BOND: Efficient and Frugal DL Model Co-design for Botnet detection on IoT Gateways

Himanshu Gandhi
himanshu.gandhi@cse.iitd.ac.in
IIT Delhi
India

Misha Mehra
mishamehra@gmail.com
DRDO
India

Vinay J. Ribeiro
vinayr@iitb.ac.in
IIT Bombay
India

ABSTRACT

A botnet is a network of devices infected by the same malware, acting as a single entity and controlled by a botmaster. They are the biggest cybersecurity threat to carry out large-scale attacks from spamming, ransomware, data exfiltration, and denial-of-service attacks. Lightweight IoT devices without traditional security mechanisms have become favorite victims and agents to carry out botnet attacks. In our work, we seek to detect botnet-infected IoT nodes.

This paper presents BOND, a frugal Deep Learning analysis of network traffic for detecting IoT devices infected with botnet(s), correctly classifying Zero-Day attacks and newer benign traffic. BOND is designed considering the constraints of IoT gateways and betters the F1 score of standard benchmark ML algorithms and State-of-The-Art method - Kitsune, by at least 10%, with under 1 millisecond inference time and less than 150 KB of model memory.

This paper also presents labeled data-set 27-Botnet spanning 27 IoT botnet families and ten different IoT devices. We believe, it is the first data set with a separate zero-day component.

CCS CONCEPTS

• **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; **Software security engineering**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

Internet of Things, Botnet Detection, Neural Networks, Datasets

ACM Reference Format:
Himanshu Gandhi, Misha Mehra, and Vinay J. Ribeiro. 2021. BOND: Efficient and Frugal DL Model Co-design for Botnet detection on IoT Gateways. In *The First International Conference on AI-ML-Systems (AIMLSystems '21), October 21–23, 2021, Bangalore, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3486001.3486237

1 INTRODUCTION

Connected DVR cameras bringing down the Internet in Eastern USA, or shutting down more than a million websites, including WikiLeaks by attacking hosting provider OVH or Connected cameras cutting off the entire country of Liberia from the world by shutting down the Internet, or coffee machine asking for ransom ? [3, 26, 34] These are not figments of imagination or scenes from a Sci-Fi thriller but attacks of IoT botnets - compromised devices like cameras, routers, wearables, Smart Appliances infected by botnet malware. IoT botnets differ from traditional botnets since IoT devices are Always-On, easily exploited (e.g., default passwords, no security, wireless communication, rarely monitored and maintained, Linux software monoculture), and allow a more extensive network of compromised devices. Their sparse and vast geographical spread further makes traditional defense mechanisms redundant.

A recent security report [21] observes 87% botnets attacks target Linux/IoT platforms, decline of older Windows-based botnets, 500% rise in botnet attacks (Mozi botnet being chief culprit) since October 2020, especially due to Work-From-Home in Covid-19 pandemic times. Thus, IoT botnet detection is an important and relevant problem with increasing IoT deployments.

Despite extensive work over the last 15 years [10, 16–18, 31], increase in IoT Botnet attacks indicates that:

- **Botnet detection is an open problem**: Tracking down botnets is still an unsolved problem, as evidenced by increasing botnet attacks.
- **Accuracy is a misleading measure**: Botnet traffic is much less than the benign traffic, and thus a case of highly imbalanced classes [22, 35], making F1 score better metric than accuracy [1].
- **Model resilience under Zero-day attacks**: Zero-day cyber-attacks cause extensive damage. Hence, classification of unfamiliar traffic profiles should be an integral part of data sets and model evaluation.

The nature of IoT and botnets require that any solution for detecting botnet adhere to the following requirements:

(1) **Lightweight** computation, and minimal memory overhead so that the normal network operations are unhindered.
(2) **Generalizable** to correctly classify newer types of benign and malicious network traffic.
(3) **Low Inference time** for early detection of infected devices, without slowing down IoT gateways packet processing with a typical packet delay under 10 ms [44].

The above mentioned observations and IoT network specific requirements are our guiding principles for curating data-set focused on IoT botnets and for designing A classification model for detecting botnet infected devices in IoT network by analysing the network logs of IoT gateway.

In this paper, we propose **BOND (BOtNet Detector)** as a Deep Learning(DL)-based solution, optimized to reduce over-fitting, decrease prediction time, decrease model memory and generalize well in terms of learning newer patterns. BOND is evaluated against

AIMLSystems '21, October 21–23, 2021, Bangalore, India

ACM ISBN 978-1-4503-8594-7/21/10…$15.00
https://doi.org/10.1145/3486001.3486237

our dataset 27-Botnet, and zero-day dataset 27-Botnet-0Day. It has an inference time almost 2.8X faster than the State-of-the-Art method - Kitsune [31]. BOND is suitable for deployment on IoT gateways to detect infected devices.

This paper's contributions are summarized below:

- **BOND (BOtNet Detector)** a lightweight, scalable DNN based solution deployable on IoT gateways. To the best of our knowledge, we are the first to show an ML model generalizable for 27 different botnet families and capable of detecting zero-day attacks better than competing techniques. We describe the model (Section 2), its evaluation (Section 3) and compare it with other related work (Section 4).
- Our **dataset 27-Botnet** incorporates network logs of 18 days, of diverse set of IoT devices and botnets. Zero-day botnet based dataset, 27-Botnet-0Day, is also the first-of-its-kind. We present the dataset in Section 2.1, its Zero-day component (Section 2.2) and then contrast it with other datasets (Section 4).

2 FRAMEWORK

2.1 Data-set

Experimentation and benchmarking against a realistic data, which mimics the real-world traffic and network anomalies, is crucial for evaluation of any detection method [4]. Perception of botnets as only agents causing DDoS attacks has led to data-sets focusing on network attacks limiting their suitability for botnet detection.

Most of existing botnet data-sets also suffer from problems of generality and realism. They mostly include data from usually two or three botnets reflecting specific botnet behaviour. Lack of real IoT background traffic further limit their effectiveness to handle the novel threats that botnets pose. Generalization of detection models by testing them against unseen traffic, is not considered [4]. We seek to fill this gap by providing a Zero-Day dataset component containing new traffic.

To address this challenge of a relevant, diverse and realistic data-set, we curate a botnet data-set with real botnet network logs and real-world IoT logs. Our work focuses on the network communication of the botnets, irrespective of the botnet working stage (infection, confirmation of infection to C&C, downloading attack module, malware execution, C&C communication, attack).

Table 1: Complete list of IoT Botnet used to create 27-Botnet. Mirai and Satori used in zero day scenario, were never used with other 25 bots while creating the 27-Botnet dataset.

S. No.	Botnet Name	S. No.	Botnet Name
1	Alina	14	Moose/Elan
2	Amnesia	15	OMG
3	Andromeda	16	Optima
4	BlackEnergy	17	Umbra
5	Dark Comet	18	Slingshot
6	Dirtjumper	19	Spike/ Dofloo
7	Dridex	20	Tsunami
8	ExploitKit/ DNSChanger	21	VertexNet
9	Floki Bot	22	Wicked
10	Hydra	23	Hide 'N Seek (HNS)
11	Illusion	24	Pony
12	BashDoor/ BASHLITE/ gafgyt/ Torlus/ Qbot	25	Zeus Panda Banker/ Panda Banker/ Zeus
13	Leet		
Zero Day Attack			
Mirai Satori/ Okiru/ Satori-Okiru			

2.2 How did we construct our dataset ?

Network communication logs of 260 botnet binaries spanning 27 botnet families (Table 1) collected from VirusShare [1] and VirusTotal [2] repositories. Benign network logs,of 18 days, from IoT Analytics project at UNSW [20] have been used for the packet logs for benign IoT applications. Table 2 lists the IoT devices whose network communication has been utilized for the dataset.

These logs were divided into 2 sets. Logs of 25 botnet families and 8 IoT devices were used for formulating benchmark data-set, 27-Botnet. Logs of 2 botnet families and 2 IoT devices (not in 27-Botnet)

[1] VirusShare Malware Binary Repository: https://virusshare.com/.
[2] VirusTotal Malware Binary Repository: https://www.virustotal.com/gui/home/upload.

Table 2: IoT Devices (for Benign Logs) in dataset. The logs of these IoT devices simulate real world scenarios.

S. No.	Device	Device Type
1	WEMO Motion Sensor	Motion Detector for Smart lights
2	WEMO Power Switch	Smart Light Switch
3	Samsung Camera	Smart Camera
4	TP Link Plug	Smart Plug to control devices
5	Netatmo Camera	Smart Camera
6	Philips Hue Bulb	Smart Bulb
7	AmazonEcho	Smart Speaker
8	chromecast	Smart Media Streaming device
9	ihome	Bluetooth Speaker and Alarm
10	lifx	Wi-Fi enabled LED smart lighting

Table 3: 27-Botnet dataset. Overlay of network traffic of randomly sampled botnet binary over benign device communication scenario

Dataset	Size (number of data points)	Remarks
`27-Botnet`	Test 15.42 M, Train 39.75 M, Total 55.17 M	25 botnet families
`27-Botnet-0Day`	2.7M	Mirai & Satori botnets and benign IoT data unseen in Test and Train

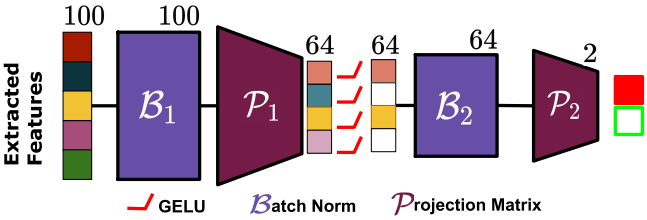

Figure 1: Architecture of BOND

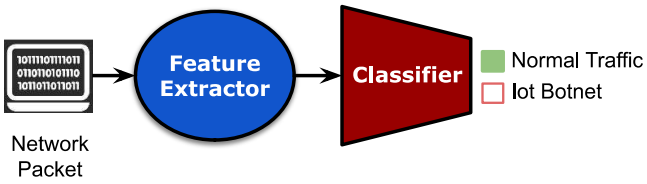

Figure 2: BOND Pipeline

were used for making the Zero-day data-set - 27-Botnet-0Day, as shown in Table 3.

Mirai and Satori were chosen as Satori has borrowed significant code from Mirai and is thus very similar to Mirai but different from the rest.

In each case, the malicious and benign traffic were replayed using tcpreplay with random periodicity of botnets' communication and interwoven into a single data-set. Ratio of Botnet traffic to benign traffic is kept close to the real-world ratio of 1:99 or 1:100. The random periodicity of botnets was observed in case of some botnets, for making detection of communication with C&C server and other members of the botnet harder to detect.

2.3 Network Packet - Features

Kitsune's libraries [3] are used for converting the 27-Botnet dataset's network logs (in pcap format) to CSV format and for feature extraction. As a result, benchmarking is done on the same feature set and the same packet processing as used by Kitsune.

Input feature computation at $O(1)$ memory and processing cost by Kitsune, to design low-footprint model are also instrumental in choosing Kitsune for benchmarking. Re-use of Kitsune's library for statistical techniques of damped incremental statistics for aggregation of traffic data and incremental update of summarized feature data also reduce the model footprint need by BOND. Use of same 115 traffic statistics (23 features over 5 time instants) for capturing temporal snapshots of packet information from sender and receiver perspective, and re-use of feature processing also further aids the like-by-like comparison of the various methods.

2.4 BOND- Botnet Detection

Notations: BOND is a highly frugal bot detection which is carefully designed using state-of-the-art deep learning modules. BOND can be trained on any network flow packet (p) with hand crafted features $\mathbf{f}_p$ having an indicator $\mathbf{y}_p = [1, 0]^\top$ if p is from a IoT Botnet else $[0, 1]^\top$. Figure 1 shows the overall architecture of BOND.

[3]Kitsune libraries: https://github.com/ymirsky/KitNET-py.

2.5 Architecture

BOND uses batchnorm β_1 to address for ICS in input packet features f followed by linear project of $\beta_1(\mathbf{f})$ to a 300-dimensional output space using $\mathbf{f} \times 300$ projection matrix $\mathcal{P}_1$ followed by a non-linearity (σ) to compute packet p encoded representation $\mathbf{g}_p$. Here σ_1 is GeLU as it does not suffer from the problem of vanishing gradients. For classification $\mathbf{g}_p$ is normalized using batchnorm layers β_p and linearly projected using 300×2 matrix ($\mathcal{P}_2$) to 2 output units corresponding to malicious and benign respectively with sigmoid (σ_2) as the activation function. Computation of output probabilities $\mathbf{o}_p$ from features $\mathbf{f}$ can be given as follows:

$$\mathbf{g}_p = \sigma_1(\mathcal{P}_1 \cdot \beta_1(\mathbf{f})) \quad (1)$$

$$\mathbf{o}_p = \sigma_2(\beta_2(\mathcal{P}_2 \cdot \mathbf{g}_p)) \quad (2)$$

2.6 Training, Prediction and Prevention using BOND

Model is regularised by using a dropout layer after every GeLU non-linearity at the time of Training. Dropout probability is set to 0.5. Model parameters are trained using Binary Cross Entropy using Adam optimizer with step size of 0.001 and maximizing the following objective function:

$$\beta_1^*, \beta_2^*, \mathcal{P}_1^*, \mathcal{P}_2^* = \underset{\beta_1, \beta_2, \mathcal{P}_1, \mathcal{P}_2}{\text{argmax}} \frac{\Sigma_{p=1}^{N} \left(\mathbf{y}_p \cdot \mathbf{o}_p + (1 - \mathbf{y}_p) \cdot (1 - \mathbf{o}_p)\right)}{N} \quad (3)$$

At the time of prediction dropout layer probability is set to 0 and probabilities of traffic being malicious or benign traffic for feature vector f of packet p is give by Equation 2. Please refer to Figure 2 for an illustrative example of prediction pipeline. In the figure system with an IoT bot is shown in top left corner of the image. Feature vector f for this bot infected system is detected as malicious by BOND. While other network traffic is successfully passed through the detector. The anomalous node can then be found out by finding the intersection of the IP addresses common to the flagged packet strides. The packets from the infected IP address can then be subject to techniques like null-routing etc., so that the rest of the nodes can be saved from botnet infection.

3 EXPERIMENTS AND QUALITATIVE ANALYSIS

3.1 Evaluation

Most of the research papers in botnet scenarios focus on overall accuracy of the system. However, ratio of benign to malicious data can in IoT scenarios be up to 1:99 therefore any model can trivially achieve 99% accuracy by simply predicting all the traffic as benign. In order to have a meaningful comparison this paper focuses on F1 score of the malicious data (higher is better). Furthermore, we also compared methods according to memory usage and prediction time as the model should not use much memory as well and it should have prediction in micro seconds so that network can quickly identify IoT network nodes infected by botnets and take measures without affecting network bandwidth. Methods were also compared

with performance on zero-day attacks which is one of the essential feature in any botnet detection algorithm.

3.2 Benchmark Algorithms

BOND has been compared with **state-of-the-art Kitsune** as well as extreme lightweight models like Logistic Regression (LR), Isolation Forest (IF) and Gaussian Mixture Model (GMM). Code for Kitsune was provided by the authors. Unfortunately, we could not compare BOND with BotHunter, BotSniffer and BotMiner as code was not publicly available.

All the methods were run on on Ubuntu 16.04 on Intel i7-6700 @ 3.4GHz CPU, 16 GB RAM. For a fair like-by-like comparison, each algorithm is trained using the same features.

3.3 Results

We compared our method with other competitors on the basis of following metrics: a) Accuracy in terms of F1 score, b) Prediction time and c) Model size. BOND's main competitor is Kitsune as it was designed for low latency and faster training as well as prediction. Therefore we have compared performance of each method on basis of prediction time and model size as well. Please refer to Figure 3 and table 4 for detailed comparison of model size and prediction time We have also compared BOND and Kitsune on complete dataset. Furthermore, we have also compared BOND's performance on zero-day attacks. Please refer to Figure 3 for performance on the proposed dataset and performance on zero-day attacks.

F1 Score: BOND is slightly worse than RF but more than 20% points better than all other models on 27-Botnet. We outperform the second best model by at least 5% on zero-day detection on the 27-Botnet-0day dataset.

Prediction time: BOND is extremely frugal in terms of prediction time. Most of the operations are matrix multiplications which can be easily parallelised. As a result BOND can be up to 2 orders of magnitude faster as compared to Kitsune and al. Note that other methods can be an order of magnitude faster in prediction but at the cost of F1 score.

Model size: Model size for BOND is significantly less than Kitsune while giving State-of-the-Art accuracy. Less model size is directly associated with less memory consumption as well as power usage. Lower model size is preferable with IoT devices like smart watches, having very low battery backup and restricted working memory.

Zero-day attacks: With increasing threat of new bots everyday it is very essential for methods to be able to generalise for previously unseen bot. BOND can significantly outperform Kitsune on Zero day attacks by 5%. This indicates that BOND can easily generalise to previously unseen bot attack and can give state-of-the-art F1 Score. Please refer to Figure 3 for more details.

Accuracy as an in-appropriate metric In IoT scenarios, botnet traffic can be significantly less than the actual internet traffic (1:99 ratio). Though easy to get very high accuracy, real world performance can be seen by considering F1 score.

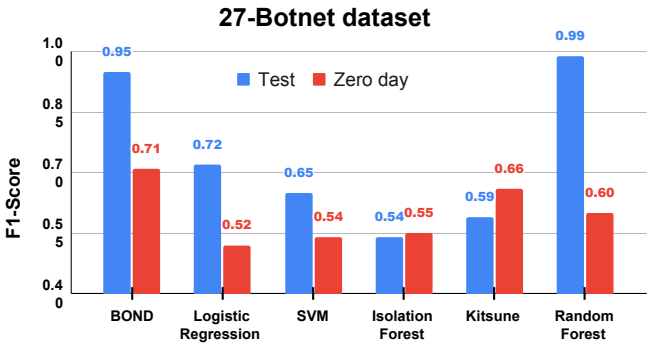

Figure 3: BOND can significantly outperform Kitsune by 35% for F1 Score and by 5% on zero day

Table 4: BOND's comparison with Baseline algorithms with respect to training time, prediction time and model size on 27-Botnet dataset.

Method	Training Time (hr)	Model Size (kB)	Prediction Time (μs)
BOND	3.46	101.16	2.8
Logistic Regression	0.16	1.48	0.73
SVM	0.11	1.72	0.62
Isolation Forest	0.80	771.19	56.16
Kitsune	0.03	154.00	722.42
Random Forest	0.62	26611.13	o.70

3.4 Qualitative Analysis

BOND can give state-of-the-art results on real world scenarios of zero-day as well as it can be easily deployed in any network infrastructure to detect botnet activities. This can be attributed to the superior encoding of network packet p by the bot as $\mathbf{g}_p$ as described earlier in section 2.5. To visualize the embedding quality we used standard dimension reduction tool, t-SNE and plotted in 2D space. Please see Figure 4 for the plot. For the sake of clarity we selected only 30,000 randomly selected botnet packets and 30,000 randomly selected benign packets. It can be clearly seen from the plot that BOND could learn to embed benign data (marked in blue) such that its embeddings can be significantly different from the botnet embeddings. Another observation which can be made from the plot is most of the benign data points have significant overlap with each other. This can be attributed to the fact that botnets are known to re-use codes from each other. To conclude BOND can learn superior quality embeddings which in-turns power its classifier to give state-of-the-art results.

We understand that organizational need for a botnet detector can be very different as a highly secure network like a Govt. Agency would require a high recall for botnet detection and a streaming service would require high precision. One comes at the cost of other. If we were to keep only high recall most of the packets would be flagged as malicious and therefore could lead to considerable lag while if we keep only high precision we could speed up the network traffic but we risk detecting botnet. It should be noted that BOND have 2 output units (One for malicious and other one for benign) instead of just one (like Kitsune). Therefore it can always adjust its precision and recall according to the scenario. Nevertheless we have kept weight for both the units 0.5 for the sake of this paper and reported the numbers.

Table 5: Data-sets Used in Community for Botnet work. 27-Botnet consists of significantly large number of bots which makes it unique in terms of complexity. 27-Botnet provides network logs of zero day attack for algorithms to benchmark their performance in real world settings.

S. No.	Dataset	Focus	IoT Logs	Real World Logs	No. of Real Botnet Families
1	CAIDA-Conflicker [2]	Botnet attack	No	Yes	1 botnet - Conflicker
2	CTU-13 [13]	Botnet attack scenarios	No	Yes	7 (non-IoT)
3	ISCX-Botnet 2014 [5]	Botnet detection in Internet	No	No	16 botnets (only 1 IoT botnet)
4	IoT-DS [7], [6]	IoT botnet detection	Yes	Yes	7
5	**27-Botnet**	IoT botnet detection	Yes	Yes	27

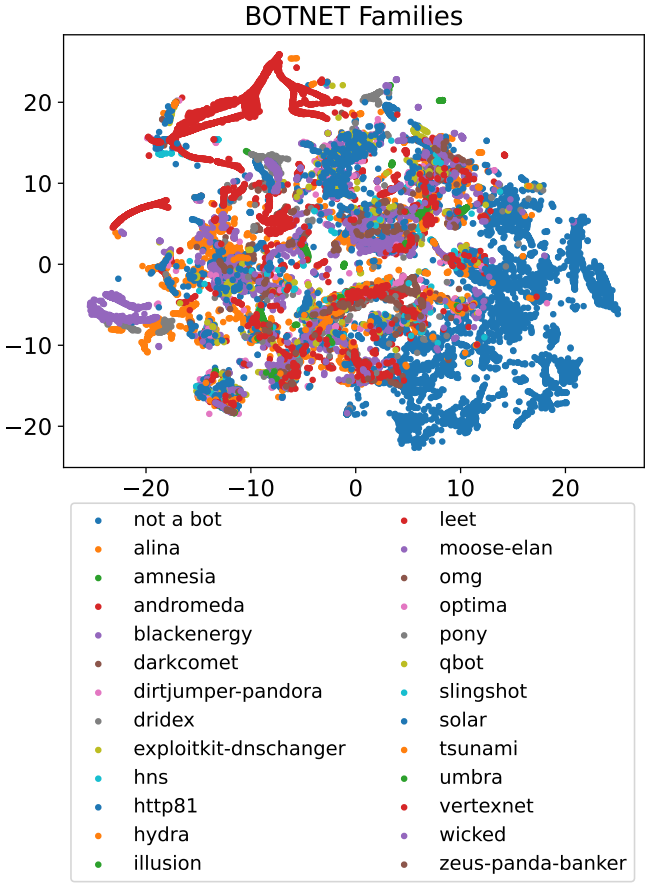

Figure 4: Qualitative analysis for BOND. BOND could encode Benign and malicious network traffic such that there is a significant separation between the embeddings of both types of traffic.

4 RELATED WORK

4.1 IoT Botnet Dataset

This subsection describes the work done in the areas of IoT botnet data sets. Table 5 summarizes the popular data sets being used research in Botnet detection, and compares it with our dataset.

DARPA-98, KDD-99 data-sets were seminal in IDS benchmarking. CAIDA Conflicker dataset [2] has real network logs of 3 days of 2008 Conflicker botnet attack. However their relevance has reduced due to emergence of newer threats.

CTU UNiversity's 2011 labelled data-set CTU-13 [13] has mix of traffic of 7 botnets and normal benign traffic. ISCX-Botnet 2014 dataset [5] is generated by combining three non-overlapping generic botnet data-sets - ISOT [46], ISCX 2012 IDS [38] and Malware Capture Facility Project [15] data-sets, but lacks any IoT specific logs.

UNSW-NB15 [33], TUIDS [9], [8] are IDS data-sets with real-world network traffic with network attack logs, and are not focused on botnets. Koroniotis et al used Ostinato tool [41] and MQTT broker to Smart Home and botnet data for BoT-IOT dataset [24].

IoT-DS [7], [6] by Bezerra et al in 2018 and 2019 contains labelled data collected from 7 real botnet families and emulated behavior of IoT based surveillance cameras and media centers.

Most data sets cater to IDS or have far less diverse malicious and benign data. They do not distinguish between generic malware, network attacks and botnets. Also, DoS and DDOs network attacks can also be caused by different tools and are not always by botnets, which makes network attacks different from detecting botnets. Thus, most data-sets focus on generating attack scenarios, making them unsuitable for the problem.

Our work is thus, the first work to incorporate network logs of such a large and diverse set of IoT specific botnets, mixed with a large variety of IoT device traffic.

4.2 IoT Botnet Detection

Numerous solutions have been proposed for botnet detection and its various related problems like detection of command-and-control (C&C) channels/ servers, infected hosts, Domain Generated Algorithms (DGAs). Some of them like BotHunter [17] have even been deployed in the field for several years now.

Detection solutions have employed techniques varying from white-list based audit (Heimdall [19]) , correlation studies from network traffic (BotHunter [17], BotMiner [16], BotSniffer [18]), two-level correlation based anomaly score [11], Hierarchical clustering (CoCoSpot [12]), Spectral density analysis of network logs (PsyBog [25]) and Machine/ Deep Learning (DISCLOSURE [10], Kitsune [31], N-BaIoT [30] and many more). Beigi et al [5] focus on the features selection for Botnet detection in their 2014 paper. Most of these works demonstrate very high accuracy (often more than 95%) in detecting botnets.

ML/ DL based techniques have been widely employed for NIDS (Network Intrusion Detection Systems, which are the superset of botnet detection systems and detect all kinds of attacks and

coming through the network) [14], [24], [32]. Researchers have used CNN [43], [39] and RNN [29], [37] based architectures which lead to state-of-the-art accuracy. However, high memory consumption by RNN and LSTM [37] and CNN's high power consumption (40W)[43], [39] makes them unsuitable for IoT.

Simliarly, auto-encoders though used extensively, have huge potential risk of wrongly classifying a new device or traffic category as malicious, as they are trained to very specific devices in the enterprise network setting.[40]. However these ML/ DL solutions do not consider the resources available for deployment, nor consider classifying new unseen patterns.

Lever at al in their 2017 IEEE S&P paper [27] demonstrate that monitoring network traffic allows malware to be detected much before attacks are launched.

Given the similarity in communication techniques used in IoT and Ad-Hoc networks, we also note that ad-hoc network based works [46], [23], [28],[36], [42] , [45], discuss multi-layer or hierarchical detection techniques. They highlight that traditional techniques IDSs like Snort and Bro, Firewalls and Deep Packet Inspection (DPI) do not work well, due to low capability nodes. They also stress on the need to have low prediction times, low memory footprint, and caution about mis-classifying environmental and mobility related misbehaviour as anomalies indicating botnet activity.

Thus, despite numerous ML/ DL based solutions being proposed, there have not been many light weight botnet detection systems respecting the constraints of IoT systems to be able to be usable.

5 CONCLUSION

We presented **highly frugal Neural Network BOND (BOtNet Detector) using state-of-the-art deep learning modules for detecting IoT network nodes infected with botnets** and evaluated its detection performance. BOND performed significantly better than competing techniques - **2X faster in classification, 25% lesser in memory consumption and 4% better zero-day detection of unseen botnets and benign data traffic, while being efficient enough to run on single core CPU**.

This paper also presents a new labelled IoT Botnet data-set **27-Botnet** curated by overlay of network communication of 27 actual IoT botnets and 10 diverse IoT sensors, spanning 18 days, which we believe is **one of the world's largest data-set closest to a real world IoT network infected with botnets**.

We believe BOND is a deployable, useful, lightweight, scalable method for defending IoT networks from botnets and our labelled dataset 27-Botnet would be helpful in advancing research on botnet detection in IoT networks.

ACKNOWLEDGMENTS

We would like to express our appreciation to Anshul Mittal, for his technical assistance, useful insights and constructive comments. We thank our friends Madhulika Mohanty and Aditya Ahuja who took the time to read the article and give us very useful feedback. We would also like to thank Sonal Swarnima, Radhika Dharwadkar and Ritu Gandhi for their constant support, encouragement and countless discussions, without which this work may not have seen the light of the day.

REFERENCES

[1] 2020. https://developers.google.com/machine-learning/crash-course/classification/accuracy
[2] 2020. http://www.caida.org/data/passive/telescope-3days-conficker dataset.xml
[3] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 1093–1110. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis
[4] Adam J Aviv and Andreas Haeberlen. 2011. Challenges in experimenting with botnet detection systems. In *Proceedings of the 4th conference on Cyber security experimentation and test*. 6–6.
[5] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani. 2014. Towards effective feature selection in machine learning-based botnet detection approaches. In *2014 IEEE Conference on Communications and Network Security*. 247–255. https://doi.org/10.1109/CNS.2014.6997492
[6] Vitor Hugo Bezerra, Victor Guilherme Turrisi da Costa, Sylvio Barbon Junior, Rodrigo Sanches Miani, and Bruno Bogaz Zarpelão. 2019. IoTDS: A One-Class Classification Approach to Detect Botnets in Internet of Things Devices. *Sensors* 19, 14 (2019), 3188.
[7] Vitor Hugo Bezerra, Victor G Turrisi da Costa, Ricardo Augusto Martins, Sylvio Barbon Junior, Rodrigo Sanches Miani, and Bruno Bogaz Zarpelao. 2018. Providing IoT host-based datasets for intrusion detection research. In *Anais do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*. SBC, 15–28.
[8] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. 2013. Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials* 16, 1 (2013), 303–336.
[9] Monowar H Bhuyan, Dhruba K Bhattacharyya, and Jugal K Kalita. 2015. Towards Generating Real-life Datasets for Network Intrusion Detection. *International Journal of Network Security* 17, 6 (2015), 683–701.
[10] Leyla Bilge, Davide Balzarotti, William Robertson, Engin Kirda, and Christopher Kruegel. 2012. Disclosure: detecting botnet command and control servers through large-scale netflow analysis. In *Proceedings of the 28th Annual Computer Security Applications Conference*. 129–138.
[11] Chia-Mei Chen and Hsiao-Chung Lin. 2015. Detecting botnet by anomalous traffic. *Journal of Information Security and Applications* 21 (2015), 42 – 51. https://doi.org/10.1016/j.jisa.2014.05.002
[12] Christian J Dietrich, Christian Rossow, and Norbert Pohlmann. 2013. CoCoSpot: Clustering and recognizing botnet command and control channels using traffic analysis. *Computer Networks* 57, 2 (2013), 475–486.
[13] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. 2014. An empirical comparison of botnet detection methods. *computers & security* 45 (2014), 100–123.
[14] Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security* 28, 1-2 (2009), 18–28.
[15] S García. 2013. Malware Capture Facility Project. CVUT University. https://agents.fel.cvut.cz/malware-capture-facility
[16] Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee. 2008. BotMiner: clustering analysis of network traffic for protocol-and structure-independent botnet detection. In *Proceedings of the 17th conference on Security symposium*. USENIX Association, 139–154.
[17] Guofei Gu, Phillip Porras, Vinod Yegneswaran, and Martin Fong. 2007. BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation. In *16th USENIX Security Symposium (USENIX Security 07)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/16th-usenix-security-symposium/bothunter-detecting-malware-infection-through-ids-driven
[18] Guofei Gu, Junjie Zhang, and Wenke Lee. 2008. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. In *NDSS*.
[19] J. Habibi, D. Midi, A. Mudgerikar, and E. Bertino. 2017. Heimdall: Mitigating the Internet of Insecure Things. *IEEE Internet of Things Journal* 4, 4 (Aug 2017), 968–978. https://doi.org/10.1109/JIOT.2017.2704093
[20] Ayyoob Hamza, Hassan Habibi Gharakheili, Theophilus A Benson, and Vijay Sivaraman. 2019. Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity. In *Proceedings of the 2019 ACM Symposium on SDN Research*. 36–48.
[21] Mina Hao. 2020. Botnet Trend Report 2019-6 - NSFOCUS, Inc., a global network and cyber security leader, protects enterprises and carriers from advanced cyber attacks. https://nsfocusglobal.com/botnet-trend-report-2019-6/
[22] Sarah Harun, Tanveer Bhuiyan, Song Zhang, Hugh Medal, and Linkan Bian. 2017. Bot Classification for Real-Life Highly Class-Imbalanced Dataset. 565–572. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.102

[23] Fabian Hugelshofer, Paul Smith, David Hutchison, and Nicholas J.P. Race. 2009. OpenLIDS: A Lightweight Intrusion Detection System for Wireless Mesh Networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking* (Beijing, China) *(MobiCom '09)*. ACM, New York, NY, USA, 309–320. https://doi.org/10.1145/1614320.1614355

[24] Nickolaos Koroniotis, Nour Moustafa, and Elena Sitnikova. 2019. Forensics and deep learning mechanisms for botnets in Internet of Things: A survey of challenges and solutions. *IEEE Access* 7 (2019), 61764–61785.

[25] Jonghoon Kwon, Jehyun Lee, Heejo Lee, and Adrian Perrig. 2016. PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Computer Networks* 97 (2016), 48–73.

[26] Malwarebytes Labs. 2020. Chaos in a cup: When ransomware creeps into your smart coffee maker - Malwarebytes Labs. https://blog.malwarebytes.com/ransomware/2020/10/smart-coffee-maker-ransomware/

[27] C. Lever, P. Kotzias, D. Balzarotti, J. Caballero, and M. Antonakakis. 2017. A Lustrum of Malware Network Communication: Evolution and Insights. In *2017 IEEE Symposium on Security and Privacy (SP)*. 788–804. https://doi.org/10.1109/SP.2017.59

[28] Wenjia Li and Anupam Joshi. 2008. Security Issues in Mobile Ad Hoc Networks -A Survey. *White House Papers Graduate Research in Informatics at Sussex* 17 (01 2008).

[29] C. D. McDermott, F. Majdani, and A. V. Petrovski. 2018. Botnet Detection in the Internet of Things using Deep Learning Approaches. In *2018 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN.2018.8489489

[30] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. 2018. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing* 17, 3 (2018), 12–22.

[31] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *NDSS* 5 (2018), 2.

[32] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli. 2019. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. *IEEE Communications Surveys Tutorials* 21, 1 (Firstquarter 2019), 686–728. https://doi.org/10.1109/COMST.2018.2847722

[33] N. Moustafa and J. Slay. 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942

[34] Lily hay Newman. 2016. The web-shaking Mirai botnet is splintering—but also evolving. https://www.wired.com/2016/11/web-shaking-mirai-botnet-splintering-also-evolving/

[35] Akinyemi Oyelakin. 2020. Towards Building an Improved Botnet Detection Model in Highly Imbalance Botnet Dataset-A Methodological Framework. Volume 3 (03 2020), 33–38.

[36] J. Parker, A. Patwardhan, and A. Joshi. 2006. Cross-layer analysis for detecting wireless misbehavior. In *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, Vol. 1. 6–9. https://doi.org/10.1109/CCNC.2006.1592977

[37] Wan-Chen Shi and Hung-Min Sun. 2020. DeepBot: a time-based botnet detection with deep learning. *SOFT COMPUTING* (2020).

[38] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, and Ali A Ghorbani. 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *computers & security* 31, 3 (2012), 357–374.

[39] Kapil Sinha, Arun Viswanathan, and Julian Bunn. 2019. Tracking temporal evolution of network activity for botnet detection. *arXiv preprint arXiv:1908.03443* (2019).

[40] Giacomo Spigler. 2017. Denoising Autoencoders for Overgeneralization in Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[41] P. Srivats. 2020. Ostinato Packet Generator. https://ostinato.org/

[42] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C. Y. Tseng, and T. Bowen. 2005. A general cooperative intrusion detection architecture for MANETs. In *Third IEEE International Workshop on Information Assurance (IWIA'05)*. 57–70. https://doi.org/10.1109/IWIA.2005.1

[43] Ahmad MN Zaza, Suleiman K Kharroub, and Khalid Abualsaud. 2020. Lightweight IoT Malware Detection Solution Using CNN Classification. In *2020 IEEE 3rd 5G World Forum (5GWF)*. IEEE, 212–217.

[44] HaiYang Zhang, Yanli Ma, Benzhen Guo, and Zhe Xu. 2020. Light-weight IoT Gateway with Embedded Web Server. In *2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Vol. 1. 134–137. https://doi.org/10.1109/IHMSC49165.2020.00038

[45] Yongguang Zhang, Wenke Lee, and Yi-An Huang. 2003. Intrusion Detection Techniques for Mobile Wireless Networks. *Wirel. Netw.* 9, 5 (Sept. 2003), 545–556. https://doi.org/10.1023/A:1024600519144

[46] David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, and Dan Garant. 2013. Botnet detection based on traffic behavior analysis and flow intervals. *Computers & Security* 39 (2013), 2–16.