Clust-IT: Clustering-Based Intrusion Detection in IoT Environments

Robert P. Markiewicz
S3Lab, Information Security Group, Royal Holloway
robert.markiewicz.2013@live.rhul.ac.uk

Daniele Sgandurra
S3Lab, Information Security Group, Royal Holloway
daniele.sgandurra@rhul.ac.uk

ABSTRACT

Low-powered and resource-constrained devices are forming a greater part of our smart networks. For this reason, they have recently been the target of various cyber-attacks. However, these devices often cannot implement traditional intrusion detection systems (IDS), or they can not produce or store the audit trails needed for inspection. Therefore, it is often necessary to adapt existing IDS systems and malware detection approaches to cope with these constraints.

We explore the application of unsupervised learning techniques, specifically clustering, to develop a novel IDS for networks composed of low-powered devices. We describe our solution, called *Clust-IT* (Clustering of IoT), to manage heterogeneous data collected from cooperative and distributed networks of connected devices and searching these data for indicators of compromise while remaining protocol agnostic. We outline a novel application of OPTICS to various available IoT datasets, composed of both packet and flow captures, to demonstrate the capabilities of the proposed techniques and evaluate their feasibility in developing an IoT IDS.

ACM Reference Format:
Robert P. Markiewicz and Daniele Sgandurra. 2020. Clust-IT: Clustering-Based Intrusion Detection in IoT Environments. In *The 15th International Conference on Availability, Reliability and Security (ARES 2020), August 25–28, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3407023.3409201

1 INTRODUCTION

In recent years we have observed an increased prevalence of Internet of Things (IoT) devices being used to assist in our day-to-day activities and tasks. Great technological strides have been made since the early, simple devices and this can be seen by the number of homes adopting the technology. A study [25] of 83M IoT devices, across 16M homes in North America, shows that ~70% of homes have at least one IoT device. However, even a single unsophisticated device can be utilised as an attack vector for, e.g., privilege escalation and create problems within a network. Even more, cooperatively, a vast number of devices can perform coordinated tasks including those of a malicious nature such as distributed denial of service attacks (DDoS). A notable example of which being the IoT malware Mirai [6] launching a devastating attack on the "Krebs on Security" website, exceeding 600Gbps [24]. According to a Kaspersky Lab IoT report [26], the first half of 2018 saw IoT devices attacked by over 120,000 unique malware samples – three times as many as observed in the entirety of 2017. This alarming increase in attacks has shown a similar trend through to 2019 [27], and highlights the need for adequate protection against malware specific to IoT devices.

Current literature in the field proposes several approaches to tackling this problem, from host-based detection [9] through to network-based detection [21], using various techniques: in particular, a growing trend to defend against IoT threats is the use of supervised machine learning (ML) [3]. Many existing solutions have shown to have great accuracy, performance and the ability to understand and reason about threats in a way that humans may never be able to [42]. One of the most common classes of ML currently being explored in the literature is that of Artificial Neural Networks (ANN). Whilst these algorithms are highly effective [1], they have some drawbacks. In particular, they require labelled data, they often do not produce explainable models and are largely *black-box*, and suffer from issues in construction complexity. For instance, an issue often arises when we need to supply labelled data and training models. The range of possible manufacturers, protocols, and behaviours that a single IoT device may exhibit makes it difficult for an ANN to model a smart home network fully. Often a solution utilised is to identify the IoT devices in the network by type, to then train individual ANNs per type.

Another issue that arises is that IoT devices are often resource-constrained to allow for a small form factor in applying the connecting element to a standard device, such as a fridge or television. These resource constraints force developers to often only program for very limited and specific capabilities: this presents a natural method for device identification as well as malicious activity detection. These varying behaviours could be modelled as having varying *densities* in some n-dimensional space, which some existing clustering algorithms and techniques could address. So far, many of the proposed solutions to identifying the IoT device rely on clustering the network behaviours of the devices in a benign setting and using the resultant clusters as types to train on. Although clustering has been used in the security context for a number of purposes [17] [37] [7] [44], with varying results, research tackling the area of IoT are few and far between [40] [18].

With this work, we want to explore the applicability of clustering to the IoT field. In particular, we aim to make the case for a greater exploration of unsupervised learning in IoT IDS development and more specifically *density-based* clustering. To this end, we explore the feasibility of density-based clustering in anomaly detection by putting forward a candidate algorithm, namely *Ordering Points To Identify the Clustering Structure* (OPTICS) [5], along with a suite

ARES 2020, August 25–28, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-8833-7/20/08…$15.00
https://doi.org/10.1145/3407023.3409201

of supporting methods. With benignness modelled, and devices being different enough from each other in behaviour, the question arises whether these clusters could uncover enough information for anomaly detection without the need for further supervised learning steps. Additionally, whether clustering in itself can provide adequate detection for smart homes. This work will show the potential for models to be developed that allow for greater interpretability, and handle feature extraction and selection for such models that allow for protocol agnosticism which, in the field of IoT, would allow for greater coverage of protection.

Structure of the paper. In Sect. 2 we will cover the related work in the field, while in Sect. 3 we will outline the inner workings of the existing clustering algorithms underlying our proposed system. In Sect. 4, we will propose Clust-IT, a novel application of OPTICS-OF to various IoT datasets. Section 5, will present the results of our tests, while Sect. 6 will discuss the main findings. Finally, Sect. 7 will conclude the paper.

2 RELATED WORK

Intrusion detection, and specifically anomaly detection, has been an actively explored area of research for some years with a great number of techniques and systems proposed. However, fewer works are exploring the application of these detection techniques in IoT settings. Recent work from Mirsky et al. [32] proposes and implements a plug-and-play system based on an ensemble of autoencoders. An autoencoder is a type of unsupervised artificial neural network where, by keeping the number of layers and visible neurons to a minimum, the authors can achieve accuracy scores that are comparable to other state-of-the-art models, with the benefit of being able to learn in an online setting. However, utilising techniques stemming from neural networks have the pitfall of lacking any form of explainability or even interpretability, with the system's output simply being a function evaluating the error (e.g., Root Mean Square Error, RMSE) of the reconstruction error given an input. With this work, we hope to further explore algorithms which could provide greater scope for interpretability. Additionally, the data set provided for training uses only a handful of IoT devices with little diversity in class, and as such we hope to analyse datasets which can fully explore the range of heterogeneity available in current IoT devices.

Nguyen et al. [33] also propose a system utilising artificial neural networks with the use of Gated Recurrent Units (GRU) [13] in conjunction with a system comprising of multiple federated models to be aggregated. GRUs are based on the principles and construction of Long Short Term Memory (LSTM) [16] networks but require fewer parameters and lacks an output gate. While this allows for fast and efficient processing, they are outperformed by LSTMs in a variety scenarios such as learning languages and machine translation [43] [11]. Although allowing for the sharing of models by utilising a network-connected and federated approach, this increases the complexity in the necessary setup of a system.

Midi et. al. [31] propose a system based on knowledge-driven detection. By combining an external expert-system [28] with locally collected observations, the system can provide a level of protection and decision-making that an everyday user could not provide themselves. While performing very efficiently (if run on a separate IoT board), the proposed IDS suffers from the need for extensive expert knowledge. This process can be labour-intensive and, while knowledge database may be updated given constant connection to a server, there may be periods in the execution where new attacks are observed but the system may lack the adequate knowledge of such behaviours. In this paper, we endeavour to provide a candidate solution that requires minimal knowledge of the network setup and can learn in an unsupervised manner.

Within the field research of IoT IDS and security, many papers explore the impact, patterns of, and solutions to a variety of attacks, such as routing attacks [38] [2] [30], DDoS [22] [45] [19], spoofing and replay attacks [14] [15], and specific cases and proof-of-concept implementations of botnet attacks and campaigns [8] [41] [4]. Whilst many of these solutions focus on building a detection mechanism for a single class of attack, our proposed system aims to be attack-agnostic, detecting any shift from the normal behaviour expected. Furthermore, because our system focuses on the specific behaviours of devices of the network, our system provides a solution which may identify the early stages of a botnet campaign, infection and propagation, whereas many works focus on attacks launched by the botnet.

3 UNSUPERVISED LEARNING AND CLUSTERING

To describe and motivate our work, we begin by establishing some key ideas on which our system is built. We start by describing clustering and unsupervised learning before defining OPTICS and LOF, our candidate algorithms used in Clust-IT, to better understand their applicability within a smart home or general IoT context.

Where the general definition for supervised learning aimed to generate $g : X \rightarrow Y$, where X and Y are the input and output spaces resulting training pairs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, unsupervised clustering takes as input as set of input Observations X with no corresponding label set. In other words, unsupervised learning tasks deal with unlabelled data, i.e., features given without their corresponding outputs from the system. This definition can also be extended to describe problems which aim to infer structures, features and dependencies within a data set. There is a great range of algorithms to perform such tasks with distinct properties and results. Examples include dimensionality reductions such as Principle Component Analysis (PCA) [34] [35], a common use of which is to reduce a datasets dimensional space for visualisation purposes, and clustering.

3.1 Clustering

Clustering is the ML task that groups observations, or elements, in a dataset according to their similarities; an element in cluster A is more similar to the rest of its elements than those in cluster B. There is a great range of methods for defining similarity and cluster membership with just as large a variety of metrics to calculate those similarities. A simple clustering algorithm, though widely used, is K-Means Clustering [29].

The number of centroids, k, is specified by the user before runtime and represents the number of final clusters produced. Whilst this algorithm has successful uses, it does have several shortcomings. First of all, the necessity for an initial value for k can be a

problem in a task where the number of clusters in a set is unknown and is the aim of the task. There are variants, such as X-Means Clustering [36], which aim to solve the issue of defining an initial k, but an issue remains: forced cluster membership. The K-Means (as well as X-Means) algorithm insists that all observations belong to a cluster. However, this may not be necessary or helpful in every situation. This could present similar issues to those discussed earlier in supervised learning and classification tasks. If we consider IoT environments, where the output space is unknown, and theoretically of any size, a restricted output space may not be the desired solution, especially in the presence of new IoT malware variants, or even updated benign software updates and devices. This presents even more problems when an algorithm is exposed to datasets with noise or anomalous points. The K-means algorithms centroid and membership decision rely on the total sum of member points, and so the presence of erroneous or anomalous points may drastically alter the shape and membership of the clusters. Clustering algorithms, which allow for the presence of noise, and function without the need for an initial cluster number should be explored further for the security context.

3.2 OPTICS

A candidate solution to address shortcomings of DBSCAN and similar density-based clustering algorithms is OPTICS (Ordering Points To Identify the Clustering Structure). OPTICS requires two parameters to be set before running, namely ε which describes the maximum *distance* or *radius* to be considered, and $MinPts$, which describes the minimum number of points necessary to form a cluster. Using these values, OPTICS then proceeds to calculate two more values for each point in order to form a cluster ordering: *core-distance* and *reachability-distance*. These parameters allow OPTICS to consider points from more densely packed structures in its decision process. This enables OPTICS to produce and consider clusters of varying density while retaining the ability to have nested clusters. The $core\text{-}dist_{\varepsilon,MinPts}$ is defined as the smallest distance to the $MinPts\text{-}th$ to achieve a neighbourhood of $MinPts$ when less than the given ε, otherwise it is set to undefined. A point p is said to be a core point if at least $MinPts$ are found within its ε-neighbourhood $N_\varepsilon(p)$. The $reachability\text{-}dist_{\varepsilon,MinPts}(o,p)$ of a point q from point p is set as either the distance between o and p or the $core\text{-}dist$ of p, whichever is larger, but only if p is a core point.

Unlike traditional clustering algorithms, OPTICS produces a *cluster ordering* as opposed to cluster labels in the traditional sense, however, a clustering can be obtained through techniques similar to those used in hierarchical clusterings that produce dendograms of the similarities of observations. The *cluster ordering* is produced with the steps described in Alg. 1.

The resulting output is the list of points ordered by the respective reachabilities from the previous point with observations belonging to the same cluster being close to each other. As the produced list also contains the reachability distances, a graph showing the inter-point relationship can be visualised using a reachability graph as shown in Fig. 1. This allows for interpretation from the user in deciding the point at which to separate clusters but also to gain insight into the dataset which is, after all, what we are aiming for. When compared to other density-based clustering techniques,

Algorithm 1: OPTICS Algorithm for Cluster Ordering

```
OPTICS (D, ε, MinPts, Q);
Input  : Dataset D, neighbourhood distance ε, MinPts,
         Queue Q
Output : Cluster Ordering, Reachability Distances
Select random point p and set reachability to UNDEFINED;
while Q is not empty do
    Identify ε-neighbours of p;
    Write p to ordering output;
    Update Q with the identified neighbours of p;
    Update reachability distances of all points in Q and rank
      according to distance;
    Select nearest point in Q and assign as new p;
end
return Cluster Ordering and Reachability Distances;
```

OPTICS also offers other advantages. As mentioned previously, it offers the possibility to produce clusters of varying densities. When considering the IoT landscape, where devices have wide-ranging capabilities, this is a crucial feature. Also, in comparison to DBSCAN, it is much less sensitive to parameter adjustments. DBSCAN is known to be very sensitive to its input parameter, whereas OPTICS is much less so. Not only can an infinite value be given for ε (this could consider the whole dataset for neighbourhood memberships), but when changing $MinPts$, the resultant reachability graph can still produce interpretable results with similar results to those of differing $MinPts$ values. These features are key in the design and effectiveness of the proposed IDS solution outlined in further sections.

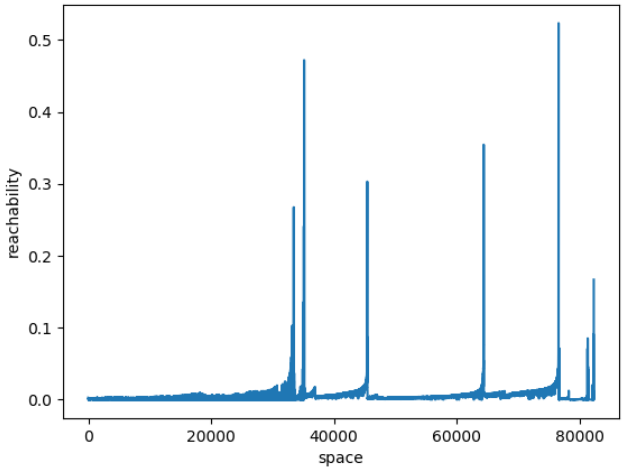

Figure 1: Example OPTICS Output

3.3 Local Outlier Factor

Local Outlier Factor (LOF) [10] is an algorithm that was proposed by the authors of OPTICS. It provides a score for the deviation of the

density of a point for its *k-nearest neighbours*. By utilising concepts outlined in OPTICS, namely *reachability-distance*, it assesses the neighbourhood density of a point and those of its neighbours to determine if it is an outlier. The benefits of such a scheme, and its use of comparisons to nearest neighbours, is that it allows for *anomalousness* to be location specific and dependant on local densities and not be measured by a global density.

The *reachability distance* used in LOF differs slightly from that used in OPTICS. It is defined as being either the *k-distance* of B, the distance to the *k-th* point from B, or the direct distance from A to B – whichever is greater. Formally we can say:

$$\text{reachability-distance}_k(A, B) = \max\{\text{k-distance}(B), d(A, B)\}$$

This definition of the *reachability density* is used to calculate the *local reachability density* (*lrd*), which is the inverse of the average distance at which A can be reached from its neighbours. These values for all *k-neighbours* of point A are then compared:

$$\text{LOF}_k(A) := \frac{\sum_{B \in N_k(A)} \text{lrd}(B)}{|N_k(A)|} / \text{lrd}(A)$$

which is the average local reachability density of the neighbours of A divided by the local reachability of A. A resulting value $A > 1$ indicates a lower density for A than its neighbours (outlier), whereas a value $A < 1$ indicates a higher density than its neighbours (inlier). Values of $A \sim 1$ indicate a similar density to its neighbours. In the context of this work, we will discuss later how the LOF value produced is used in tandem with the output of OPTICS to identify outliers; the former providing a faster, albeit more localised view into the context of points.

4 CLUST-IT

Many previous works have highlighted behavioural patterns as a key feature of IoT devices, and use this as an operational step in their solution for device identification. Many solutions use clustering for this step, but this is where the use of supervised learning often stops. We will outline the operational steps of our novel system, *Clust-IT*, making greater use of unsupervised learning than existing methods. We begin by outlining a set of requirements we aim to satisfy with Clust-IT as well as describing an assumed threat model we intend to work with. We finish the section by describing the architecture of our proposed system and its operational steps.

4.1 System Requirements and Threat Model

In this section we describe the requirements of the system, from a practical point of view and from a security point of view.

We design Clust-IT to address some key functionality requirements:

- **Interpretable**. A key issue of current supervised ML solutions is the difficulty in producing and interpretable model. Clust-IT addresses this by building into the system itself a degree of interpretability and explainability. Unsupervised learning, and more specifically clustering, aims to create subgroups in datasets. These groups can provide a platform for further manual analysis and even be used as inputs for more fine-grained clustering. Having a contextual view of portions of data is key in tackling issues of malicious activity and anomaly detection.
- **Adaptable**. Current supervised learning solutions often struggle in the presence of new attack paradigms and new families of malware. In a typical scenario, the used ML model is trained with both benign and malicious samples. However, this will show a degradation in performance as the definition of what it means for something to be malicious changes – a problem known as concept drift. Clust-IT addresses this issue by focussing on the key step of modelling benign behaviours, and including model retraining as a key step in maintaining information of the current state of the system and seen attacks.
- **Deployable**. Clust-IT aims to keep the necessary skill level to a minimum from a users perspective, and not rely on manufacturers to cooperate in usage of protocols or standards.
- **Passive**. To address both computational and complexity issues, Clust-IT works with data which will be passively captured form the home network, to remove the performance overhead and any issues of software interoperability.
- **Heterogeneous**. Clust-IT would aim to manage IoT heterogeneity issues by looking at the devices from a higher-level view, that is, using higher-level protocols relying on IP, namely TCP and UDP.

These solution and system requirements motivate our work and design choices. We hope that, by touching on these concepts, we may build a solution that is both scientifically sound as well as applicable to a wide range of scenarios, as is often the case in IoT networks.

Concerning Clust-IT's threat model, we focus on the threats that are due to IoT malware executing on, and interacting with the network. As we intend to build our model on the idea of benignness we take any action acting outside of this scope to be malicious. This could be activity originating from the device itself or the device being the target. With this approach, we can consider attacks at various stages in their operation from infection and propagation to carrying out the intended malicious action, as well as user misconfigurations which may lead to unintended consequences. These details may lead to further vulnerability and must be treated as anomalous.

We assume that malware traffic will often be *opaque*, that is to say, in some way made unreadable to the user or a monitoring solution through obfuscation, packing or encryption. We, therefore, alter our monitoring and collection strategy to account for such cases. Note that we do not employ deep packet inspection or general packet analysis, and we only deal with packet header statistics.

4.2 System Design

We assume Clust-IT to be running on a smart home network with a variety of devices in range of device *types* such as entertainment, safety, kitchen and others. We deploy Clust-IT on a single point of entry in a typical smart home environment, i.e. the router where we mirror all traffic to our system. This allows Clust-IT to passively capture and detect all activities without affecting the performance either computationally or in communication speeds of the devices. The architecture and the application scenario is depicted in Fig. 2.

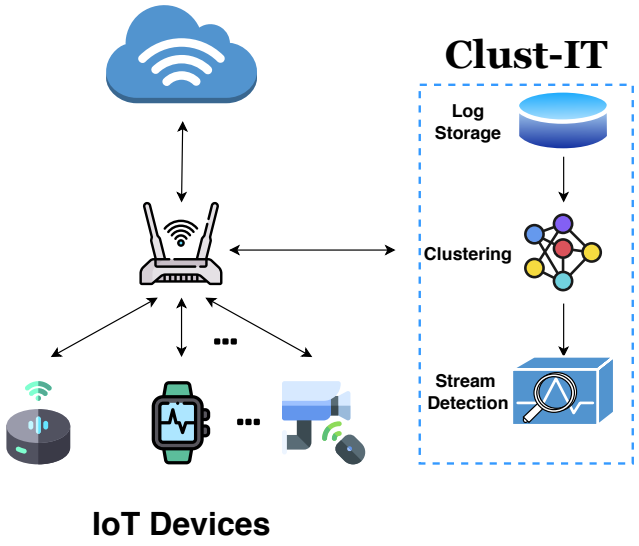

Figure 2: Clust-IT Architecture

The model itself considers that IoT adoption in homes will become more prevalent, with more heterogeneous devices being connected. Please note that, although in this work we primarily focus on the algorithmic efficacy of deploying clustering algorithms for detection purposes in this deployment scenario, we design Clust-IT with a series of high-level operational steps as to make it applicable beyond this potential scenario.

Clust-IT's *detection* system begins with a phase of *collection*. That is to say, the system would monitor and store the traffic to be processed. Although in our threat model we assume benign activity during this process, the stored activities present a ground truth for *expected* behaviour, and as such still allow for us to detect anomalies within our system. Additionally, through our use of the noise-tolerant OPTICS, we would expect behaviours that vary wildly from the usual defined device range to be discovered in our initial cluster generation.

Using the initial monitor period data, we prepare our data for use and construct our clusters utilising the aforementioned OPTICS cluster generation algorithm. The expected cluster output would aim to encapsulate most devices' limited activity within coherent clusters with little noise. We then assign the resultant cluster labels to each data point encapsulated. During normal operation of the proposed system, the computed clusters will be compared with previous runs to determine wildly varying shifts in behaviours shown by the devices. Furthermore, activities which fall outside of known labelled clusters, or labelled as noise, will be regarded as anomalous and as such malicious.

To allow for stream detection capabilities we employ a combination of k-nearest neighbours (KNN) and LOF. Once clusters are generated and labels have been propagated, we have a baseline for expected behaviours. A new incoming packet, once processed as done with packets clustered, will be assigned a label according to KNN. To add a level of confidence, and to avoid excessive false positives and negatives, we utilise LOF to determine how similar the point is with respect to its neighbours and their classes. We set a threshold for LOF, with points that fall as outliers being detected as anomalies. While with work we preliminary evaluate this method on the whole dataset, we may further expand on this by restricting the cluster membership detection by only considering previous instances of its behaviour or instances of a certain class of devices such as entertainment or lighting.

To maintain the cluster novelty, and to adjust for changes in device usage, or other interferences in normal usage, such as protocol or software updates, we recompute the clusters after a given elapsed time period. To be consistent with current IoT networks, the recomputation happens on a daily basis, although shorter (e.g., every few hours) or longer periods (e.g., every week) might be required depending on the network activity and updates in the configuration. When a re-computation happens, packets that have been deemed malicious or anomalous may be removed or labelled as such and be added to the datasets to allow for cluster labels to span attacks, or to address the issue of a newly added device to the network whose new behaviour was deemed anomalous. This re-computation step also allows for attacks that may have previously been detected as benign to be detected in the context of the behaviour of the whole network. The operational behaviour of Clust-IT is summarised in Alg. 2.

Algorithm 2: Operation of Clust-IT

Input : Logs L, LOF threshold T, new activity a, previous clusters C
Output: Detection decision and updated clusters

```
Compute clusters for L with OPTICS;
Take mode of true labels of cluster members and apply to whole cluster;
while Clust-IT is in stream detection do
    for each new activity a do
        Store in L;
        Take k-nearest neighbours;
        Take mode of their cluster labels from C;
        Perform LOF including k-nearest neighbours;
        if LOF(a) < T then
            Label as noise;
            Raise alarm;
        end
        else if LOF(a) >= T and mode = malicious then
            Label as malicious and raise alarm;
        end
        else
            Store a in L;
        end
    end
    if End of Detection Period then
        Recompute Clusters with L;
    end
end
```

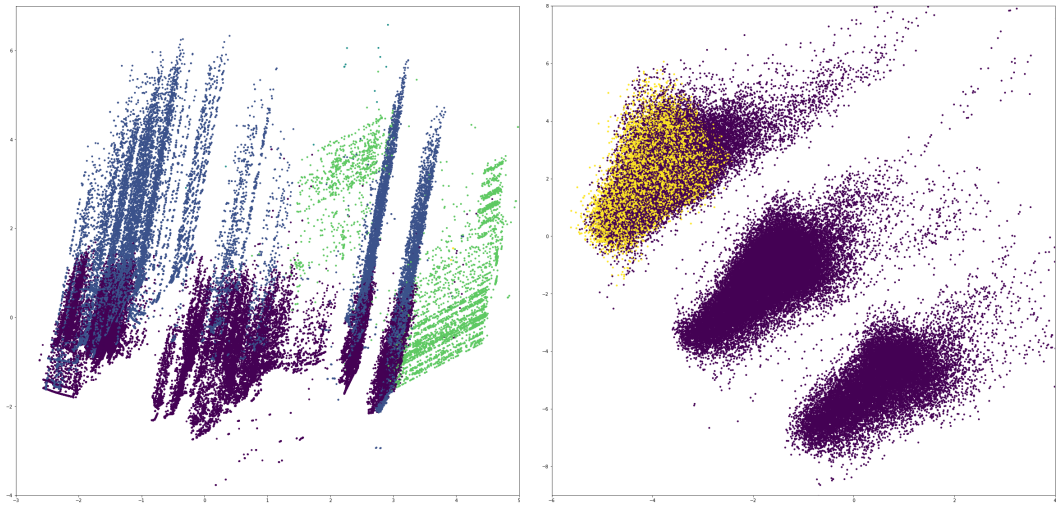

Figure 3: PCA Data Scatter with Ground Truth Labels of Dataset [23] and [32]

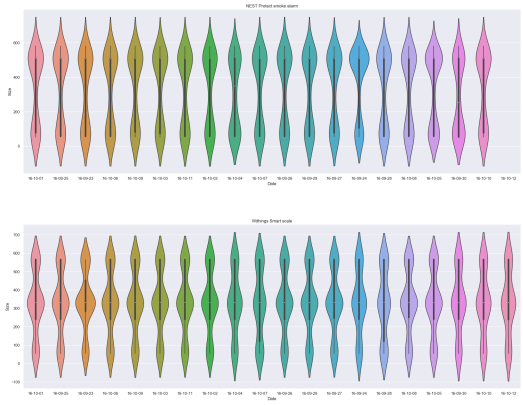

Figure 4: Quantile Violin Plots of 2 IoT devices: A NEST Protect Smoke Alarm and a Withings Smart Scale

5 EVALUATION

To evaluate the efficacy of clustering for detection in an IoT or smart home setting, we test our candidate algorithms and solutions on numerous available IoT datasets; malicious and benign. We evaluate each candidate algorithm on every collected dataset, as well as comparing against the current state of the art solutions that have used the given dataset. By evaluating against all datasets we hope to determine the transferability of the models to differing network setups. Recent work such as [25] has studied the extent to which devices are sued in the home as well as what specific types of devices are used. With Clust-IT, we use a variety of datasets which have their view of a *smart home*, as a testbed for our configuration. Among the types of devices used across the used datasets include smart televisions, Amazon Alexas, smart switches and a wide range of others.

5.1 Datasets

Several datasets [39] [23] [32] are used to evaluate the efficacy and performance of our proposed approach. Across the datasets, a set of common features is used, albeit in potentially an altered state. These features are: (i) Time, (ii) Size, (iii), Ethernet Source and Destination, (iv) IP Source and Destination, (v) IP Protocol, (vi) Port Source and Destination.These features are obtainable from packet headers and are protocol agnostic outside of IP protocols and satisfy our requirements outlined in Sect. 4.2.

We will first refer to the dataset collected from 30 IoT devices spanning a period of 20 days by researchers at the University of New South Wales Sydney [39]. We use this dataset as a model in an attempt to better illustrate Clust-IT expected behaviours in benign IoT settings, and to answer our first research question: do IoT devices, with their limitations in resource and therefore in behaviour, display certain patterns in data transmission for adequate identification when under the analysis of clustering? The data includes traces from a network consisting of 28 different IoT devices run under the assumption of no malicious activity present. Figure 4, depicting violin plots of the distribution of packet sizes sent for 2 different devices per day, presents a consistent behaviour for each monitored day. We, therefore, test the hypothesis that such a uniform behavioural pattern emerging from devices can be used to form coherent clusters which can then be used for intrusion and anomaly detection.

Utilising these same features, we use the dataset [23], and after reducing with PCA we display along with a ground truth in Fig. 3, showing the beginnings of logical cluster structures. Although, PCA could be said to be performing much of the anomaly detection here we show that in a differing dataset from [32] that less clear structures are present. Finally, to compare against current *state of the art* solutions we use datasets as used in [32] and [23] which both contain both malicious and benign observations and, in the case of [23] (BOT-IoT) a unified dataset with a variety of attacks.

	Kitsune: Mirai	Kitsune: Fuzzing	Kitsune: OS Scan	BOT-IoT
Clust-IT	DR: 0.936 FPR: 0.053	DR: 0.803 FPR: 0.234	DR: 0.958 FPR: 0.025	DR: 0.912 FPR: 0.039
K-Means	DR: 0.899 FPR: 0.114	DR: 0.812 FPR: 0.180	DR: 0.959 FPR: 0.023	DR: 0.791 FPR: 0.117
Birch	DR: 0.840 FPR: 0.002	DR: 0.801 FPR: 0.238	DR: 0.960 FPR: 0.041	DR: 0.769 FPR: 0.194
Kitsune [32]	DR: 0.997 FPR: 0.001	DR: 0.995 FPR: 0.001	DR: 0.010 FPR: 0.001	

Table 1: Clust-IT and Candidate Algorithm Detection Performance for Malicious Datasets [32] and [23]

The datasets presented in [32] also provides a range of attacks, but as separated instances and datasets.

5.2 Experiment Setup

To evaluate Clust-IT and compare against the varying datasets and solutions we create a pipeline which we apply to all our chosen datasets. We begin with a step of feature *pre-processing*, a standard step in ML tasks, removing missing features and standardising all features and scaling to unit variance. Following this step we apply Principal Component Analysis (PCA) to all datasets with *n=3* for the number of principal components. We split our datasets into *train* and *test* partitions, with our test set totalling 0.2 of the whole dataset.

We construct our models utilising the scikit-learn API [12], using their existing implementations of OPTICS, LOF, K-Means, and K-Neighbors, and do so as individual instances for each evaluation dataset. Once clusters have been achieved and extracted, for each cluster, we take the mode of the true labels within each cluster and assign it to the whole cluster. Therefore, each resulting test that would be placed within the area of this produced cluster would take the clusters derived label. In the case of K-Means, assigning a cluster label to new input is simply locating the nearest centroid to the point. In the case of OPTICS, we take *n-nearest* neighbours, taking the average of the labels, as well as taking the LOF for the extracted neighbours. Combining this score we determine if the new point belongs to the designated cluster label or should be treated as noise.

We measure the performance of Clust-IT using measures of *Detection Rate* (*DR*) and *False Positive Rate* (*FPR*). We define *DR* as the ratio of observations correctly labelled as malicious, and we define ($FPR = \frac{FP}{FP+TN}$). We aim for as high a value for *DR* while minimising *FPR*; an abundance of false alarms in intrusion detection systems may lead to overuse of resources and both computationally as well as human interaction needed to investigate said alarms.

5.3 Results

Table 1 presents our results across the used datasets. The *state-of-the-art* results presented are those presented in the original papers which constructed the relevant dataset. As shown from our results, Clust-IT has a high performance rate when compared to a naive clustering algorithm, such as K-Means and BIRCH (balanced iterative reducing and clustering using hierarchies), and can achieve high detection rates comparable to those presented by other solutions or algorithms. In the case of *fuzzing* detection, we do observe a high *FPR* and lower *DR* than achieved in other datasets. Although we train separate instances of OPTICS for each dataset presented, we retain the exact same settings and parameters during the training and cluster generation process. Additionally, we apply the same *pre-processing* steps to all datasets. For example, once the data has been cleaned, we apply a dimensionality reduction algorithm PCA to the dataset, with an *n=3* for the number of principle components. With further data preparation, including reduction of observations used, we would expect positive changes in the performance specific to each case. However, even in the case of the reduced performance, especially compared to the other algorithms, we observe that Clust-IT has a more consistent high performance across all datasets. We make the decision to apply the same settings to all datasets in an attempt to show the applicability and transferability of our proposed system.

The solution presented in [32] relies on training the solution to achieve a maximum false-positive rate and as such the statistics for its *FPR* remain the same. Although the combined performance achieves great detection and true positive rates, it shows a great decline in its detection when analysing OS Scan data. Our solution, Clust-IT, manages to achieve a high detection rate, further showing the potential for transferability.

6 DISCUSSION

Whilst having secure detection mechanisms in place for malicious activity, many current solutions lack inherent further analysis of security problems. Whilst not fully explored in this work, clustering by design is a tool whose use is most suited to data exploration. Past the point of detection, clustering offers a greater level of *interpretability* than other supervised or unsupervised solutions. Being able to have deep dive in clusters whose members are deemed to be mathematically similar, or significant when analysing noise or outliers from LOF, is a feature that could show applicability beyond smart homes. With the rise of smart cities and businesses more widely using IoT devices in their operations, a SOC team could gather vital intel using more techniques allowing this greater level of data exploration. Further work would explore this potential for threat intelligence *post-detection*.

With this work, we have highlighted some common issues experienced in ML-driven solutions to detection in IoT, as well as provided some insight into the feasibility of clustering as a solution, which offer potential directions for research to tackle these problems. Although our results described in Sect. 5 show signs

of promise, they are not without their shortcomings. One such shortcoming is the problem of runtime. If we are to assume our *training* phase to be cluster generation then, similar to many supervised learning algorithms, it may necessitate significant time and computing power. However, following this stage, a classification model, such as a k-NN or SVM, will be able to classify new inputs at speed. Querying for cluster membership on new data in the case of a K-Means output too requires little time, needing only to find its nearest centroid. However, in the case of OPTICS, this is not possible without recomputing the entire cluster ordering. The shortcoming of this algorithm is the basis on which we have developed our operational steps, to allow for quick, stream like detection with LOF and k-NN, but then require re-computation of the clusters at regular time intervals. Furthermore, both in the case of OPTICS and K-Means, as well as many other clustering algorithms, the presence of new points may hugely alter the structure and shape of the resultant clusters, again making re-computation necessary at regular intervals. However, this could be compared to the issue of concept drift [20] experienced in supervised learning solutions to malware detection.

There are further caveats with clustering when considering cluster quality. Many ML classification algorithms have a *score* function inherent to its running, especially when in its training phase. This enables methods, such as *grid search*, to be employed to help with parameter and features optimisation. Cluster algorithms when used in the way we propose present problems in assessing the cluster quality and resultant detection capabilities for optimisation. Measures, such as *correctly clustered*, can become difficult to assess especially when working with an undefined number of clusters. Clusters themselves can be assessed with measures as used in 5, such as *homogeneity*, *completeness* and *v-measure*. However, the correct use of these assessments relies on deep knowledge of the dataset and expected cluster output. With an algorithm such as OPTICS which may output a variable number of clusters and noise, such measures may hinder optimisation. For example, an instance of traffic containing few malicious or *different from normal* observations, may be optimised such that all data registered as noise as such a cluster output would optimise the resultant *homogeneity* or *completeness* scores. Further issues arise with other cluster assessment scores such as *Silhouette Coefficient Score*. This score evaluates the *intra-cluster* distance and *mean nearest-cluster* distance for each sample. This assumes a relatively uniform clustering structure with logical centroids assignable to each. In the case of OPTICS, which relies on a density-based approach and its ability to produce clusters of arbitrary shape and size, as well as allowing for sub-clusters, the silhouette score for our resultant clusters may often be very low, but not indicative of the true cluster quality. Further work would explore adequate cluster scoring mechanisms to determine the quality of our produced model.

Whilst the datasets used for our evaluation are developed with real IoT devices, there are questions concerning the data being considered as a real-world scenario. In the case of [32], the network is composed of a combination of virtual machines and IoT devices. Although the presence of virtual machines may not be too far-fetched, the limited range and scope of devices used in the dataset may hinder the transferability of the dataset. Additionally, the attacks deployed on the network are run independently of each other on isolated instances, and while it is clear the work shows the efficacy of their solution within an IoT network, a case which includes a variety of attacks could help develop more complex detection mechanisms. In the case of the dataset used for [23], although the dataset is used as a dataset for botnet detection and correlation, the data split of malicious/benign observations is weighted heavily toward the malicious. Whilst it may serve its purpose in its works presented use-case, it presents an unrealistic environment for real-world solutions for intrusion detection and model building. This highlights the issue present in the IoT and smart home field that datasets are hard to come by, and their activity shows very specific behaviour. That is to say, the datasets are not in and of themselves flawed, but the general lack of datasets in IoT research presents challenges for researchers hoping to evaluate potential solutions.

7 CONCLUSION

IoT is becoming ever more pervasive in our lives and is increasingly heterogeneous. This increases the potential attack landscape for would-be malicious actors. This expanded threat landscape requires us to create solutions which can be easily applied to systems containing many differing devices, and susceptible to a variety of attacks. We have explored the feasibility of clustering in IoT landscapes for intrusion detection and our preliminary results show the promise of our proposed solution, called Clust-IT. Additionally, we have outlined the caveats of utilising clustering, and more specifically density-based clustering, as a solution to this field. We propose further work which would seek to develop optimisation strategies that incorporate the whole detection pipeline of the system to improve the runtime. In addition, we aim to to factor in realistic proportions of malicious traffic when introducing attacks.

ACKNOWLEDGEMENTS

This research of Robert P. Markiewicz is supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/P009301/1). Daniele Sgandurra's work was partially funded by European Union's Horizon 2020 research and innovation programme under grant agreement No 779391 (FutureTPM).

IoT icons made by Freepik from www.flaticon.com and additional cluster icon made by Becris from www.flaticon.com as used in Figure 2.

REFERENCES

[1] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon 4*, 11 (2018), e00938.

[2] Airehrour, D., Gutierrez, J., and Ray, S. K. A lightweight trust design for iot routing. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)* (2016), IEEE, pp. 552–557.

[3] Al-Garadi, M. A., Mohamed, A., Al-Ali, A., Du, X., Ali, I., and Guizani, M. A survey of machine and deep learning methods for internet of things (iot) security. *IEEE Communications Surveys Tutorials* (2020), 1–1.

[4] Angrishi, K. Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. *arXiv preprint arXiv:1702.03681* (2017).

[5] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 1999), SIGMOD '99, ACM, pp. 49–60.

[6] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., and Zhou, Y. Understanding the mirai botnet. In *USENIX Security Symposium* (2017).

[7] Bayer, U., Comparetti, P. M., Hlauschek, C., Kruegel, C., and Kirda, E. Scalable, behavior-based malware clustering. In *NDSS* (2009), vol. 9, Citeseer, pp. 8–11.

[8] Bertino, E., and Islam, N. Botnets and internet of things security. *Computer*, 2 (2017), 76–79.

[9] Breitenbacher, D., Homoliak, I., Aung, Y. L., Tippenhauer, N. O., and Elovici, Y. Hades-iot: A practical host-based anomaly detection system for iot devices (extended version). *arXiv preprint arXiv:1905.01027* (2019).

[10] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2000), SIGMOD '00, ACM, pp. 93–104.

[11] Britz, D., Goldie, A., Luong, M.-T., and Le, Q. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906* (2017).

[12] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122.

[13] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[14] Emerson, S., Choi, Y.-K., Hwang, D.-Y., Kim, K.-S., and Kim, K.-H. An oauth based authentication mechanism for iot networks. In *2015 International Conference on Information and Communication Technology Convergence (ICTC)* (2015), IEEE, pp. 1072–1074.

[15] Esfahani, A., Mantas, G., Matischek, R., Saghezchi, F. B., Rodriguez, J., Bicaku, A., Maksuti, S., Tauber, M. G., Schmittner, C., and Bastos, J. A lightweight authentication mechanism for m2m communications in industrial iot environment. *IEEE Internet of Things Journal 6*, 1 (2017), 288–296.

[16] Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)* (1999), vol. 2, pp. 850–855 vol.2.

[17] Gu, G., Perdisci, R., Zhang, J., and Lee, W. Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th Conference on Security Symposium* (USA, 2008), SS'08, USENIX Association, p. 139–154.

[18] Hafeez, I., Antikainen, M., Ding, A. Y., and Tarkoma, S. Iot-keeper: Detecting malicious iot network activity using online traffic analysis at the edge. *IEEE Transactions on Network and Service Management 17*, 1 (2020), 45–59.

[19] Hamza, A., Gharakheili, H. H., Benson, T. A., and Sivaraman, V. Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity. In *Proceedings of the 2019 ACM Symposium on SDN Research* (2019), ACM, pp. 36–48.

[20] Jordaney, R., Sharad, K., Dash, S. K., Wang, Z., Papini, D., Nouretdinov, I., and Cavallaro, L. Transcend: Detecting concept drift in malware classification models. In *26th USENIX Security Symposium '17* (Vancouver, BC, Aug. 2017), USENIX Association, pp. 625–642.

[21] Kasinathan, P., Pastrone, C., Spirito, M. A., and Vinkovits, M. Denial-of-service detection in 6lowpan based internet of things. In *2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob)* (2013), IEEE, pp. 600–607.

[22] Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. Ddos in the iot: Mirai and other botnets. *Computer 50*, 7 (2017), 80–84.

[23] Koroniotis, N., Moustafa, N., Sitnikova, E., and Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, 2018.

[24] Krebs, B. KrebsOnSecurity Hit With Record DDos, 2016.

[25] Kumar, D., Shen, K., Case, B., Garg, D., Alperovich, G., Kuznetsov, D., Gupta, R., and Durumeric, Z. All things considered: an analysis of IoT devices on home networks. In *28th USENIX Security Symposium '19* (2019), pp. 1169–1185.

[26] Lab, K. New iot-malware grew three-fold in h1 2018. https://www.kaspersky.com/about/press-releases/2018_new-iot-malware-grew-three-fold-in-h1-2018, 2017. Accessed: 2019-10-18.

[27] Lab, K. Iot under fire: Kaspersky detects more than 100 million attacks on smart devices in h1 2019. https://www.kaspersky.com/about/press-releases/2019_iot-under-fire-kaspersky-detects-more-than-100-million-attacks-on-smart-devices-in-h1-2019, 2019. Accessed: 2019-10-18.

[28] Leondes, C. T. *Expert systems: the technology of knowledge management and decision making for the 21st century*. Elsevier, 2001.

[29] Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory 28*, 2 (1982), 129–137.

[30] Mehta, R., and Parmar, M. Trust based mechanism for securing iot routing protocol rpl against wormhole &grayhole attacks. In *2018 3rd International Conference for Convergence in Technology (I2CT)* (2018), IEEE, pp. 1–6.

[31] Midi, D., Rullo, A., Mudgerikar, A., and Bertino, E. Kalis—a system for knowledge-driven adaptable intrusion detection for the internet of things. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (2017), IEEE, pp. 656–666.

[32] Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089* (2018).

[33] Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., and Sadeghi, A.-R. Dïot: A federated self-learning anomaly detection system for iot. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019), IEEE, pp. 756–767.

[34] Pearson, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572.

[35] Pei, J., Hu, Y., and Xie, W. Pca-based visualization of terahertz time-domain spectroscopy image. In *MIPPR 2007: Multispectral Image Processing* (2007), vol. 6787, International Society for Optics and Photonics, p. 67871M.

[36] Pelleg, D., Moore, A. W., et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml* (2000), vol. 1, pp. 727–734.

[37] Perdisci, R., Lee, W., and Feamster, N. Behavioral clustering of http-based malware and signature generation using malicious network traces. In *NSDI* (2010), vol. 10, p. 14.

[38] Raza, S., Wallgren, L., and Voigt, T. Svelte: Real-time intrusion detection in the internet of things. *Ad hoc networks 11*, 8 (2013), 2661–2674.

[39] Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., and Sivaraman, V. Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing 18*, 8 (2019), 1745–1759.

[40] Sivanathan, A., Gharakheili, H. H., and Sivaraman, V. Inferring iot device types from network behavior using unsupervised clustering. In *2019 IEEE 44th Conference on Local Computer Networks (LCN)* (2019), pp. 230–233.

[41] Soltan, S., Mittal, P., and Poor, H. V. Blackiot: Iot botnet of high wattage devices can disrupt the power grid. In *27th USENIX Security Symposium '18* (2018), pp. 15–32.

[42] Tschandl, P., Codella, N., Akay, B. N., Argenziano, G., Braun, R. P., Cabo, H., Gutman, D., Halpern, A., Helba, B., Hofmann-Wellenhof, R., Lallas, A., Lapins, J., Longo, C., Malvehy, J., Marchetti, M. A., Marghoob, A., Menzies, S., Oakley, A., Paoli, J., Puig, S., Rinner, C., Rosendahl, C., Scope, A., Sinz, C., Soyer, H. P., Thomas, L., Zalaudek, I., and Kittler, H. Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study. *The Lancet Oncology 20*, 7 (2019), 938 – 947.

[43] Weiss, G., Goldberg, Y., and Yahav, E. On the practical computational power of finite precision rnns for language recognition. *arXiv preprint arXiv:1805.04908* (2018).

[44] Wicherski, G. pehash: A novel approach to fast malware clustering. *LEET 9* (2009), 8.

[45] Zhang, C., and Green, R. Communication security in internet of thing: preventive measure and avoid ddos attack over iot network. In *Proceedings of the 18th Symposium on Communications & Networking* (2015), Society for Computer Simulation International, pp. 8–15.