Packet Filtering and Sampling for Efficient Slow Denial of Service Detection in Resource Scarce IoT Networks

Andy Reed
Sch of Computing & Communications
The Open University
Milton Keynes, United Kingdom
andy.reed@open.ac.uk

Laurence S. Dooley
Sch of Computing & Communications
The Open University
Milton Keynes, United Kingdom
laurence.dooley@open.ac.uk

Soraya Kouadri Mostefaoui
Sch of Computing & Communications
The Open University
Milton Keynes, United Kingdom
soraya.kouadri@open.ac.uk

***Abstract*—There has recently been considerable interest in automatic detection strategies for recognising application layer security threats such as *Hypertext Transfer Protocol* (HTTP) Slow *Denial of-Service* (Slow DoS) attacks in Internet of Things (IoT) networks. Most existing approaches however, fail to take cognisance of the substantial resource constraints imposed upon IoT environments, which limits the applicability and deployment of many Slow DoS detection mechanisms. This paper addresses this significant security threat for resource scarce IoT nodes and networks in proposing an accurate and computationally efficient approach to packet-based intrusion detection of HTTP Slow DoS activity. The paper both critically analyses and measures the impact of applying network attribute filtering and packet sampling to reduce the computational overheads on the resource constrained IoT Slow DoS detection node. The unique solution proposed uses a dataset synthesised from a live IoT environment comprising both legitimate and malicious network events in the form of legitimate HTTP traffic and Slow DoS attacks. Experimental results corroborate that combining filtering at the Border Router of only in-bound packets containing no TCP payload with a systematic packet sampling scheme at a sampling ratio of up to 1:64, the processing overheads on the detection node are significantly reduced. The novel contribution presented is a resource efficient solution, garnered by employing systematic sampling to seamlessly and accurately support selective attribute-based intrusion detection of HTTP Slow DoS attacks in IoT networks.**

***Index Terms*—Internet of Things, Intrusion Detection, Slow DoS, Sampling, Attribute Filtering, Systematic Sampling.**

I. Introduction

The emergent and increasingly pervasive nature of the IoT paradigm in information and communication technologies, means that IoT nodes, networks and systems are penetrating evermore diverse application fields from industrial, manufacturing, logistics and transportation to health care and home automation [1]. One of the main challenges in designing effective IoT solutions is application layer security, since while some IoT nodes may be equipped with the requisite capability to self-manage their protection, many IoT nodes are resource-scarce and so especially vulnerable to web-based malicious attacks. One major security threat to web-based IoT functionality is the Slow Hyper Text Transfer Protocol (HTTP) *Denial of Service* (or Slow DoS) attack, which leads to a significant degradation of service, and even at worst, a complete cessation of an application's ability to operate and deliver the desired client service [2]. HTTP is not the only application layer protocol used in the IoT domain. A review of contemporary IoT communications protocols in terms of vendors and platform providers, [3] highlights that as well as HTTP, *Constrained Application Protocol* (CoAP) and *Message Queuing Telemetry Transport* (MQTT) hold a large portion of the market sector. More recently, HTTP/3 has evolved as an effective means of improving communication speeds. By utilising *User Datagram Protocol* (UDP) in preference to HTTP, HTTP/3 affords a more lightweight alternative, though there are still numerous IoT nodes available that utilise HTTP over TCP [3], as evidenced in the CIC IoT 2023 [4], and HTTP IoT DoS dataset [5] utilised in this research. It is thus apparent that securing HTTP over TCP and web-based IoT nodes will, for the foreseeable future, continue to be of paramount importance.

While various detection systems have been proposed for identifying Slow DoS attacks, most focus upon post event analysis using *Machine Learning* (ML) [6] and *Artificial Intelligence* (AI) [7] techniques which require large datasets. The basis of many DoS detection systems is predicated on the total volume of network events necessary to provide a realistic snapshot of the environment, and realise a meaningful and accurate set of outcomes. Recent proposals [8], and [9] have sought to rationalise the dataset size by exploiting fewer attributes to render them more propitious to real time packet analysis.

All transmitting node packets display various characteristics, such as size, time sent and received, together with the source and destination information. It is these discrete *attributes* that are exploited in many HTTP DoS detection proposals, [10]. To address the innate resource node constraints in an IoT Slow DoS detection system, this paper critically investigates the resource impact of introducing suitable packet sampling schemes as a dataset reduction mechanism to support a real-time solution. The test dataset used comprises both

legitimate and malicious Slow DoS attack activity within a live IoT network, with a key requirement being that minimal node resources are expended. The novel contribution in this work is seamlessly embedding this packet sampling scheme into existing attribute-based Slow DoS detection models [8], without compromising detection accuracy.

The remainder of this paper is organised as follows: Section II defines the problem domain with Section III evaluating the suitability of different packet sampling schemes for dataset reduction. In Section IV, the evaluation environment is discussed, while Section V presents a critical results analysis of the packet sampling performance at different sampling rates. Section VI provides some concluding remarks and future work directions.

II. Background

Since Slow DoS attacks are designed to masquerade as legitimate client HTTP requests and responses, they negatively impact a detection node's ability to reliably identify malicious packets, leading to potential misclassification [10]. Slow DoS attacks exploit the thread based functionality of web servers, such as Apache, by initiating seemingly genuine connections. Once a TCP connection is established, the web server waits until the client application either completes the required tasks, or until a local server timeout value expires before closing the connection. It is specifically this connection orientated environment that Slow DoS attacks exploit, where multiple slow or incomplete connections lead to the web server becoming overloaded and unresponsive. While there have been several security modules developed for the Apache server, these offer only limited protection against Slow DoS attacks, [11]. DoS attacks targeting web-based servers are an increasing IoT security threat [12], so the need for a highly efficient, resource sensitive solution is paramount, with the main impetus being to lower the number of attributes required to facilitate reliable and accurate detection.

Real time Slow DoS detection requires traffic monitoring either by the edge node of an IoT network or via a dedicated monitoring node. An example of the former is the *border router* (BR) where all in-bound network traffic is processed and forwarded, though inspecting every in-bound packet for potential anomalies is resource intensive. This has led to alternative detection strategies being deployed to realise a more efficient solution [13].

As the principal role of the BR is packet forwarding, any sampling or inspection at this point naturally incurs a processing cost for every k^{th} packet. Thus, in an IoT context, careful consideration must be given to deploying the best sampling strategy and detection architecture [14].

On resource conservation of the edge node, [15] attempted to bridge the gap between a distributed DoS detection scheme for IoT and ML techniques by pre-processing and using sparse autoencoders to extract network event data, essentially data filtering before further ML classification. Whilst results are promising, their ML analysis is undertaken on a high specification computer utilising a Core i7 processor and 64GB RAM, so beyond the capacity of low resource IoT at a 1:1 ratio. Most dataset driven approaches focus their evaluation on measurable network attributes, like the size or length of a TCP/IP packet, the payload size and the protocol, essentially sampling the entire dataset. [16] observed a direct correlation in the performance overhead between IoT detection node resource consumption and the volume of network packets under inspection and their size. Interestingly, not all network attributes were equally weighted, so for instance the detection node incurred considerably greater processing costs with a large TCP payload i.e., a 1400-byte data transfer, compared with a typical TCP control packet which has a zero-byte payload. Thus, as well as the volume of attributes, cognisance of the attribute type needs to be factored into the packet filtering. To support research in testing malicious network activity, various publicly available datasets have been created. While ML approaches to *Intrusion Detection Systems* (IDS), often yield high accuracy, most intrusion detection datasets are incomplete and cannot be generalised to live environments, so ML classifiers are seldom applied in real world deployments. Another important point is that most datasets are inherently imbalanced due to the varied attack types often included. This contrasts with live environments where legitimate network events generally make up the predominate class. Researchers, such as [17] and [18] have identified class imbalance in datasets as a critical risk to the accuracy of ML approaches for IDS, and recommend further processing to nullify any imbalance. However, introducing such additional processing in an IoT context could have a detrimental impact on the detection node, and raise questions as to the applicability of synthetically balancing the sampled network events. On packet sampling, [19] noted that care should be taken as sampling has the potential to distort the output. The following section evaluates two approaches to the sampling of network attributes, and compares the results both pre and post sampling to identify any distortion.

III. Sampling Network Events

A comparative analysis of various sampling algorithms in [19], compared thirteen different schemes, with the trace files for each attack being measured based on 30% of the overall attack traffic. The two most seemingly resource efficient sampling schemes identified in [19] are thus critically evaluated here to identify the most appropriate sampling strategy to adopt for an efficient IoT network Slow DoS detection model.

A. Systematic Packet Sampling

Packets are sampled deterministically in this scheme, with every k^{th} packet being systematically counted and retained in a process controlled by an internal counter, so for a 16:1 sampling ratio, only every 16^{th} packet in the flow is retained. This simplicity means nominal processing overheads are incurred, proportional to the chosen sampling ratio. Although the term simple innately implies basic functionality, this is certainly not the case. In [20], while acknowledging this sampling is

a straightforward statistical approach, they effectively implemented systematic packet sampling as part of their work to develop an IDS formed on Bayesian-based trust management with sampling for IoT big data environments. Whilst this approach is viable for high volume, multi-dimensional network activity, the sequential and ordered nature can negatively impact the accuracy of the sampling, in for instance, scenarios where a short burst of network activity falls either side of a sample period, or where packets follow a prescribed pattern and arrive in short intervals.

B. Random Packet Sampling

As with systematic sampling, this is a deterministic sampling approach in which the scope of the randomness can be predefined and reduces the probability of overlooking network events that may generate a trivial packet stream, such as a reconnaissance attack by a port scan. A possible limitation however, is that for each process, a random number generating algorithm needs to run for each set of samples leading to a higher processing cost compared with systematic sampling. [21] discussed the appropriateness of randomness in detection schemes such as *heavy hitters*, where high volume traffic is present, and inculcated the need to trade-off between a desired accuracy and processing performance.

C. Comparing Systematic and Random Sampling

Comparing the respective random and systematic sampling performance in the context of the resource constrained IoT scenarios, the observations of [14] are important. The authors identified that random sampling will always be more computationally intensive and proportional to a probabilistic approach. It was also evidenced that random sampling can severely impact the accuracy of time-based or multi-point metrics such as end-to-end delay because of the lack of correlation of critical data-points. There is thus a significant likelihood that random sampling will impede the latency, node delay and delta time analysis employed for attribute based HTTP Slow DoS detection, which are core components used in [8], and [22].

IV. Evaluation Environment

For the sampling evaluation, the *HTTP IoT DoS dataset* was utilised [5], where responsibility for the sampling is undertaken by the BR, and given its storage constraints of 128MB DRAM and 64MB Flash memory, the sampling scheme had to be as efficient as possible. For this evaluation, the IoT detection node is represented by a Raspberry Pi model 4 board with 4GB RAM, 16GB Flash memory. All sensor nodes are more acutely resource constrained with only 18K RAM, 256KB of Flash and a 20MHz CPU. The dataset comprises 912,986 network events, in the form of TCP/IP packets captured in a live IoT network. Of which, 302,340 are in-bound including legitimate traffic, which in this evaluation are denoted as *Legitimate Nodes* (LN), while the Slow Read, Slow Get and Slow Post events are categorised as *Malicious Node* (MN) traffic. All in-bound network packets target the web server at HTTP port #80. The live HTTP IoT DoS dataset contains packets generated by all three Slow DoS variants including legitimate HTTP traffic, whilst limiting the dataset size to only 562MB. The dataset is in industry standard *Packet Capture* (PCAP) trace file format. Any set of associated network events, such as the three way handshake are measured in terms of a trace period.

Table I provides the cumulative total of in-bound only network events per node type, from which it is seen that combined MN events constitute 16.6% while LN events make up $\approx$ 50% of the dataset. The remainder of the dataset is made up of 610,000 high volume HTTP DoS attack packets, including web-server responses, along with operational and control packets. In comparing *in-bound only* traffic, an important point to stress is the respective number of LN and MN packets which has an acceptable class imbalance of 0.99%, at the pre-sampling stage, affording a negligible bias towards the majority LN class, as evinced in Table I. This mitigates the need for further processing to redress the impact of packet sampling, with the best post-sampling outcome being to either retain or reduce class imbalance.

TABLE I
CUMULATIVE NUMBER OF IN-BOUND NETWORK PACKETS GENERATED

Source traffic	# Packets	IoT dataset %
Total MN	151,058	16.6
Total LN	151,282	50.2

A. Dataset Reduction

To establish a ground truth for the sampling process, the evaluation concentrates on selecting a single attribute common to all nodes, namely the TCP application data payload [8]. The rationale for this choice is that TCP *segment length* (ls) is shared by all nodes during a TCP client server three-way handshake, with the most prevalent value being (ls) = 0 bytes. For packets where *ls = 0*, it is observed no TCP payload data is included, so only the header and TCP flag values comprise the overall *lp*. As seen in Eq. 1 the node generated throughput is calculated by the number of bytes transmitted over a discrete trace period $(\hat{x_t})$ in which these packets exist.

$$N_{th} = \left(\frac{N_{flags+headers}}{\hat{x}_t} \right) \tag{1}$$

In [8], byte values, along with the length of packets and segments were identified as key IP-based flow attributes applicable for sampling, so pre-sampling all in-bound TCP/IP packets where ls = 0 are filtered out. This selective attribute approach decreases the dataset size to be sampled to $\approx$ 1.00%, which is a substantial saving compared with full dataset driven ML approaches. The corresponding reduction and results displayed in Table II, reflect a marginal bias of 8.3% for LN. A crucial aim of the sampling experiments is to maintain where possible a sample bias as close to 8.3% as possible.

TABLE II
NUMBER OF TCP/IP PACKETS WHERE ls = 0 BYTES

MN Packets	101,727
LN Packets	120,322
Sample Bias (%)	0.83

V. SAMPLING RESULTS

It should be noted that the average packet length where *ls = 0* is 64 bytes which contrasts with the average packet length in the HTTP DoS dataset of 166 bytes. As such, there is an implication that focusing on *ls = 0*, will incur lower processing overheads and thus be more efficient. To validate this hypothesis, Eq. 2 and 3 illustrate the potential throughput on a per node basis, N_{th} for each packet processed during a conversation. This is achieved by observing a single TCP/IP conversation where TCP/IP flags and headers are calculated along with the TCP payload. In this example, a MN transmits a total of 1193 bytes, at an average rate of 21 bytes/s.

$$N_{th} = \left(\frac{N_{flags+headers}}{\hat{x}_t}\right) + \left(\frac{N_{ls}}{\hat{x}_t}\right) \tag{2}$$

$$N_{th} = \left(\frac{734 + 459}{56.4}\right) \tag{3}$$

However, by only sampling in-bound packets where *ls* = 0 bytes the throughput is reduced to an average of 13 bytes/s. As such, sampling where *ls = 0* bytes, not only acts to level out the class bias, but also reduces node processing requirements and bandwidth consumption. This approach imposes an average saving of 8 bytes/s per TCP/IP conversation, which is significant when considering that the HTTP DoS dataset generates an average rate of 168 KB/s. Therefore, based on this reduced dataset, both random and systematic sampling schemes can now be equitably compared to evaluate their respective sampling performance.

A. Random Sampling

Using the reduced pre-sampled IoT dataset, random sampling is firstly evaluated by observing whether the scheme retains the same class balance compared with the original data. Table III presents the random sampling results at different ratios at binary increments from 1:64 through to 1:512. Following sampling, the number of packets is respectively observed as 63.61% MN and 77.12% LN. This demonstrates the impact of this scheme on increasing the class bias for LN samples to a maximum of 0.80% and minimum of 0.91%, which is an acceptably balanced post-sampled dataset, so no further processing is needed for Slow DoS detection.

B. Systematic Sampling

In contrast, the systematic sampling results in Table IV reveal a less significant change in class balance to that of random sampling. While the ground truth of all in-bound packets with the ls attribute having a null value had an initial balance in favour of the LN after sampling at 1:64, it is only slightly lower i.e., from 0.99% to 0.98% which within tolerance of retaining a balanced dataset of attributes. In reflecting upon the sampling range from 1:128 to 1:512, the results confirm that 1:64 has the most empathy to balancing the key attribute samples. Importantly, the results reveal that across the 1:128 to 1:512 range there is a notable improvement over random sampling, with the corollary that sampling beyond 1:64 may be appropriate, given a benefit analysis of the nexus between accuracy of the data to be sampled and lower processing overheads.

TABLE III
RANDOM SAMPLING OF PACKETS WHERE ls = 0 BYTES

Node \ Ratio	1:64	1:128	1:256	1:512
Total MN	1403	746	372	188
Total LN	1750	831	405	209
% of samples	1.41	0.71	0.34	0.17
Sample bias (%)	0.80	0.89	0.91	0.89

TABLE IV
SYSTEMATIC SAMPLING OF PACKETS WHERE ls = 0 BYTES

Node \ Ratio	1:64	1:128	1:256	1:512
Total MN	1599	801	403	200
Total LN	1560	778	387	195
% of samples	1.42	0.71	0.35	0.18
Sample bias (%)	0.98	0.97	0.96	0.96

C. Post Sampling Reduction

It has been seen that packet based detection consumes resources directly proportional to the volume and size of packets being processed, so incorporating a sampling step affords a valuable prospect to lowering the resource overheads, concomitant with the aim of maintaining a computationally efficient detection model. Fig. 1 highlights the trend lines which reflect a common reduction in packet values for each source node, where pre (a) and post-sampling (b) the ratio of packets for LN and MN are comparable in terms of the overall packet volume, which is critical for maintaining the same class balance as pre-sampling. Maintaining a comparable ratio post sampling affords a lower overhead approach to the analysis of Slow DoS and LN attributes used in [8] and similar approaches where the comparable volume of malicious to legitimate attributes are crucial.

TABLE V
COMPARING IN-BOUND-PACKET PERCENTAGES

Source	Packets (%)	Post-Sampling (%)
Total MN	50.0	49.4
Total LN	50.0	50.6

Table V reveals that for the cumulative number of in-bound packets, (see also Table I), the reduction of 1:64 post systematic sampling retains a similar percentage of packets per node in each of the three Slow DoS variants, with similar reductions observed in the respective numbers of MN and LN.

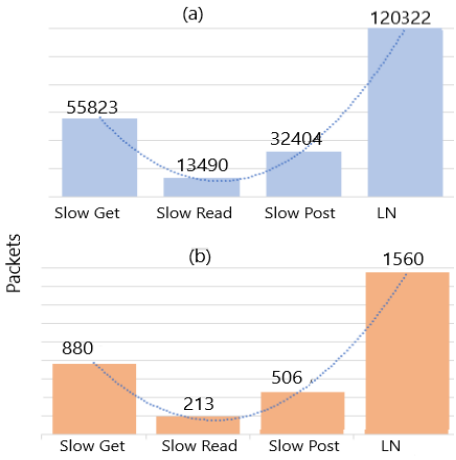

Fig. 1. Trend lines for pre-sampling (a) and 1:64 post-sampling (b)

Although the LN and MN overall percentage reductions are comparable, only 1.9% fewer Slow Get packets were collected, compared to an increase of 1.9% for Slow Read, and 0.7% for Slow Post. This indicates that by systematically sampling only in-bound packets where *ls* = 0 at a ratio of 1:64 only requires 0.3% of the dataset, which equates to 0.2 MB packets, which is a significant improvement on full packet inspection. These results corroborate the verdict that a systematic approach to dataset sampling of MN and LN network attributes can augment the performance of existing packet based Slow DoS detection models in terms of lowering operational overheads by sampling fewer packets whilst retaining detection accuracy. Sampling thus affords an attractive addition to embed with existing IoT Slow DoS detection scenarios.

D. Post Sampling Accuracy

In [8], Slow DoS traffic was identified through *Packet Length* (*lp*) analysis, where packets of a specified length could be used to identify potential Slow DoS traffic. Packets which fell within a specified byte range were labeled as *Candidate MN*. As such, to evaluate the accuracy of the sampling scheme employed here, all packets where $lp \in \{66, 74\}$ are compared pre, and post-sampling. From the results in Table VI it is apparent that Slow DoS generates a high percentage of packets in the {66, 74} byte range, compared to only 0.8% generated by the LN.

TABLE VI
COMPARATIVE PACKET LENGTH IN BYTES WHERE $lp \in \{66, 74\}$

	Pre-Sampling %	Post-Sampling %
MN	99.2	99.8
LN	0.8	0.2

By identifying packet lengths where $lp \in \{66, 74\}$ as potentially malicious affords a detection accuracy of 98.6%, which is based on the pragmatic judgement that packets in this range are Slow DoS generated. Employing systematic sampling gives a marginal improvement to 99.8% accuracy post sampling, so demonstrating that applying this scheme at a sampling ratio of 1:64 is a viable Slow DoS detection solution. The corresponding resource overheads upon the sampling node are critically analysed in the next section. It was also seen that sampling beyond 1:64, incurred a slight reduction in accuracy, implying there is a trade off between resources saved and overall detection accuracy.

E. Sampling Resource Observations

Both observed and recorded values are based on a pragmatic traffic profile over a period of 120 secs, during which 10 legitimate HTTP requests per sec along with Slow DoS traffic are generated. This profile reflects the traffic scenario where the BR encounters the heaviest network load. In these experiments, sampling was conducted on the BR, with the results showing that for random sampling, the CPU load on the BR increased by an average 6% over the observed average baseline (None) in Fig. 2, while for systematic sampling the load increased by 4%. When no packet sampling is applied, as expected then the BR incurs the greatest CPU load.

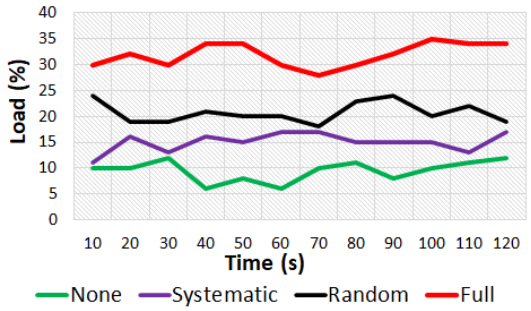

Fig. 2. BR CPU load comparison

For any node, the CPU processing and memory overheads are directly impacted by the volume of packets passing through the network interface, as such the higher the number of *packets per second* (PPS), the greater the resources consumed. For resource scarce IoT node processing this has a critical impact on CPU load and associated energy requirements. Fig. 2 reveals that CPU load variance is directly correlated with packet processing, so resource conservation can be considered in terms of node *energy consumption (EC)*. Given each node must process packets (*pk*), including request (pk_{req}), and response (pk_{res}) packets, this means $EC = pk_{req} + pk_{res}$. By employing a packet sampling strategy, the processing overheads and *EC* are thereby significantly reduced to just 0.3% of the original dataset.

In Section V, the dataset consisted of packets generated at a rate of 168 KB/s, which equates to an average of 222.5 PPS during a trace period of 3317 s. Thus by applying systematic sampling at 1:64, the processing requirements reduce to = 0.95 PPS, which represents an overall reduction in processing and memory overheads of 99.62%, whilst retaining an acceptable level of balance comparable to the original dataset. This equates to $912986/3317 = 252.2$ PPS processed at the capture interface.

In this research, the sampled network attributes of the IoT Slow DoS dataset, where in-bound packets exhibit values of $ls = 0$ bytes are representative of the original dataset, so the results can be generalised. As such this approach addresses the common problem of retaining an acceptable sample balance to ensure equity and accuracy of the sampling scheme, along with the numerical results. Using a live IoT Slow DoS attack dataset with a sampling ratio of 1:64 lowered the processing requirements upon BR by $\approx 99.7\%$ while concomitantly retaining an acceptable level of balance with the original dataset. The post systematic sampling results highlight that Slow DoS detection proposals based on selective attribute filtering and analysis in [8] and [17], as well as full packet capture could be enhanced by adopting a systematic sampling scheme to reliably reduce resource utilisation and consumption of each respective detection proposal. Although the findings have focused on resource saving in an IoT context, this approach to selective attribute filtering and sampling can be extended into other network domains.

In the context of resource scarce IoT environments, any additional cost sustained by the sampling scheme must pragmatically balance the efficiency and accuracy of the detection model with the extra overheads incurred. The evaluation results presented have compellingly demonstrated that embedding pre-sampling into an existing real-time Slow DoS detection model in a live IoT network is a beneficial design option with significant savings achievable.

VI. Conclusion

This paper has critically investigated the impact of different IoT network packet sampling schemes as a reduction mechanism to provide accurate detection of HTTP Slow DoS attacks, whilst ensuring an energy efficient approach, suitable for resource scarce environments. By selectively filtering and sampling only in-bound TCP/IP packets where the TCP payload attribute is equal to zero bytes, presents a significant lowering of resource overheads for IoT nodes in the detection of Slow DoS traffic. Experimental results compellingly demonstrate that by only filtering packets with a zero-byte payload, combined with systematic sampling at a 1:64 ratio on the IoT BR, markedly reduces computational overheads on the IoT Slow DoS detection node compared to either full or random sampling, while upholding high detection accuracy. This strategy is especially attractive for real-time IDS in resource constrained IoT environments. Future work will focus upon developing a quantitative framework of the resource savings realised by embedding systematic sampling of network attributes on IoT detection nodes, as an integral component in a computationally lightweight Slow DoS detection model.

References

[1] S. Roy, J. Li, B. J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Future Generation Computer Systems*, vol. 127, pp. 276–285, Feb 2022.

[2] T. Hirakawa and T. Takata, "The Trade-Off Between the False-Positive Ratio and the Attack Cost of Slow HTTP DoS," *Advances in Intelligent Systems and Computing*, vol. 1264 AISC, pp. 225–237, Aug 2020.

[3] L. De Nardis, A. Mohammadpour, G. Caso, U. Ali, and M. G. Di Benedetto, "Internet of Things Platforms for Academic Research and Development: A Critical Review," *Applied Sciences 2022, Vol. 12, Page 2172*, vol. 12, no. 4, p. 2172, Feb 2022.

[4] UNB, "CICIoT2023," 2023, available at:. [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2023.html

[5] A. Reed, "HTTP DoS Dataset in PCAP format for Wireshark," 2021, [Accessed on 04 January 2022]. [Online]. Available: https://ordo.open.ac.uk

[6] I. A. Alnuman and M. Al-Akhras, "Machine Learning DDoS Detection for Generated Internet of Things Dataset (IoT Dat)," *2020 2nd International Conference on Computer and Information Sciences, ICCIS 2020*, Oct 2020.

[7] N. Garcia, T. Alcaniz, A. González-Vidal, J. B. Bernabe, D. Rivera, and A. Skarmeta, "Distributed real-time SlowDoS attacks detection over encrypted traffic using Artificial Intelligence," *Journal of Network and Computer Applications*, vol. 173, pp. 1084–8045, Jan 2021.

[8] A. Reed, L. S. Dooley, and S. Kouadri, "A Reliable Real-Time Slow DoS Detection Framework for Resource-Constrained IoT Networks," in *2021 IEEE Global Communications Conference: IoT and Sensor Networks (Globecom2021 IoTSN)*. Madrid: Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1–6.

[9] E. Cambiaso, M. Aiello, M. Mongelli, and I. Vaccari, "Detection and classification of slow DoS attacks targeting network servers," in *ACM International Conference Proceeding Series*. Association for Computing Machinery, Aug 2020.

[10] M. Sikora, T. Gerlich, and L. Malina, "On Detection and Mitigation of Slow Rate Denial of Service Attacks," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, vol. Oct-2019. IEEE Computer Society, Oct 2019.

[11] M. Catillo, A. Pecchia, and U. Villano, "No more DoS? An empirical study on defense techniques for web server Denial of Service mitigation," *Journal of Network and Computer Applications*, p. 103363, Apr 2022.

[12] L. Ma, Y. Chai, L. Cui, D. Ma, Y. Fu, and A. Xiao, "A Deep Learning-Based DDoS Detection Framework for Internet of Things," in *IEEE International Conference on Communications*, vol. 2020-June. Institute of Electrical and Electronics Engineers Inc., Jun 2020, pp. 1–6.

[13] A. Mudgerikar, "Edge-Based Intrusion Detection for IoT devices," *ACM Trans. Manage. Inf. Syst*, vol. 11, 2020. [Online]. Available: https://doi.org/10.1145/3382159

[14] J. M. C. Silva, P. Carvalho, and S. R. Lima, "Inside packet sampling techniques: exploring modularity to enhance network measurements," *International Journal of Communication Systems*, vol. 30, no. 6, pp. 159 – 165, Apr 2021.

[15] A. Tabassum, A. Erbad, A. Mohamed, and M. Guizani, "Privacy-Preserving Distributed IDS Using Incremental Learning for IoT Health Systems," *IEEE Access*, vol. 9, pp. 14 271–14 283, 2021.

[16] J. Arshad, M. A. Azad, M. M. Abdellatif, M. H. Ur Rehman, and K. Salah, "COLIDE: A collaborative intrusion detection framework for Internet of Things," *IET Networks*, vol. 8, no. 1, pp. 3–14, 2019.

[17] C. Calvert, C. Kemp, T. Khoshgoftaar, and M. Najafabadi, "Detecting slow http post dos attacks using netflow features," *The thirty-second international FLAIRS conference*, pp. 387–390, 2019.

[18] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network," *IEEE Access*, vol. 8, pp. 32 464–32 476, 2020.

[19] H. H. Jazi, H. Gonzalez, N. Stakhanova, and A. A. Ghorbani, "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling," *Computer Networks*, vol. 121, pp. 25–36, Jul 2017.

[20] W. Meng, "Intrusion Detection in the Era of IoT: Building Trust via Traffic Filtering and Sampling," *Computer*, vol. 51, no. 7, pp. 36–43, Jul 2018.

[21] S. Li, L. Luo, D. Guo, Q. Zhang, and P. Fu, "A survey of sketches in traffic measurement: Design, optimization, application and implementation," 2021.

[22] E. Cambiaso, M. Aiello, M. Mongelli, and I. Vaccari, "Detection and classification of slow DoS attacks targeting network servers," *ACM International Conference Proceeding Series*, 2020.