Arguments Against Using the 1998 DARPA Dataset for Cloud IDS Design and Evaluation and Some Alternative

Paulo Faria Quinan, Issa Traore, Isaac Woungang, Abdulaziz Aldribi, Onyekachi Nwamuo

Arguments Against using the 1998 DARPA Dataset for Cloud IDS Design and Evaluation and Some Alternative

Onyekachi Nwamuo[1], Paulo Magella de Faria Quinan[1], Issa Traore[1], Isaac Woungang[2], Abdulaziz Aldribi[3]

[1]University of Victoria, ECE Department, Victoria, BC Canada

onyekachien@uvic.ca,quinan@uvic.ca,itraore@ece.uvic.ca

[2]Department of Computer Science, Ryerson University, Toronto, Ontario, Canada

iwoungan@ryerson.ca

[3]Qassim University, Buraydah Saudi Arabia

Abstract. Due to the lack of adequate public datasets, the proponents of many existing cloud intrusion detection systems (IDS) have relied on the DARPA dataset to design and evaluate their models. In the current paper, we show empirically that the DARPA dataset by failing to meet important statistical characteristics of real world cloud traffic data center is inadequate for evaluating cloud IDS. We present, as alternative, a new public dataset collected through a cooperation between our lab and a non-profit cloud service provider, which contains benign data and a wide variety of attack data. We present a new hypervisor-based cloud IDS using instance-oriented feature model and supervised machine learning techniques. We investigate 3 different classifiers: Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) algorithms. Experimental evaluation on a diversified dataset yields a detection rate of 92.08% and a false positive rate of 1.49% for random forest, the best performing of the three classifiers.

Keywords: cloud IDS, cloud security, machine learning, IDS evaluation, Hypervisor-based IDS.

1 Introduction

In today's IT and business world, there has been a significant increase in the public adoption of cloud computing for the production systems and services support, and there seems to be no end in sight [33]. However, the growth in the public adoption of the cloud paradigm has increased organizations exposure to a wide variety of cyber attacks and vulnerabilities. Intrusion detection system (IDS) is one of the key tools being used or explored in combatting cloud attacks.

Until now, the availability of a cloud dataset has been one of the major challenges hampering the progress of the research on cloud IDS. The majority of the works done so far on cloud IDS was done using conventional datasets like the DARPA 1998 or the

KDD'99 datasets [1,3,5,10]. More so, the datasets used in the works done on a cloud environment are not made available for public use, in some cases for privacy concerns. These factors have denied the cloud researchers of an all-encompassing real-world cloud intrusion dataset to carry out their work on. As shown in previous studies [4,7], there are strong differences between cloud network data and conventional network data in terms of their characteristics such as flow inter-arrival time, packet-level communication, load ratios of the internal / external traffic flow, and so forth. On the other hand, the design of anomaly detection models involves constructing normal activity baselines from previously collected sample activity data. Hence, constructing cloud anomaly detection models using conventional network data would fail to capture adequately cloud network behavior considering the aforementioned differences between cloud network and conventional network data.

The objective of the current work is to provide an empirical justification for the need for a dataset collected specifically in a real cloud environment compared with using a conventional network dataset in developing cloud IDS. Furthermore, we explore the design of cloud anomaly detection using supervised machine learning techniques. Specifically, three machine learning algorithms are studied: logistic regression (LR), random forest (RF) and support vector machine (SVM).

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 highlights informally the deficiencies of the DARPA dataset and introduces as alternative a real Cloud IDS dataset. Section 4 provides empirical evidence supporting the claim that the DARPA dataset does not meet key characteristics of cloud data. Section 5 presents a new hypervisor-based cloud IDS model using supervised machine learning. Finally, Section 6 makes concluding remarks.

2 Related Works

To address the lack of public cloud-specific datasets, some researchers have focused on generating new datasets, such as [9,13,14]; however, to the best of our knowledge none of these datasets are openly available. As a result, many existing cloud IDS proposals have relied on conventional IDS datasets for development and evaluation, using primarily the DARPA IDS dataset or the KDD CUP dataset. We discuss some of these proposals in the following.

Bhat et al. [3] proposed an approach for detecting intrusions in virtual machine environment on cloud using traditional and multiclass (hybrid) machine learning algorithms. The following machine learning algorithms were considered: Naïve Bayes Tree (NB Tree) classifier, hybrid of NB Tree and Random Forest. The NSL-KDD'99 dataset was used for evaluation and it was observed that hybrid machine learning models perform better than the traditional or individual algorithms. In using single classifiers, their evaluation generated accuracies of 95%, 91% and 98% for each of Random Forest, K-NN and SVM, respectively, while the combination of NB Tree and hybrid of NB Tree and Random Forest resulted in a high accuracy of 99% and low false

positive rate of 2%. More so, the hybrid of Random Forest and weighted K-Means amounted to 94.7% accuracy and 12% false positive rates.

Modi and Patel [12] presented an approach that integrates hybrid Network Intrusion Detection Systems to cloud computing environment. The experimental set up involves using the Eucalyptus infrastructure for the simulation of a cloud computing environment, while the KDD IDS dataset was used for the evaluation of their work. Their research framework involved the integration of signature-based detection and anomaly detection. They utilized Snort, for the signature-based intrusion detection and three machine learning classifiers viz the Bayesian, Associative (a machine learning model using association rule) and Decision Tree classifiers singularly and collectively for the network anomaly detection. The experimental result of their proposal for the three classifiers and their collective ability yielded a true positive rate (TPR) of 97.14%, and a false positive rate (FPR) of 1.17%.

Muthurajkumar et al. [15], used the combination of fuzzy SVM and random feature selection algorithms (RSFSA) to propose a cloud intrusion detection model. In their experiment, they built two sets of intrusion detection model, one with the whole data features and the other after introducing feature selection. A dataset consisting of 10% of KDDCUP was used for the experiment and analysis of their approach. The average detection rate from their experimental results before and after applying the RSFSA to the Fuzzy SVM classifier are 86.88% and 94.15%, respectively. Their work confirmed that feature selection plays an important role in the classifiers' detection accuracy. It would have ben interesting to evaluate this proposal using a real cloud intrusion detection dataset.

Chou et al. [5] proposed an adaptive network-based intrusion detection system for the cloud environment using the DARPA 2000 and the KDD Cup 1999 datasets. Their approach used spectral clustering, an unsupervised learning algorithm to build a decision tree-based detection model for detecting an anomaly in an unlabeled network connection data. They used Bro-IDS to generate connections records from the raw packets. Their experimental result on the DARPA dataset yielded a detection rate of 95% and a false positive rate of 4.5% while the KDD Cup 1999 dataset yielded a detection rate of 90% and a false positive rate of 5%. Their approach is not enough robust as it could not detect DOS and some probing attacks which create a great amount of connections.

Ahmad et al. [1] presented an intrusion detection model that uses Dendritic Cell Algorithm for detecting intrusions in cloud computing environment. The experimental evaluation was conducted using the DARPA 1999 dataset. The network-based attributes were used as signals in their experiments. They carried out their experiment on a total of 187 threat events of Week 4 and Week 5 of the DARPA 1999 dataset and the algorithm achieved a detection rate of 79.43% and a false positive rate of 13.43%. In their work, they demonstrated that using Dendritic Cell algorithm could provide a solution in detecting attacks in the cloud environment.

Kannan et al. [8] proposed a host-based cloud intrusion detection system which uses a genetic algorithm based feature selection and a Fuzzy SVM based classifier for deciding if an event is intrusion or not. The cloud environment was simulated with Proxmox VE 1.8 which is an open source virtualization environment while the evaluation was done using the KDD'99 cup dataset. In the experimental results, a detection rate of 98.51% and a false positive rate of 3.13% were obtained.

Zhao et al. [17] put forward an anomaly detection system based on an unsupervised learning algorithm, namely the K-means clustering algorithm. The dataset chosen for their experiment was the KDD Cup 99 dataset. For the comparative analysis of the performance of their proposed approach, they used the Particle Swarm Optimization (PSO) and Backpropagation (BP) Neural Network algorithms to test the performance of their proposed algorithm. The K-means algorithm performed better than the other two algorithms, yielding a false positive rate (FPR) of 3.56% as against FPR of 6.78% and 5.75% for PSO and BP neural network algorithms, respectively. And in terms of false negative rate (FNR), 7.65% was achieved in contrast to 10.46% and 13.75% obtained using PSO and BP neural network algorithm, respectively. Their study was not carried out with a real cloud IDS dataset but it however worth its salt as it highlighted the possibility of predicting several types of attacks in the cloud.

Xiong et al. [16] proposed an anomaly detection method for cloud computing systems based on two approaches, viz the Synergetic Neural Network (SNN) algorithm and the Catastrophe theory (CT) algorithm. They used the DARPA dataset for their experiment and focused their work on the network traffic information. Their experiment yielded an overall average detection rate of 83% on the SNN algorithm and 86.62% overall average detection rate on the CT algorithm. The experiment also yielded an overall average of 8.3% false positive rate on the SNN algorithm and an overall false positive rate of 9.06% on the CT algorithm.

Li et al. [11] proposed an artificial neural network (ANN) based cloud IDS. The experimental part of their proposal involved simulating a cloud environment using Ubuntu Enterprise Cloud (UEC), a Eucalyptus-powered cloud platform and evaluating the result on 10% of the KDD'99 dataset. The experiment yielded an average detection rate of 99% and an average detection time of 37.1 seconds. One of the drawbacks here is that the ANN takes huge training time for large databases, therefore, the anomaly detection algorithm may incur an increased cost if retraining is required due to change in traffic behaviour as in the case of the cloud computing environment. More so, the simulated dataset can not stand in as a real cloud dataset.

3 Datasets for Cloud IDS Evaluation

In this section, we compare the characteristics of conventional network data with cloud network data, and give an overview of the ISOT cloud IDS evaluation dataset.

3.1 Cloud Network vs. Conventional Network Data Characteristics

Great disparity exists when considering the proximity of both the cloud and conventional network data centers. While the cloud data centers are distributed globally, the non-cloud data centers are always situated in a close proximity to their users or on the premises of the serving organizations. The global placement of the cloud datacenters satisfies the requirements for geo-diversity, geo-redundancy and regulatory constraints [4]. Studies [4,7] have shown that the characteristics of cloud network traffic are different from the conventional network traffic in so many ways as explained in the following.

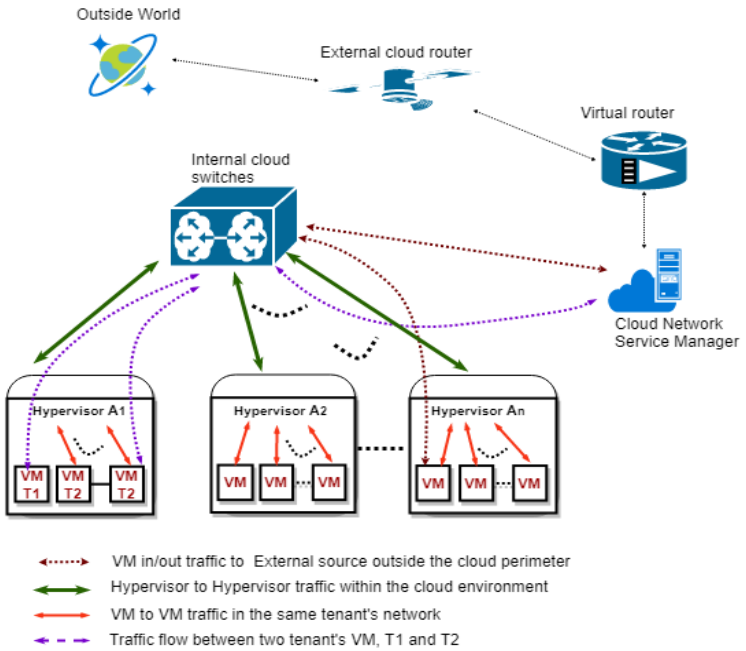

Figure 1: Network Flow in ISOT-CID

Network Flow: Empirical studies in [4], [7] have shown that the inter-arrival time for 80% of the traffics in the cloud network is usually under 1ms while with the conventional network, this can be between 4ms and 40ms as the traffic does not change that quickly. In their work, it was also noted that the number of active flows for any given second at a switch is at most 10,000 flows and that new flows can also be highly instantaneous in arrival. These studies also went on to explain how the flow inter-arrival time affects the kind of processing that can be done for each new flow and the usefulness of logically centralized controllers for the flow placement. The cloud network traffic is usually bursty in nature with the ON/OFF intervals being characterized by heavy-tailed distributions. Their analysis also shows that in a cloud

computing environment, the load ratios of the internal/external traffic flow between the instance to instance or instance and other sources are usually high. In [7], it was also discovered that in a conventional network data, 80% of flows are usually smaller than 10Kb in size as compared with a cloud network data. On the one hand, the flow of communication patterns in a cloud network is usually high due to the numerous applications being hosted and high link utilization across the cloud's multiple layers. On the other hand, with the traditional network, the communication flow pattern and the link utilization are usually small in size. Figures 1 and 2 show the network flows for a typical cloud environment based on the ISOT-CID and the DARPA 1999 IDS evaluation dataset which was collected by simulating a conventional network environment [2]. In the ISOT-CID environment shown in figure 1, we can see significant variability in the network flows including the hypervisor to hypervisor network flows, the in/out traffic flows from VM to external source not within the cloud environment, the VM to VM network flows, and traffic flow between tenants VMs [2]. The conventional network comprises of limited network flows as can be seen in figure 1.4 which has only two network flows viz external and internal traffic flows [2].

Topology: The physical topology of a cloud data center follows a canonical 3-Tier architecture which consists of the core layer or the uppermost layer, aggregation layer or the middle layer and the edge layer or the lower link layer. In contrast, the traditional data centers follow a 2-Tiered topology in which the core layer and the aggregation layers are collapsed to form one layer [4]. In a typical cloud network, data is either centralized or outsourced and provided to the users on-demand irrespective of their geographic location. This relieves the data owner of the full control of their data as the cloud service providers now manage and maintain the data. The cloud data also has the flexibility of being scaled up or down by automated means. Some giant cloud service providers such as Amazon, Google and Microsoft do have cloud data centres dispersed geographically for the provision of universal data access to the various users [4].

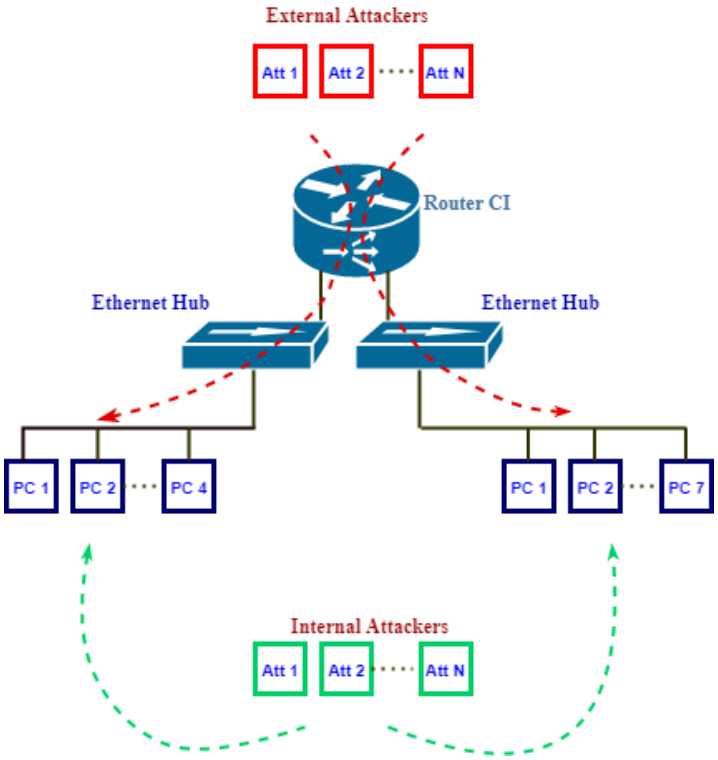

Figure 2: 1999 DARPA IDS Evaluation Dataset Network Flow

3.2 Overview of the ISOT Cloud IDS Dataset

The ISOT-CID is a publicly available dataset that was collected in a real world environment using the infrastructure of Compute Canada, a nonprofit cloud service provider that extends its services in the areas of providing the computational needs of researchers [2]. There were two phases involved in the ISOT-CID data collection procedure, namely Phase 1 in 2016 and Phase 2 in 2018. The data in the two phases were collected on the same production environment based on OpenStack from various cloud layers such as hypervisors, guest hosts layers and the network layer. The dataset size is more than 8 terabytes and it contains data of different formats such as the memory dumps, CPU and disk utilizations, system call traces, system logs and network traffic [6]. Another advantage of the ISOT-CID is that it is labelled and includes both normal and attack activities. The current work is based on the network traffic attributes of the ISOT-CID.

The ISOT-CID collection environment contains three hypervisor nodes viz, node A, node B, and node C. The collecting environment is also composed of 10 virtual machines or instances (VM1 to VM10) launched in three different cloud zones A, B, and C [2].

The benign data in the ISOT-CID came from web applications/traffic and administrative activities [6]. Some of the web application activities include account registration, blog activities, and web browsing. The web traffic statistics revealed that more than 160 legitimate users were involved in the generation of the normal data which comprises of 60 human users and 100 robots [2]. While the administrative activities cut across instance routine maintenance, system rebooting, application updates, file creation, machine access via SSH and remote server access.

Table 1 presents all the attacks covered in ISOT-CID, such as probing, DoS, information disclosure, R2L, input validation, backdoors and authentication breach, etc. These attacks were grouped into insider or outsider attacks depending on its source [2]. On the one hand, the inside malicious activities were perpetrated by either an insider within the cloud environment who had a root access on the hypervisor nodes or by a compromised VM within the cloud environment used as a stepping stone. Some of the inside attacks were backdoor and Trojan horse, network scanning, password cracking, DoS attacks, and so forth. On the other hand, the outside malicious activities emanated from outside the cloud environment with the ISOT-cloud environment being the primary target. Some of the outside attacks are made up of the application layer (layer 7) and network layer (layer 3) DoS attacks, input validation attacks, SQL injection, path/directory traversal and cryptojacking (unauthorized cryptomining). For instance, Figure 3 shows a timeline for the attack scenario in Phase 1 Day 2 (2016-12-15).

Table 1: Attacks covered in the ISOT-CID dataset [2]

Attack Target Layer	Insider Attack Types	Outsider Attack Types
Application Layer		➢ SQL Injection ➢ Web Vulnerabilities Scanning ➢ Cross-site Scripting (XSS) ➢ Dictionary/Brute Force login attack ➢ Fuzzers ➢ HTTP Flood DOS ➢ Directory/Path Traversal
Network Layer	➢ Trojan Horse ➢ Backdoor (reverse shell)	➢ Synflood Dos ➢ Unclassified (unsolicited traffic)

	➢ Unauthorized Cryptomining (download/install/run cryptominer) ➢ UDP Flood DOS ➢ Stepping Stone Attack ➢ Ports and Network scanning ➢ Synflood DOS ➢ Revealing Users Credentials and Confidential Data by Insider ➢ Dictionary/Brute Force login attack	➢ DNS amplification DOS ➢ Ports and Network scanning ➢ Dictionary/Brute Force login attack

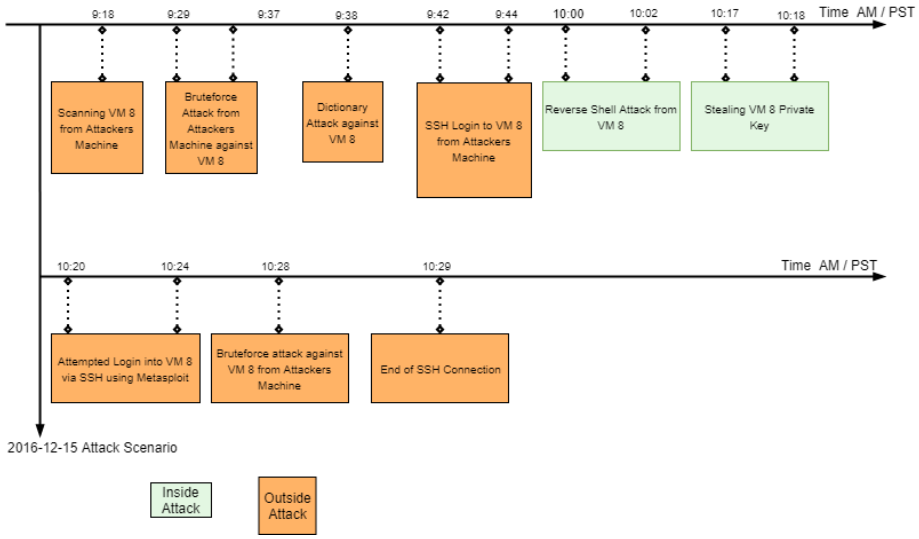

Figure 3: Timeline for Phase 1 Day 2 (2016-12-15) Inside and Outside Attacks

Composition of ISOT-CID Network Traffic Data

The ISOT-CID is composed of three levels of network communications namely: external, internal and local traffic [6]. In the ISOT-CID context, the external traffic is the traffic between the instance and an outside machine. The internal traffic or the hypervisor traffic is between the hypervisor nodes. And finally, the local traffic is the traffic between two VMs on the same hypervisor node. The ISOT-CID network data also comes in kinds, one being without payload on both hypervisors and VMs and the other involving the full network traffic only on the hypervisors [2]. The ISOT-CID network traffic/packet statistics are shown in Table 2.

Table 2: ISOT-CID Network Traffic Distribution [2]

Phase	Total Normal Traffic	Total Malicious Traffic	Total Packets
1	22,356,769	15,649	22,372,418
2	9502872	2,006,382	11,509,254

4 Traffic Characterization

We analyzed the network traffic data of the ISOT-CID and the DARPA 1998 datasets by looking at the network communication patterns at the flow-level. The idea is to show how similar or different they are in terms of data transmission behaviour at the flow level. This is also in line with the work done in [4] where the authors used traffic engineering techniques to distinguish between cloud data center networks and conventional or traditional networks. We considered three metrics in our empirical data analysis, viz Number of Flows, Flow Inter-arrival Times, and Flow of Traffic characteristics of the two datasets.

4.1 Number of Active Flows Characteristics

Figure 4 represents the empirical cumulative distribution function (CDF) of the number of active flows at different switches within 120 seconds time window for both ISOT-CID Phase 2 Day 1 (2018-02-16) dataset and DARPA 1998 Tuesday week 4 Training dataset. Our findings based on the distribution reveals that, the number of active flows for the ISOT-CID is between 2,000 to 6000 about 90% of the time. In the case of the DARPA 1998 dataset, the number of active flows is between 20 and 1000 in 90% of the time interval. This empirical observation supports the results of a prior work on data center traffic [4]. It is also considerable to note that the latency assigned by a controller to a new flow is determinant on the lengths of the flows [4].

4.2 Flow Inter-arrival Time Characteristics

Additionally, we examined the empirical CDF of the flow inter-arrival times under a 120 seconds time window on ISOT-CID Phase 2 Day 1 (2018-02-16) dataset and

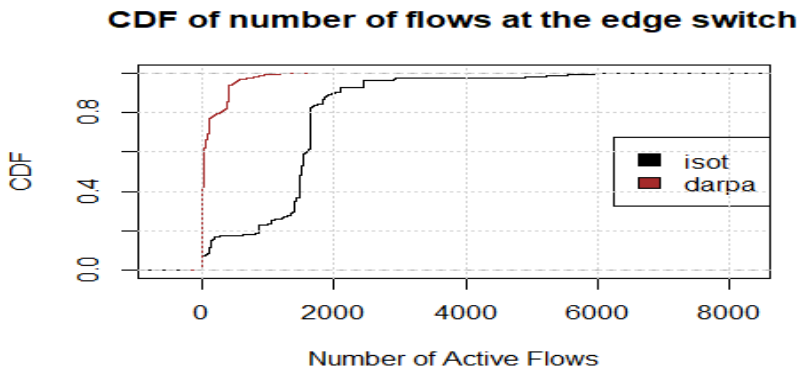

Figure 4: The CDF of the distribution of the number of flows at the edge switch in ISOT and DARPA

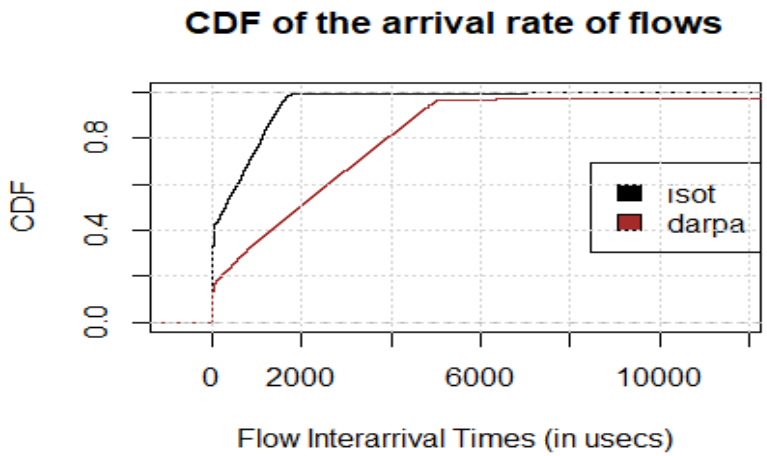

Figure 5: The CDF of the distribution of the flow inter-arrival time in ISOT and DARPA

DARPA 1998 Tuesday week 4 Training dataset as represented in figure 5. We discovered that the flow inter-arrival time for 80% of the new flows arriving at the monitored switch is 1*ms* for the ISOT-CID dataset and 4*ms* for the DARPA dataset. These results suggest that DARPA is characterized by a smaller number of flows than the ISOT_CID dataset. This empirical observation also supports the results of a prior work [4].

The flow inter-arrival times affects the scalability of the controller because a significant number of new flows arrive at a given switch within an interval of few microseconds [4]. Therefore, it is recommended to use multiple CPU's per controller and multiple controllers to compute the routes in order to scale the throughput of a centralized control framework.

4.3 Flow-Level Communication Characteristics

The aggregate network transmission behaviour of the two datasets were examined using a day's traffic from each respectively. We based the analysis on two network flow metrics, viz the *extra-rack* traffic and the *intra-rack* traffic. The *extra-rack* traffic signifies the traffic leaving the switch rack or internal hosts for other internal hosts or external destinations, this is easily measured while the *intra-rack* traffic represents the amount of traffic that stays within the rack or node [4]. On the one hand, the *intra-rack* traffic for ISOT-CID, was computed by taking the difference between the volume of traffic generated by the instances attached to the hypervisor nodes and the traffic exiting the nodes.

Table 3: Extra-Rack and Intra-Rack traffic composition for ISOT-CID and DARPA 1998, showing the number of packets.

Flow Metric	ISOT-CID phase 2 Day 1 (2018-02-16) dataset	DARPA Tuesday week 4 Training dataset
Extra-Rack Traffic	693426	932843
Intra-Rack Traffic	2177148	864810
Total Traffic	2870574	1797653

Table 4: Percentage composition of Extra-Rack and Intra-Rack traffic for ISOT-CID and DARPA 1998

Flow Metric (%)	ISOT-CID phase 2 day 16 dataset	DARPA Tuesday week 4 Training dataset
Extra-Rack Traffic (%)	24.16	51.89
Intra-Rack Traffic (%)	75.84	48.11
Total Traffic (%)	100	100

On the other hand, the *intra-rack* traffic for DARPA 1998 dataset was computed by taking the difference between the volume of traffic generated by the servers or host attached to the switches and the traffic exiting the switches. Table 3 shows the *extra-rack* and *intra-rack* traffic compositions, while Table 4 provides the percentage representation of these two metrics. Figure 6 depicts a bar graph showing the ratio of

extra-rack to *intra-rack* traffic in the selected day traffic of both the ISOT-CID and the DARPA 1998 datasets.

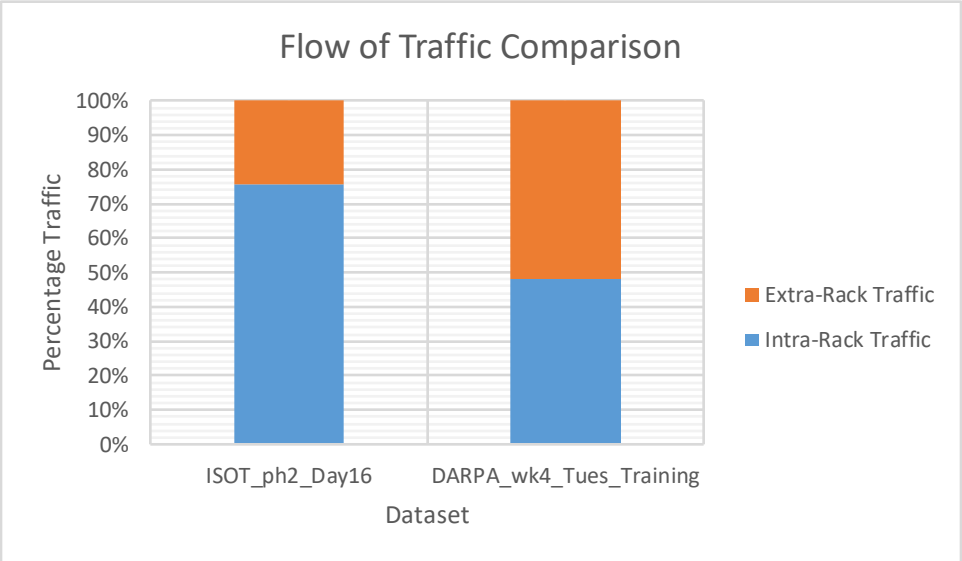

Figure 6: Comparison of the ratio of extra-rack to intra-rack traffic for ISOT-CID and DARPA 1998 datasets

The result of the analysis shows that for the ISOT-CID dataset, 75.84% of the traffic is confined to within the hypervisor node in which it was generated while 24.16% of the traffic leaves the nodes. This result is in contrast with the DARPA 1998 dataset in which only 48.11% of the traffic stays within the communication nodes and 51.89% of the traffic leaves the nodes. The result of this network traffic analysis supports the observations made in prior studies [4,7] of network traffic characterization of data centers.

5 Hypervisor-based Cloud IDS using Supervised Machine Learning

In this section, we explore the effectiveness of hypervisor-based cloud anomaly intrusion detection using supervised machine learning. Specifically, three machine learning algorithms are studied: logistic regression (LR), random forest (RF) and support vector machine (SVM).

5.1 Feature Model

Because the VM instances in the cloud environment share the same hypervisor, to improve the cloud computing intrusion detection process, the feature extraction should be such that it takes into account the correlated behavior of the instances. A network

flow on the other hand can be seen as a bidirectional packet streams between two hosts or the movement of network traffic across different network points, usually from a source to a destination and vice versa. We first grouped the captured hypervisor packets in the pcap file formats into a stream of packet flows based on a time window δt using a flow based forensic and network troubleshooting traffic analyzing tool called Tranalyzer. Eighty raw features were extracted from the packet headers and some of the raw features are represented in table 5.

Table 5: Some of the raw features extracted from the hypervisor network traffic using the traffic analyzer tool (Tranalyzer)

Feature	Description
flowInd	The flow index
timeFirst	Date/time of first packet
timeLast	Date/time of last packet
duration	Flow duration
srcIP	Source IP
srcPort	Source port
dstIP	Destination IP
dstPort	Destination port
numPksSnt	Number of transmitted packets
numPksRcvd	Number of received packets
numBytesSnt	Number of transmitted bytes
numBytesRcvd	Number of received bytes

We used a three-dimensional features space in this thesis work namely frequency-based features, entropy-based features and load-based features based on the work of Aldribi et al. [2].

Two categories of frequency-based features were adopted for each VM instance, namely, the ‘in-frequency’ and the ‘out-frequency’ features. The in-frequency represents the frequency of the packets incoming to a specific instance from any source or endpoint while the out-frequency represents the outgoing packets from a specific instance back to the respective sources.

The load-based features were extracted by taking the ratio of the matching *in* an *out* frequency features as proposed in [2].

The entropy associated with the probability distribution of network traffic occurrences at ingress and egress points during the observation time window was computed for the specific instances. Given a distribution of probabilities $P = \{p_1, p_2, \ldots\ldots, p_N\}$ having N variables, entropy is defined as

$$H_s = -\sum_{i=1}^{N} p_i \, log_2 p_i \qquad (1)$$

Where $0 \leq p_i \leq 1$ and $\sum_{i=1}^{N} p_i = 1$. And in our case, p_i represents the probability of the distinct frequency features of the traffic during the observation time window. For instance, the entropy of the source IP is calculated by first computing the appearing probability associated with the source IP which is gotten by taking the ratio of the number of packets with the specified source IP address and the total number of packets observed in the flow, after that the entropy equation of (1) is adopted to get the value. The entropy is minimum ($H_s = 0$) at maximum flow concentration or when the features exhibit a deterministic behaviour. On the other hand, the entropy H_s is maximum ($H_s = log_2 N$) at maximum flow dispersion or when the feature is fully at random.

5.2 Data Preparation

The data preparation steps undertaken in this research work in order to get the best features in the dataset are shown in fig 7.

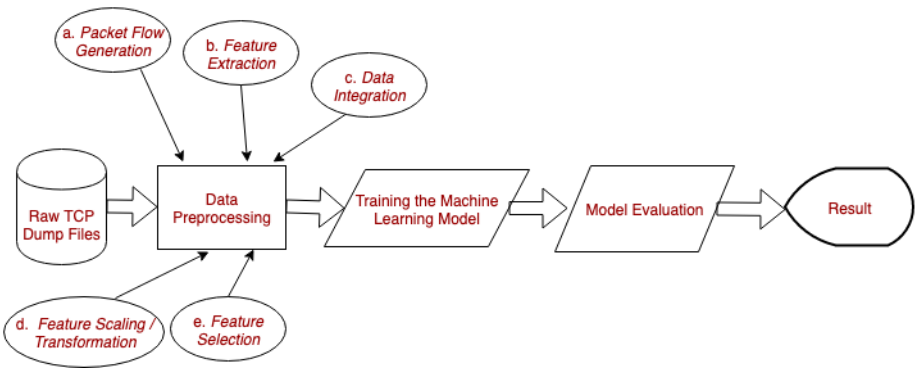

Figure 7: Data Preparation Subsystem

We leveraged the computational power of pandas, an open source software library for *Python* programming language to developed a python script to extract the aforementioned three feature categories of our feature model. The extracted features were transformed to the same scale between 0 and 1 using the *min-max* normalization approach. To reduce the dimensionality and complexity of the feature space, we used the *CARET* R-library, a tree-based feature selection technique. The *CARET* R-package gives a percentage score to all the features with the noisy features having a percentage score of zero (0) in accordance to their statistical significance, that is information gain.

The most important features are then used to train the machine learning model. Table 6 show the features and their overall significance to the model prediction as indicated by the *CARET R*-package.

Table 6: Features and their overall importance

Feature	Definition	Overall Significance (%)
$f_i^{in}(t)$	The total number of packets flowing to e_i during $[t , t + \delta t]$ divided by δt.	100
$L_i(t)$	Load feature matching the ratio of the total number of packets flowing to and from endpoint e_i during $[t , t + \delta t]$.	97.382
$f_{i,iP,d,dp}^{out}(t)$	The number of packets flowing from the endpoint e_i to e_d during $[t , t + \delta t]$ divided by δt.	87.251
$f_i^{out}(t)$	The total number of packets flowing from e_i during $[t , t + \delta t]$ divided by δt.	86.431
$f_{s,sp,i,iP}^{in}(t)$	The number of packets flowing from the endpoint e_s to e_i during $[t , t + \delta t]$ divided by δt.	85.590
$max_{ip}\{f_{i,ip}^{out}(t)\}$	The maximum number of packets over *ip* flowing out of e_i during $[t , t + \delta t]$ divided by δt.	8.298
entropy_dp	Entropy of the destination port	8.023
$f_{i,ip}^{out}(t)$	The number of packets flowing from specific *ip* in e_i to all *dp* in all endpoints e_d during $[t , t + \delta t]$ divided by δt.	7.877
numPksRcvd	Number of received packets	7.877
numPktsSnt	Number of transmitted packets	6.992
$L_{max\ (ip)}(t)$	Load feature matching the ratio of the maximum number of packets over the instant port (ip) flowing to and from endpoint e_i during $[t , t + \delta t]$.	3.098
$f_{i,ip}^{in}(t)$	The number of packets flowing from all *sp* in all endpoints e_s to specific *ip* in e_i during $[t , t + \delta t]$ divided by δt.	2.915
$L_{i,ip}(t)$	Load feature matching the ratio of the number of packets flowing from all source ports in all endpoints e_s to and from specific instance port (ip) in e_i during $[t , t + \delta t]$.	2.750
$f_{sp,i}^{in}(t)$	The number of packets flowing from specific *sp* in all endpoints e_s to all *ip* e_i during $[t , t + \delta t]$ divided by δt.	0.00
$L_{s,sp,i,ip}(t)$	Load feature matching the ratio of numbers of packets flowing to and from the endpoint e_s to e_i during $[t , t + \delta t]$.	0.00
$L_{sp,i}(t)$	Load feature matching the numbers of packets flowing from specific source port in all endpoints e_s to and from all instance ports in endpoint e_i during $[t , t + \delta t]$.	0.00

entropy_srcPort	Entropy of the source port	0.00
entropy_srcIP	Entropy of the source IP	0.00
$max_{ip}\{f_{i,ip}^{in}(t)\}$	The maximum number of packets over *ip* flowing to e_i during [*t* , *t* + δt] divided by δt.	0.00
$f_i^{out}(t)$	The total number of packets flowing from e_i during [*t* , *t* + δt] divided by δt.	0.00

5.3 Model Evaluation using ISOT-CID

In this research work, the performance of the three machine learning algorithms was measured using the detection rate (DR) (also known as true positive rate (TPR)) and the false positive rate (FPR) which are two metrics commonly used for IDS performance computation.

The observation time window δt was set at 120 seconds for the network flow aggregation and feature extraction. The data was processed following the steps described in figure 7. The DR and the FPR obtained from the three machine learning algorithms for each VM instances over different attack days and their respective overall results were computed. The overall performance of the machine learning classification algorithms is summarised in table 7 for ISOT-CID (covering both phases 1 and 2).

Table 7: Comparison of overall performance for ISOT-CID

Algorithm	***Overall***	
	FPR (%)	*DR (%)*
Logistic Regression	*2.61*	*90.52*
Random Forest	*1.49*	*92.08*
SVM	*1.84*	*92.06*

The random forest algorithm was the best of the three-machine learning algorithms in terms of performance with a detection rate of 92.08% and a false positive rate of 1.49%.

Table 8: Confusion matrix for random classifier based on ISOT-CID

		Prediction	
		Attack	Normal
Reference	Attack	TP=114291	FN=9831
	Normal	FP=4428	TN=292775

Table 8 shows the confusion matrix for the random forest classifier. The matrix shows the number of flows being classified; the cells contain the numbers of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

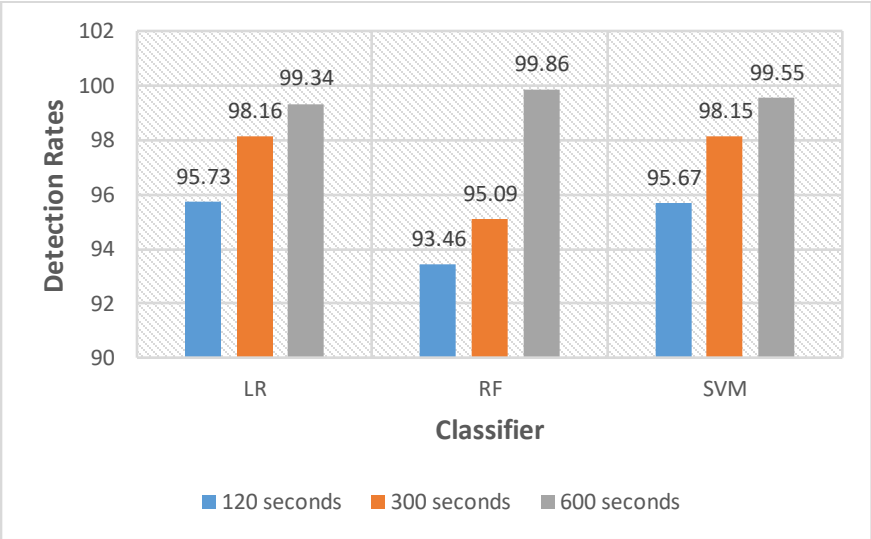

Figure 8: Effect of observation time window on the detection rate

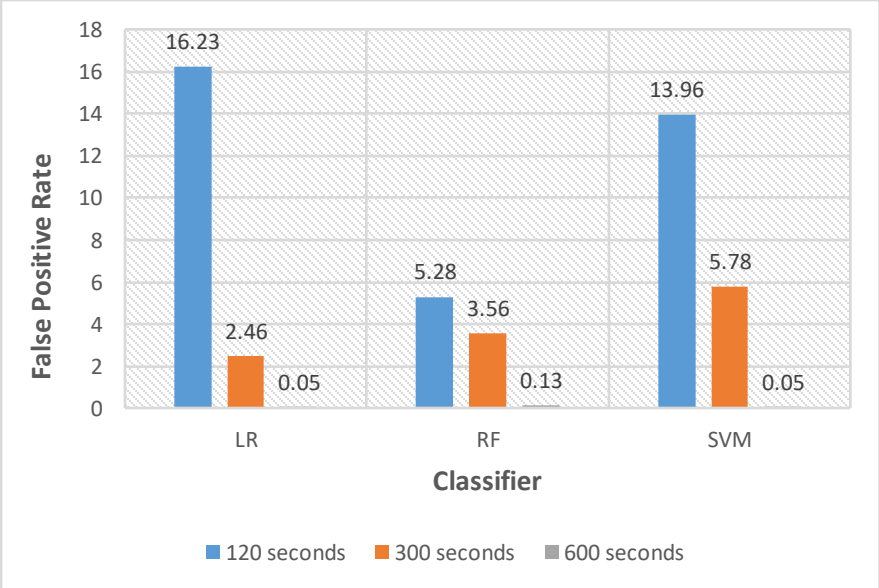

Figure 9: Effect of observation time window on the false positive rate

We examined also the effect of varying the size of the observation time window δt on the DR and FPR of the three machine learning models. Day 5 of phase 2 hypervisor B data was used for this analysis. Figures 8 and 9 show the effect on the detection rate and the false positive rate, respectively. On the one hand, as the observation time window increases, the amount of data available for decision increase thereby presenting the model with a more balanced dataset which will in turn aid in better decision making. On the other hand, as the time window increases, the decision time is delayed which represents an increased window of vulnerability.

6 Conclusion

Security and privacy remain one of the main issues faced by cloud computing adopters and consequently, there is an urgent need for the IT professionals and subject matter experts to come up with a system that can both detect and protect the cloud infrastructure from malicious activities. In this paper, we carried out an empirical analysis on the DARPA intrusion evaluation dataset and showed its deficiencies when compared to the ISOT-CID which is a real cloud computing dataset. The results support previous work done on network traffic characterization of data centres. It is the claim of this work that due the deficiencies, the DARPA dataset should not be used as a genuine dataset in the design and evaluation of cloud IDS.

Also, we investigated cloud intrusion detection using different supervised machine learning models. The performance results obtained using the machine learning algorithms are encouraging meaning that if more effort and study is channeled into it, academia and researchers can come up with a better way to protect the cloud computing environment against intrusions.

In this paper, the empirical study for the characterization of network traffic to substantiate the difference between a cloud dataset and a conventional dataset was only limited to three flow-level metrics viz, the number of active flows, flow inter-arrival time, and flow -level communication patterns. Our future work will consist of extending the presented work by exploring other empirical means to further understand the nature of network traffic of the cloud and conventional dataset/datacenters like in the areas of packet-level communication, link utilizations and many more.

References

1. A. Ahmad and M. N. Kama, “CloudIDS: Cloud Intrusion Detection Model Inspired by Dendritic Cell Mechanism,” 2017.
2. A. Aldribi, I. Traore, and B. Moa, “Hypervisor-Based Cloud Intrusion Detection through Online Multivariate Statistical Change Tracking,” Computer & Security, Elsevier, 88 (2020).
3. Amjad Hussain Bhat, Sabyasachi Patra, and Dr. Debasish Jena, “Machine Learning Approach for Intrusion Detection on Cloud Virtua machines,” *IJAIEM*, vol. 2, no. 6, 2013.

4. Benson T., A. Akella, and D. A. Maltz, “Network traffic characteristics of data centers in the wild”, Proceeding IMC '10 Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, pp. 267-280
5. H.-H. Chou and S.-D. Wang, “An adaptive network intrusion detection approach for the cloud environment,” in 2015 International Carnahan Conference on Security Technology (ICCST), 2015, pp. 1–6.
6. Equinix “To Maximize the Cloud, Focus on the Network” Equinix whitepaper on Cloud. Available: https://equinix.app.box.com/embed/s/fakirceasoamu4ofxnt9rv6h4ujjytjm. [Accessed: 28-Jun-2019].
7. Kandula S., S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, “The Nature of Datacenter Traffic: Measurements and Analysis”. Proceeding IMC '09 Proceedings of the 9th ACM SIGCOMM conference on Internet measurement Pages 202-208, 2009.
8. A. Kannan, G. Q. Maguire, A. Sharma, and P. Schoo, “Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks,” Proc. - 12th IEEE Int. Conf. Data Min. Work. ICDMW 2012, pp. 416–423, 2012.
9. H. A. Kholidy, F. Baiardi, Cidd: A cloud intrusion detection dataset for cloud computing and masquerade attacks, in: Information Technology: New Generations (ITNG), 2012 Ninth International Conference on, IEEE, pp. 397-402.
10. H. K. K. Hyukmin Kwon, Taesu Kim, Song Jin Yu, “Self-similarity based lightweight intrusion detection method for cloud computingLecture Notes in Artificial Intelligence 6592 Edited by Subseries of Lecture Notes in Computer Science,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6592 LNAI, no. PART 2, pp. 353–362.
11. Z. Li, W. Sun, and L. Wang, “A neural network based distributed intrusion detection system on cloud platform,” Proc. - 2012 IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst. IEEE CCIS 2012, vol. 1, pp. 75–79, 2013.
12. C. N. Modi and D. Patel, “A novel hybrid-network intrusion detection system (H-NIDS) in cloud computing,” Proc. 2013 IEEE Symp. Comput. Intell. Cyber Secur. CICS 2013 - 2013 IEEE Symp. Ser. Comput. Intell. SSCI 2013, pp. 23–30, 2013.
13. M. Moorthy, M. Rajeswari, Virtual host based intrusion detection system for cloud, International Journal of Engineering and Technology (IJET) 5 (Dec 2013-Jan 2014) 5023-5029.
14. S. Mukkavilli, S. Shetty, L. Hong, Generation of labelled datasets to quantify the impact of security threats to cloud data centers, Journal of Information Security 7 (April 2016) 172-184.
15. S. Muthurajkumar, K. Kulothungan, M. Vijayalakshmi, N. Jaisankar, and A. Kannan, “A Rough Set based Feature Selection Algorithm for Effective Intrusion Detection in Cloud Model.”
16. W. Xiong et al., “Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications,” Inf. Sci. (Ny)., vol. 258, pp. 403–415, Feb. 2014.
17. X. Zhao and W. Zhang, “An anomaly intrusion detection method based on improved K-means of cloud computing,” Proc. - 2016 6th Int. Conf. Instrum. Meas. Comput. Commun. Control. IMCCC 2016, no. 61272172, pp. 284–288, 2016.