request instigation times $rit(r_{i+1}), rit(r_i)$ observed for the requests r_{i+1}, r_i

$rttv_- \leftarrow rtt(r_{i+1}) - rtt(r_i)$ (Eq 6)

// (Eq 6) moves the absolute difference between request termination times $rtt(r_{i+1}), rtt(r_i)$ observed for the requests r_{i+1}, r_i

End

A vector bps_+ denotes the bandwidth consumed by each attack prone session that exists in S_{+ve} , and the other vector bps_- denotes the bandwidth consumed by each benevolent session that exists in S_{-ve} . The metric "bandwidth per session" is assessed for all of the sessions exists in S_{+ve}, S_{-ve} as follows:

For each session $\{s_i \exists s_i \in S_{+ve} \wedge i = 1,2,...,|S_{+ve}|\}$ Begin

$bps_+ \leftarrow \sum_{r \in R_{+ve}} \{bw(r) \exists r \in S_i\}$ // aggregating the bandwidth consumed by the requests those exists in session s_i , and moving it to the vector bps_+

End

Similarly, the bandwidth per session will assess for each of the benevolent session exists in S_{-ve}

For each session $\{s_i \exists s_i \in S_{-ve} \wedge i = 1,2,...,|S_{-ve}|\}$ Begin

$bps_- \leftarrow \sum_{r \in R_{-ve}} \{bw(r) \exists r \in S_i\}$ // aggregating the bandwidth consumed by the requests those exists in session s_i , and further moving it to the vector bps_+

End

Along these metric values, the proposed model also taken the request sequence, and session source sequence into consideration to training the target classifier. The sequence of requests exists in the set R_{+ve} has sorted in ascending order of their instigation time and considered to train the target classifier about attack prone traffic flows. Similarly, the requests exist in set R_{-ve} also being considered in ascending order of their instigation time to train the target classifier about benevolent request flow.

Further, a vector src_+ is formed, which contains the sequence of sources those initiated attack prone sessions. The sources are sequenced in ascending order of their session instigation time. Similarly, the vector src_- contained the sources of the benevolent sessions and sorted further in ascending order of the corresponding session instigation times.

All of these metrics Session instigation time variances, session termination time variances, Request instigation time variances, Request termination time variances, Bandwidth per Session, Requests Sequence, and Session Source Sequence, which depicted for attack prone and benevolent sessions are used further to train the target classifier.

In regard to the context of the proposal, the classifier Adaboost [16] has considered that trained by the proposed metrics and used to detect and defense the DDOS attack prone traffic flows.

4 Experimental Study

This section explores the strategic approach of dataset generation, experimental setup, and the assessment of the performance by comparing the results obtained from the experiments.

4.1 The Data

DDOSFlowgen [17] is an open source model allows synthesizing the flow based DDOS attack under multiple views. This model enables to generate the attack flows from synthesized different attacking networks with adjustable attack vectors, amplification factors, and the attack sources count per network. In addition, the model allows generating normal traffic. The functional model is visualized in Figure 1. The different threats such as flooding scenario, IOT device scanning probe by Mirai [18] are possible to synthesize through DDoSFlowzen.

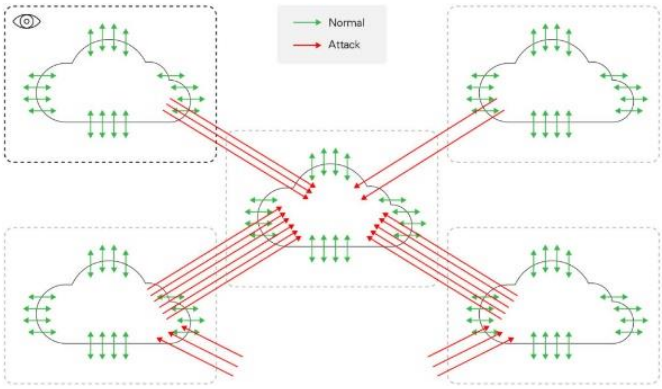


Figure 1: Attacking Scenario of the DDOSFlowgen [17]

The particulars of the traffic synthesized using DDoSFlowgen are explored in Table 1.

Table 1: The Particulars of The Traffic Flow Synthesized Using DDoSFlowgen

	Session's count	Request's Count	bandwidth in MB	sources of requests count
Attack Prone	39767	676042	61611	573
Normal	35610	676596	34206	332

4.2 The Classifier

The supervised learning strategy that widely denotes as machine learning or classification can perform many of classifiers exists. One among these classifiers is Adaboost, which denoted in contemporary literature as scalable and robust binary classifier un regard to classification assessment metrics such as fall-out, false alarming rate, accuracy, specificity, and sensitivity. Henceforth, the Adaboost has chosen for this experimental study as a classifier. The Adaboost learns from the proposed traffic flow features as metrics and performs binary classification that labels the target traffic flows of IOT as DDOS attack prone or benevolent

4.3 Experimental Setup

The experimental study carried on proposed model and performance analysis was carried by exhibiting the comparison of results obtained from proposed model TFFM, and other two contemporary models called DDOS Attack preventive measure [12], and MECShield [13]. The overall experiments carried in 5 folds, and statistical metrics "specificity, sensitivity, accuracy, false alarming, and fall-out" [19] are used to assess the performance of the models.

4.4 Experimental Results

The metrics sensitivity and specificity indicate the detection rate of attack prone sessions, and detection rate of benevolent sessions respectively. From the fivefold experimental study, the average of sensitivity, and specificity observed for proposed TFFM are (sensitivity: 0.982 ± 0.003, specificity: 0.967 ± 0.0061). The average of sensitivity and specificity observed for other two contemporary models are (sensitivity: 0.908 ± 0.025, specificity: 0.919 ± 0.012) and (sensitivity: 0.901 ± 0.024, specificity: 0.9113 ± 0.013) in respective order of MECShield, and DDOS Attack Preventive Measure (visual comparison of the sensitivity and specificity are depicted in Figure 2, and Figure 3 in respective order).

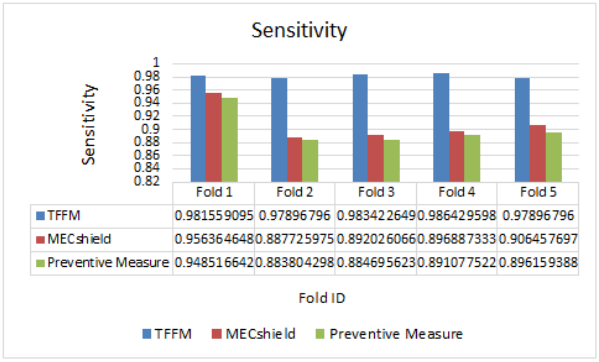


Figure 2: The comparison of the values obtained for the metric "sensitivity" under fivefold experimental study

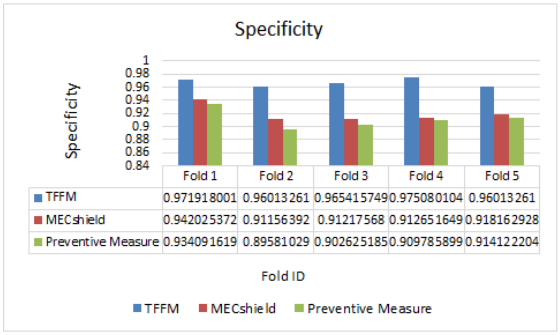


Figure 3: The comparison of the values obtained for the metric "specificity" under fivefold experimental study

The average of the values depicted for the metric "accuracy" under fivefold experimental study are visualized and compared in Figure 4. This metric denotes the classification ability of the target model for both classes (positive, and negative) of the given data. The depicted results in Figure 4 exhibiting that the average of classification accuracy observed for proposed model TFFM, and other two contemporary models called MECShield, and Preventive Measure are 0.974 ± 0.004, 0.914 ± 0.018, and 0.906 ± 0.018 respectively.

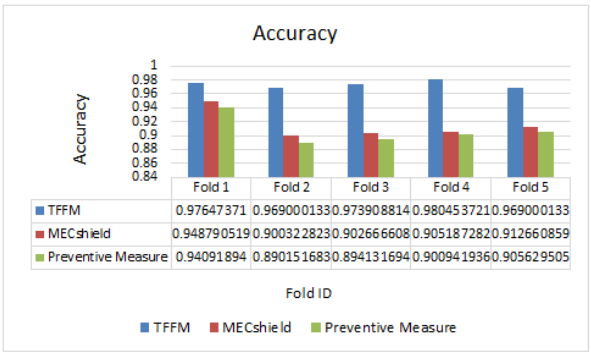


Figure 4: The comparison of the values obtained for the metric "Accuracy" under fivefold experimental study

The results obtained for other important metrics fall-out, false alarming rate are exhibited in Figure 5, and Figure 6 respectively.

The metric "fall-out" depicts the ratio of benevolent sessions described as attack prone, and the metric false alarming rate indicates the ratio of sessions miss-classified by the target model. The fall-out observed for the proposal TFFM is 0.0182 ± 0.003, the fall-out observed for MECShield is 0.0921 ± 0.025, and 0.099 ± 0.024 is the value of fall-out observed for Preventive Measure Technique.

The false alarming rates 0.026 ± 0.004, 0.086 ± 0.018, and 0.094 ± 0.018 are observed in the respective order of the proposed TFFM, and the other contemporary models MCSheild, and Preventive Measure technique.

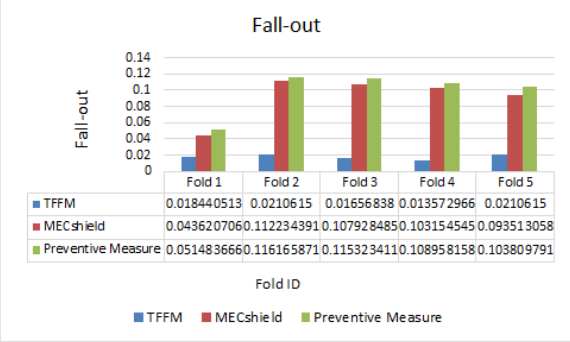


Figure 5: The comparison of the values obtained for the metric "fall-out" under fivefold experimental study

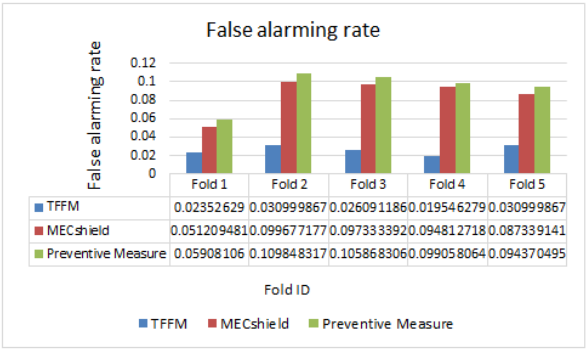


Figure 6: The comparison of the values obtained for the metric "false alarming rate" under fivefold experimental study

Table 2: The Average Metric Values and Their Standard Deviation Observed From Fivefold Experimental Study

Metric ↓	Model ↓	Avg., of five-fold ↓	std.dev ↓
specificity	TFFM	0.966536	0.006087
	MECShield	0.919316	0.011598
	Preventive Measure	0.911287	0.012999
sensitivity	TFFM	0.981869	0.002833
	MECShield	0.907892	0.025025
	Preventive Measure	0.900851	0.024253
accuracy	TFFM	0.973767	0.004416
	MECShield	0.913926	0.017918
	Preventive Measure	0.906355	0.018091
false alarming	TFFM	0.026233	0.004416
	MECShield	0.086074	0.017918
	Preventive Measure	0.093645	0.018091
fall-out	TFFM	0.018141	0.002846
	MECShield	0.09209	0.025021
	Preventive Measure	0.099148	0.024253

The overall results obtained for the assessing metrics indicating that the proposed model TFFM is scalable with minimal deviations that compared to the other two contemporary models. The deviation of metric values observed from the fivefold experiments and their comparisons are explored in Table 2.

5 Conclusion

The phenomenal growth in usage of the Internet of things (IoT) is observed to be incessant since recent past. The contemporary studies in recent literature evincing the growth of application-level DDOS attacks on IOT networks and the credible evidence in this regard is Mirai and other similar botnets, which continuously search for IOT devices and attempts to hold the control on IOT devices tracked to perform flow-based DDOS attack scenario. The conventional methods to defend these attacks are not optimal due to their weak counter measures. In this regard, the contributions of this manuscript endeavored to define a novel approach to detect and defend the flow based DDOS attack scenarios. The prosed model defined novel features of the traffic flow as metrics to differentiate the attack-prone and benevolent traffic flows. The experimental study was evincing the scalability and robustness of the proposed model that trains the classifier on proposed metrics. The comparative study was evincing that the proposed model outperformed the other contemporary models considered for the experimental study. The future research can extend this model to defend other intrusion factors.

References

[1] Filip Jelic, "Analysis: Record DDoS Attacks by Mirai, IoT Botnet," Deep.Dot.Web, Nov. 2016.
[2] Pettey, Christy. "The Internet of Things and the Enterprise." Gartner (2015).
[3] Ali, Sardar, et al. "On mitigating sampling-induced accuracy loss in traffic anomaly detection systems." ACM SIGCOMM Computer Communication Review 40.3 (2010): 4-16.
[4] Krebs, Brian. "Who makes the IoT things under attack." KrebsonSe-curity (2016).
[5] Douligeris, Christos, and Aikaterini Mitrokotsa. "DDoS attacks and defense mechanisms: classification and state-of-the-art." Computer Networks 44.5 (2004): 643-666.
[6] Mirkovic, Jelena, and Peter Reiher. "A taxonomy of DDoS attack and DDoS defense mechanisms." ACM SIGCOMM Computer Communication Review 34.2 (2004): 39-53.
[7] Tariq, Usman, ManPyo Hong, and Kyung-suk Lhee. "A comprehensive categorization of DDoS attack and DDoS defense techniques." International Conference on Advanced Data Mining and Applications. Springer, Berlin, Heidelberg, 2006.
[8] Jung, Jaeyeon, Balachander Krishnamurthy, and Michael Rabinovich. "Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites." Proceedings of the 11th international conference on World Wide Web. ACM, 2002.
[9] Tao, Yuan, and Shui Yu. "DDoS attack detection at local area networks using information theoretical metrics." Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on. IEEE, 2013.
[10] Misra, Sudip, et al. "An adaptive learning routing protocol for the prevention of distributed denial of service attacks in wireless

mesh networks." Computers & Mathematics with Applications 60.2 (2010): 294-306.
[11] Zhang, Fangjiao, et al. "Research on redundant channel model based on spatial correlation in IOT." International Conference on Trustworthy Computing and Services. Springer, Berlin, Heidelberg, 2012.
[12] Zhang, Congyingzi, and Robert Green. "Communication security in internet of thing: preventive measure and avoid DDoS attack over IoT network." Proceedings of the 18th Symposium on Communications & Networking. Society for Computer Simulation International, 2015.
[13] Dao, Nhu-Ngoc, et al. "Securing Heterogeneous IoT with Intelligent DDoS Attack Behavior Learning." arXiv preprint arXiv:1711.06041 (2017).
[14] Kohonen, Teuvo. "Essentials of the self-organizing map." Neural networks 37 (2013): 52-65.
[15] https://www.corero.com/resources/ddos-attack-types/mirai-botnet-ddos-attack.html
[16] An, Tae-Ki, and Moon-Hyun Kim. "A new diverse AdaBoost classifier." Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on. Vol. 1. IEEE, 2010.
[17] https://galois.com/blog/2017/04/simulating-ddos-attacks-ddosflowgen/
[18] Kolias, Constantinos, et al. "DDoS in the IoT: Mirai and other botnets." Computer 50.7 (2017): 80-84.
[19] Powers, David Martin. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).