EasyChair Preprint

№ 2512

Performance Analysis Of Network Anomaly Detection Systems in Consumer Networks

Darsh Patel and Rahul Raman

January 31, 2020

Performance Analysis Of Network Anomaly Detection Systems in Consumer Networks

P. Darsh[1] and R. Raman[2]

[1,2]School of Computing Science and Engineering , Vellore Institute of Technology

***Abstract*—There has been a significant increase in the amount of smart home appliances such as intelligent thermostats, internet connected baby monitors, lights being used, these devices have been a target for a number of cyber-attacks till date. Using only fundamental network information such as Source and Destination Ports, Packet Sizes, TCP Flags, Time between subsequent packets, necessary features can be extracted to detect the aforementioned anomalies. This paper aims analyze some critical operating system performance metrics on detecting such anomalies. It also consists of the taxonomy of various approaches to classify anomalies and detailed description on capturing and cleaning network packets.**

***Index Terms*—Anomaly Detection, IoT, Machine Learning**

I. Introduction

It has been projected that there would be a huge increase in the number of IoT devices by 2020, many of these IoT devices are fundamentally insecure. [1] With this increase in number of IoT devices, There has also been an uptick in the number of attacks targeting such IoT devices.

Sweeping attacks on key internet services around the world have been launched with botnets powered by IoT devices such as security cameras and wireless routers, topping 1.1 terabits per second [2]. These IoT devices basically act as network probes from where attackers can hear and see what is going on inside the network infact resources of these devices can also be used to mine cryptocurrencies, etc.

Network anomaly detection is a wide topic which boasts numerous research, articles, surveys as well as books. [3]–[5] major part of it aims at thwarting attacks on commercial networks. [6] There is very little research done on how the hardware performance for various anomaly detection methods.

Outlier detection has been a huge part of research by the statistical community [3], [7]–[16], but with the numerous recent advancement in machine learning, it has been playing a notable role in anomaly detection too.

While it seems attractive theoretically, this technique has it's own set of disadvantages, such as the intrinsic complexity of the system to the high false positive rates, determining which event triggered the alarm, etc. these problems need to be addressed before wide adaptation of anomaly based detection systems.

The primary aim for this paper is to provide the readers with a proof of concept system for detection of anomalies in consumer networks and also evaluating the performance metrics for the most commonly available hardware boards

A. Distinction from existing research

Existing research on anomaly detection algorithms for network traffic has been targeted towards commercial scenarios. Out of the multitude of attacks consumer network face, only a few are a real concern for consumer networks.

For example, Distributed denial of service(DDoS) Attacks are generally aimed at high profile websites and essential infrastructure for fortune 500 companies, and many researchers have taken up the challenge to mitigate such attacks, also the fact that a very large portion of the traffic for such DDos attacks come from 'Zombie' devices taken hostage by nefarious actors are found in consumer networks.

Detecting anomalies on a consumer network can have the following challenges:

- Lack of Network infrastructure: Most consumer networks, do not have managed switches with a dedicated monitoring port, thus packet capturing tools need to be installed on users computers or a low power device connected the network to monitor network.
- Public datasets are not available: There exist publicly available, labeled datasets with millions of records and attack types have been made available for research purposes. [17] Infact, research has beed done on the quality of such datasets. But, no consumer network data has been collected or published for obvious privacy reasons.
- Less data points: Consumer networks do not have multiple subnets containing thousands of devices generating a multitude of network traffic over various protocols which can be analyzed for patterns. For example: In a typical commercial environment, One can analyze traffic on the DC and then analyzing user login data or the average overall bandwidth use which correlate work hours. This cannot be done on a typical consumer network.

B. Anomaly based intrusion detection

It refers to finding uncommon patterns or irregularities in network traffic that do not adhere to the expected behavior these, patterns are often referred as anomalies or outliers. [3]

C. Tools Used for this research

The well known Network sniffer `wireshark` [18] was utilized to intercept network data which was then stored as a PCAP file. This PCAP file which consists a lot of unnecessary data was then processed, compressed, and converted to CSV files using `netcap` [19]. `matplotlib` [20], which is a

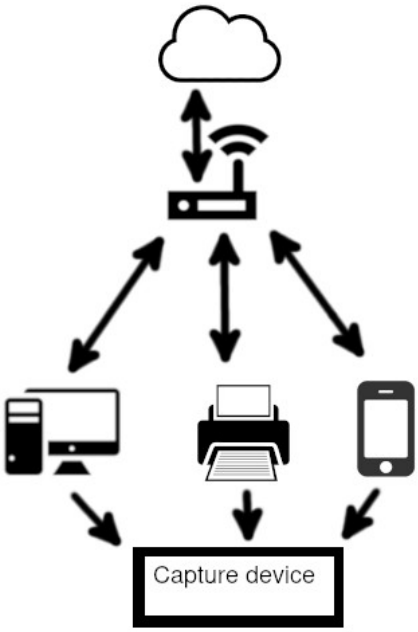

Fig. 1. Capture device listening for capture data sent via programs installed on network devices

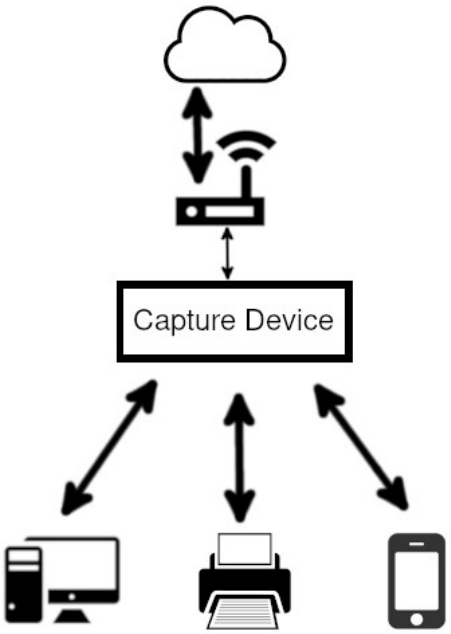

Fig. 2. Capture device acting as a network bridge

python library was then used to plot the graphs identifying useful features in the data. Python Library `pandas` [21] was then utilized to extract these features from the data. Machine Learning library `scikit learn` [22] was utilized for training ML Models.

II. Processing data and Packet Capturing

As mentioned earlier in I-A Packet capturing and processing has been one of the most daunting tasks in this area research. More so, for networks without the dedicated hardware to do so. Typical Consumer Routers do not have a 'monitor port' like most commercial switches do. Thus a low power dedicated device such as a RaspberryPi can be used for capturing and storing network traffic. A Similar network monitor device has been mentioned in [23]. A capture device can be setup in two ways inside a local network

- As a server which listens to packet capture data sent via dedicated programs installed on individual devices in the network. This configuration is not suitable for the purpose of this paper as this paper focuses on monitoring traffic from IoT devices, programs which capture network traffic and send that to a server cannot be installed on these devices. Refer Fig. 1
- As a network Bridge which sits between the router and devices. This allows the capture device to capture all the traffic across the network. Refer Fig. 2

For the purpose of this paper a total of 1.5GB of network data was captured from 8 different consumer networks each containing various IoT devices such as , Smart Lights, Voice assistants and IP Cameras. The captured data was then manually cleaned for training the models.

TABLE I
DATA-POINTS COLLECTED

Sl.	Feature Name	Feature Description
1	Protocol	Layer 3 Protocol: IP,UDP,TCP
2	Source IP	Packet Source IP Address
3	Destination IP	Packet Destination IP Address
4	Source Port	Packet Source Port
5	Destination Port	Packet Destination Port
6	Frame Length	Length of the captured frame
7	Time	Time of said packet
8	TTL	Packet Time to Live
9	TCP Flags	Flags: RST,SYN,ACK,FIN

TABLE II
FEATURES EXTRACTED FROM I

Feature	Feature Description
1	Ratio Between TTL and Payload Size
2	Average Payload Size in N seconds
3	Average TTL in N seconds
4	Number of DNS queries in T Seconds
5	Average Number of ACK packets in T seconds
6	Average Number of RST packets in T seconds
7	Average Number of SYN packets in T seconds
8	Average Number of FIN packets in T seconds
9	Average time between two frames over T seconds
10	Average Number of IPV4 Frames in T Seconds

Also, Synthetic network traffic for generating anomalies such as internet sweeps, performing DDoS attacks using devices on the network was also generated. Network traffic from well known malware such as Mirai whose source code is readily available was also simulated.

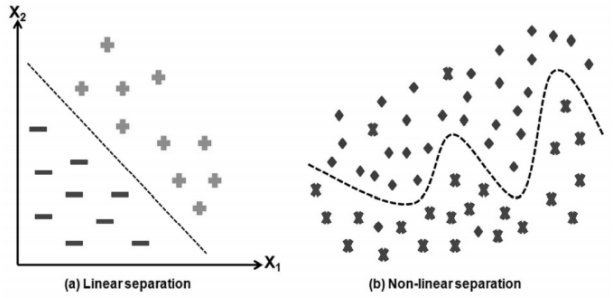

Fig. 3. Classification based anomaly detection from

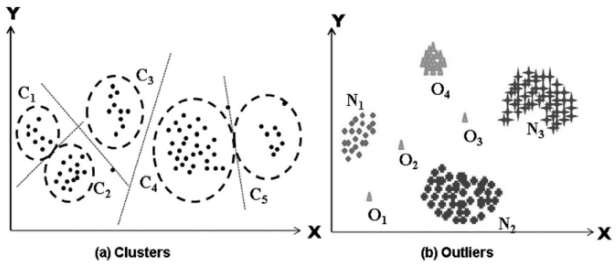

Fig. 4. Clustering and outlier based anomaly detection

III. Anomaly Detection Methods

Detailed introduction to various approaches is out of scope of this paper, existing literature can be reffered to. [24]

A. Statistical anomaly detection

'An anomaly is an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed' [25].
Thus, the occurrences with a low probability of being generated are anomalies. Prime advantage of this technique over the others is that it does not require 'Prior Knowledge' of the network's normal activity [5] ,thus it can provide accurate results about anomalous activity [25]. One promising approach to detect network anomalies is an Entropy based approach, Entropy is a measure of the uncertainty or randomness associated with a random variable. If it was more random it contains more entropy. [26]

The primary drawback of this approach is that attackers can 'train' the detection model until the traffic is considered as normal according to the statistical model. This approach also requires a lot of training data which is difficult to obtain in a consumer setting.

B. Supervised classification based anomaly detection

This approach is a supervised learning approach where model is trained using a labeled dataset, which then tries to classify new data into categories based on the training data. "Linear classification tries to find a line between the classes" [25], but the "classification boundary may be non-linear" too [25] as seen in Figure 3. These techniques have a low false positive ratio subject to suitable thresholds. [25] The prime drawback for these is that they're highly dependent and biased on the training data and thus generally cannot identify anomalies it hasn't seen before, which defeats the purpose of this research.

C. Clustering and outlier based novelty detection

1) Unsupervised approach: Fundamentally, grouping data into various sets of similar objects is called clustering. This has been represented in Figure 3(a) In Anomaly detection the primary assumption made is 'Larger clusters are normal'. [25] The rest of the clusters can be considered as anomalies. In figure 3(b), the points which do not fit into any of the clusters, are considered outliers (anomalous data-points).
It is also worth nothing that unsupervised learning approach works with unlabeled data. Due to most unsupervised learning models using both outlier detection and clustering, the computation complexity can be quiet high. [25]

2) Semi supervised approach: Semi supervised learning is prevalent in scenarios where there is very little anomalous data available to train the model but non-anomalous data is readily available, thus it's trained on a 'single class' of data and then detects 'novelties'.
Thus, this is also commonly referred to as novelty detection. [27] One-Class Support Vector Machine classifiers (OCSVM) are favourable in case of anomaly detection, in this approach as they do not require pre-labeled data sets which are expensive or difficult to obtain. [28]

The Performance and Accuracy of the models One Class Support Vector Machine and Isolation Forest has been compared in this paper. The models were trained on the features mentioned in Table II using the captured data.
The results have been mentioned in Tables III and IV and have been discussed in section IV

TABLE III
PERFORMANCE OF FEATURES USING ONE CLASS SVM (WINDOWS)

Feature	Correctly Ident. (attack)	Correctly Ident. (normal)	Falsely Ident. (attack)	Falsely Ident. (normal)	Avg. DR	Avg. FPR
1	100%	95.2%	0%	4.80%	97.60%	2.4%
2	100%	83.63%	0%	16.37%	91.81%	8.18%
3	100%	93.28%	0%	6.72%	96.64%	3.36%
4	80.24%	71.95%	19.76%	28.05%	76.09%	23.90%
5	90.10%	88.56%	9.90%	11.44%	89.33%	10.67%
6	88.10%	93.5%	11.90%	6.50%	90.8%	9.2%
7	76.23%	93.1%	23.77%	6.90%	84.66%	15.33%
8	91.85%	77.12%	8.15%	22.88%	84.48%	15.51%
9	92.4%	78.3%	7.6%	21.70%	85.35%	14.65%
10	84.5%	78.4%	15.50%	21.60%	81.45%	18.55%
Avg.	90.342%	85.304%	14.696%	9.658%	87.823%	12.177%

TABLE IV
PERFORMANCE OF FEATURES USING ONE CLASS SVM (UNIX)

Feature	Correctly Ident. (attack)	Correctly Ident. (normal)	Falsely Ident. (attack)	Falsely Ident. (normal)	Avg. DR	Avg. FPR
1	100%	92%	0%	8%	96.00%	4%
2	100%	82.24%	0%	17.76%	91.12%	8.88%
3	100%	94.3%	0%	5.7%	97.10%	2.85%
4	82.35%	70.96%	17.65%	24.04%	72.65%	20.84%
5	80.19%	89.59%	19.81%	10.41%	84.89%	15.11%
6	88.23%	94.12%	11.76%	5.8%	91.17%	8.78%
7	74.36%	94.3%	25.64%	3.7%	84.33%	14.67%
8	100%	75.81%	0%	24.19%	87.90%	12.1%
9	92.4%	77.1%	7.6%	22.9%	84.75%	15.25%
10	100%	78.3%	0%	21.7%	89.18%	10.85%
Avg.	91.753%	84.872%	8.246%	14.42%	87.909%	11.333%

IV. ANOMALY DETECTION RESULTS

All the models were optimized to use the best possible hyperparameters. The dataset is randomly split into training and test sets with a 80:20 ratio. Anomalous data-points are tested directly on the trained model.

Anomalous detection rate is defined as the fraction of anomalous data-points classified as anomalies from the total number of anomalous data-points the classifier was tested upon, denoted by `DR(a)`

$$\text{DR(a)} = \frac{\text{No. of anomalous data-points marked positives}}{\text{Total No. of anomalous data-points}}$$

Non Anomalous detection rate is defined as the fraction of non-anomalous data-points not classified as anomalies from the test dataset split. denoted by `DR(na)`

$$\text{DR(na)} = \frac{\text{No. of non-anomalous data-points marked negatives}}{\text{Total No. of non-anomalous data-points}}$$

Anomalous False positive rate is defined as the fraction of anomalous data-points marked as negatives from the total number of anomalous data-points the classifier was tested upon.`FPR(a)`

$$\text{FPR(a)} = \frac{\text{No. of anomalous datapoints marked negatives}}{\text{Total No. of anomalous data-points}}$$

Non Anomalous detection rate is defined as the fraction of non-anomalous data-points classified as anomalies from the test dataset split. `FPR(na)`

$$\text{FPR(na)} = \frac{\text{No. of non-anomalous data-points marked as positives}}{\text{Total No. of non-anomalous data-points}}$$

One Class Support Vector Machine

One Class SVM outperforms unsupervised models such as Isolation Forest, Also, similar to Isolation forest, optimizing the hyperparameters helps overcome the problems due to imbalanced classes.
One Class SVM requires a small sample size to train the model and proves to be very accurate in most of the cases. [29]
It is also worth noting that a semi-supervised learning model with high accuracy requiring a small sample size to train deems to be an ideal model for anomaly detection.

Unix Vs. Windows

It is clearly evident from the results of the tests that detecting network anomalies for both the operating systems are the same , given a 5% margin for errors. This can be due to the fact that the underlying implementation of common protocols such as SSH, FTP, etc is the same, regardless of the OS it runs on.

TABLE V
EMBEDDED DEVICE SPECIFICATIONS

Board	CPU	Cores	Frequency	Memory	OS
RaspberryPi	ARM Cortex-A53	4	1.2GHz	1GB LPDDR2	Debian Kernel 4.4.34
MacBook Pro	Intel i5	4	2.7GHz	8GB LPDDR3	macOS 10.13.2
Intel UP	Intel Atom z8350	5	1.4GHz	4GB DDR3L	Ubuntu kernel 4.15.0
Intel NUC	Intel i7	4	3.5GHz	8GB DDR4	Ubuntu kernel 4.15.0

V. HARDWARE ANALYSIS

As discussed previously, a packet analysis device must be placed inside the network which would capture, clean and analyze the network data. It becomes essential that such a device is low powered but also capable to perform the tasks discussed previously.

It is essential for the analysis device to have low power consumption, noise and enough performance to run basic machine learning models. For the scope of this survey, the following devices were tested.

A. Experiment Setup

The time metrics were measured using the system clock on individual systems, TensorFlow and Caffe are used were compiled on each system with the compile time flags '-funsafe-math-optimizations -ftree-vectorize' for benchmarking performance The Memory metrics were measured using the 'ps' command. Power consumption for the devices were measured using Texas Instruments INA219 power monitor IC by attaching it to the powerline. The power consumption was measured over the period of 10 minutes for idle consumption , and for 5 minutes over multiple inferences using the caffe model for under load metric.

TABLE VI
POWER CONSUMPTION

Metric	RaspberryPi	MacBook Pro	Intel UP	Intel NUC
Idle (w)	1.9	13.1	12.7	14.3
Under Load(w)	10.8	33.4	31.2	32.5

B. Memory Footprint

SqueezeNet on Caffe2 Model consumes avg. 4000MB on the Intel NUC, avg. 2000MB on both MacBook Pro and Intel UP while it consumed only 532MB on RaspberryPi.

C. Power Consumption

The RaspberryPi Board powered by a standard 5V 2A power USB power supply consumes the least power when it's and under load, while both the MacBook Pro, Intel NUC and Intel UP have similar power consumption.

D. Hardware Summary

The RaspberryPi being a low powered board provides subpar Inference times on the standard caffe model, but consumes drastically less power in general than the other devices. Intel UP provides an optimal mix of power consumption and performance if theres adequate utilization of it's resources. Thus, for a dedicated network traffic analysis device, cheap low powered boards such as RaspberryPi or Intel's UP can be utilized according to the amount of data to be analyzed.

TABLE VII
TIMING METRICS ON CAFFE2 MODEL

Metric	RaspberryPi	MacBook Pro	Intel UP	Intel NUC
Import Package(s)	1	0.3	0.8	0.7
Load Model(s)	23.3	6.2	14.7	4.1
Inference Time(s)	3.51	0.9	1.01	0.5
Total Time(s)	27.81	7.40	16.51	5.30

VI. CONCLUSION

As this research demonstrates, network anomalies can be detected with high accuracy without requiring large labelled datasets. The models prove to accurately detect common network anomalies such as DDoS attacks and internet sweeps being performed by network devices.Other 'Novelties' in network traffic were also flagged which demonstrates the ability to detect Zero Day attacks on or using consumer devices. Semi supervised models such as One Class SVM are clearly the forerunners for practical purposes due to their high detection rates and the fact that they do not need extensive labeled datasets which are unavailable for consumer networks. A deep learning approach using Deep Auto Encoder Networks [30] might also prove effective as semi-supervised ML models have shown promising results in this research. Handling more categorical variables and unbalanced classes in Semi supervised learning models offers a promising direction for future research especially for securing the next generation of IoT devices.

REFERENCES

[1] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning ddos detection for consumer internet of things devices," *CoRR*, vol. abs/1804.04159, 2018.

[2] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, 2017.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, 2009.

[4] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, 2007.

[5] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers and Security*, 2009.

[6] C.-H. Lo and N. Ansari, "Consumer: A novel hybrid intrusion detection system for distribution networks in smart grid," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, p. 33–44, Jun 2013.

[7] Z. Zhang, J. Li, C. N. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification," in *Proceedings of IEEE Workshop on Information Assurance and Security*, 2001.

[8] C. Manikopoulos and S. Papavassiliou, "Network intrusion and fault detection: A statistical anomaly approach," *IEEE Communications Magazine*, 2002.

[9] P. K. Chan, M. V. Mahoney, and M. Arshad, "A machine learning approach to anomaly detection," *Department of Computer Sciences*, 2003.

[10] M. Mahoney and P. Chan, "Learning rules for anomaly detection of hostile network traffic," 2004.

[11] K. Wang and S. J. Stolfo, "Anomalous Payload-Based Network Intrusion Detection," 2010.

[12] S. Xiuyao, W. Mingxi, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, 2007.

[13] P. Chhabra, C. Scott, E. D. Kolaczyk, and M. Crovella, "Distributed spatial anomaly detection," in *Proceedings - IEEE INFOCOM*, 2008.

[14] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *Eurasip Journal on Advances in Signal Processing*, 2009.

[15] F. Simmross-Wattenberg, J. I. Asensio-Pérez, P. Casaseca-De-La-Higuera, M. Martín-Fernandez, I. A. Dimitriadis, and C. Alberola-López, "Anomaly detection in network traffic based on statistical inference and α-stable modeling," *IEEE Transactions on Dependable and Secure Computing*, 2011.

[16] M. Y. , "A Nonparametric Adaptive Cusum Method And Its Application In Network Anomaly Detection," *International Journal of Advancements in Computing Technology*, 2012.

[17] G. Creech and J. Hu, "Generation of a new IDS test dataset: Time to retire the KDD collection," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2013.

[18] "wireshark," https://www.wireshark.org/.

[19] "Netcap," https://github.com/dreadl0ck/netcap.

[20] "matplotlib," https://matplotlib.org.

[21] "pandas," https://pandas.pydata.org.

[22] "scikit-learn," https://scikit-learn.org.

[23] A. Mukerji and J. Rothstein, "Detecting anomalous network application be havor," 2012.

[24] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, p. 708–713, 2015.

[25] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys and Tutorials*, 2014.

[26] A. S. S. Navaz, V. Sangeetha, and C. Prabhadevi, "Entropy based anomaly detection system to prevent ddos attacks in cloud," *CoRR*, vol. abs/1308.6745, 2013.

[27] N. S. Arunraj, R. Hable, M. Fernandes, K. Leidl, and M. Heigl, "Comparison of Supervised , Semi-supervised and Unsupervised Learning Methods in Network Intrusion Detection System (NIDS) Application," *Anwendungen Und Konzepte Der Wirtschaftsinformatik (AKWI)*, 2018.

[28] M. A. Rassam, M. A. Maarof, and A. Zainal, "Adaptive and online data anomaly detection for wireless sensor systems," *Knowledge-Based Systems*, vol. 60, p. 44–57, Apr 2014.

[29] X. He, G. Mourot, D. Maquin, J. Ragot, P. Beauseroy, A. Smolarz, and E. Grall-Maës, "Multi-task learning with one-class SVM," *Neurocomputing*, 2014.

[30] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, "Detection of anomalies in large scale accounting data using deep autoencoder networks," *CoRR*, vol. abs/1709.05254, 2017.