

Self-Configuring NetFlow Anomaly Detection using Cluster Density Analysis

Kieran Flanagan*, ***, Enda Fallon*, Abir Awad**, Paul Connolly***

*Software Research Institute, Athlone Institute of Technology, Athlone, Ireland

**Faculty of Computing, Engineering and Science, University of South Wales, UK

***The NPD Group, Inc, IDA Business Park, Athlone, Co. Westmeath, Ireland

k.flanagan@research.ait.ie, efallon@ait.ie, abir.awad@southwales.ac.uk, paul.connolly@npd.com

Abstract— The growing number of malicious network attacks has resulted in the need for a fast, reliable method to identify possible malicious activity. For any organization, it is critical that confidential and proprietary data is sufficiently secured to address both legal and contractual obligations. The changing nature of security attacks has caused a surge of interest in anomaly detection mechanisms. Such mechanisms are suitable as they can dynamically adapt to changed network conditions and threats without security personnel intervention. While anomaly detection mechanisms have significant potential, they are technically limited. Many anomaly detection approaches are unsuitable for real time environments. The approaches also typically operate based on “what is common is normal”. Mechanisms are typically singular in focus analysing data on one specific type. This paper proposes a novel framework to detect anomalies previously hidden within current detection techniques. The approach is easily extensible taking input from many security assessment applications; network traffic, asset criticality. Using time based correlations with historic data; a method for generating a normalized view of activity on the network is achieved. Once normality has been established for specific time intervals an extensible environment is implemented which allows for the active monitoring of anomalies in real-time. Anomalies which had sufficient “commonality” to remain undetected by other mechanisms are identified and analysed. The proposed solution is completely autonomous, capable of acting independently with no previous knowledge required. The presented results describe NetFlow activity of the NPD Groups’ network over a 24-hour period and outline real world anomalies that were detected.

Keywords— Anomaly Detection, NetFlow, Clustering, Density Analysis

I. INTRODUCTION

As a result of the growing number of malicious network attacks, research into new methods of anomaly detection has increased in prominence. The need for a fast, reliable method to identify possible malicious activity has grown in response to emerging threats. Protecting confidential and proprietary data is of paramount importance to any organisation to ensure that both legal and contractual obligations are kept. In addition, the data stored may not necessarily be the property of the company storing and handling the data. Malicious activity such as Botnets and Port Scans are increasing in frequency.

These attacks, while simple, have the potential to allow for unauthorised access onto the network.

Within any organisation, it is common to use network monitoring and analysis tools to identify anomalous behaviour on the network. Tools, such as McAfee ePO and Tipping Point are signature based models, which require a known example of a threat to be catalogued and a signature generated. The signature based model, while highly accurate, fails if a novel attack occurs (e.g. zero-day vulnerabilities), as no previous signature exists. This limitation gave rise to anomaly based detection mechanisms. Such methods require no signature database, but instead model the “normal” traffic on a network and alerts to any activity that happens outside of these normality bounds.

While much recent research has focused on anomaly based systems [1]–[5], key limitations apply when attempting to adapt these approaches to a real-time system. Foremost amongst these limitations is computational cost. Within commonly used distance based outlier detection mechanisms, the need for distance based calculations for each new sample can be overwhelming for high volume data. This gave rise to optimised algorithms designed to mitigate this limitation. Algorithms such as Fast Local Outlier Factor (FastLOF) [6] and Micro-Cluster based Outlier Detection (MCOD) [7] reduce the overall cost of the range queries with varying degrees of success.

These approaches are technically limited as they assume that commonality of similar samples equates to normality. Within the context of NetFlow analysis, this assumption limits the anomaly detection capability of recognising changes in traffic behaviours. As an example, a large increase in a particular type of traffic may be indicative of data exfiltration. Traditional methods categorise this increase in activity, and equate this to an increase in normality.

This paper proposes a system in which we extend the anomaly detection capabilities of MCODE, creating MCODET (Micro Clustering Outlier Detection in Time-series.). Additionally, we create an abstraction level to equate a change in commonality to a change in normality. In doing so, an extensible framework is generated. This enables the proposed system to detect changes in the underlying network behaviour that may be indicative of malicious behaviour or other possible abnormalities.

This paper is organised as follows. In Section II, an overview of related work is given. In Section III, a brief overview of the technology used is presented. In Section IV, we propose a framework for the identification of anomalies within NetFlow data. The architecture is presented as well as an overview given on algorithms created and used. In Section V, testing methodology is presented. In Section VI, results obtained from live data are presented and analysed. In Section VI, conclusions and future work are presented.

II. RELATED WORK

Recent research into anomaly detection has largely focused on applying anomaly detection mechanisms on network data in order to successfully identify malicious behaviour. The data processing requirements of such approaches however make them unsuitable for real time network analysis. Methods such as principal component analysis [3], K nearest neighbour [8] and ensemble techniques [9], [10] have limitations when used to analyse real time network streams.

The relatively expensive computational requirements of such approaches have raised interest in clustering techniques. Such techniques mitigate the calculations needed when associating anomaly detection with big data. [11], [12]. Aggregated data, such as NetFlow streams, can be used to reduce the calculations further [13]. Several limitations have been identified with such approaches. Kumari et al [14] analyses a clustering technique for use with an anomaly detection system for network data. The approach sets a distance based threshold as the 100-th farthest data point from the obtained cluster centroids. This threshold is a common theme across multiple anomaly detection solutions [15]–[17]. This work illustrates that using a common threshold over all clusters within real data is non-representative of the various forms of traffic created. As an example, traffic from different applications does not act in a similar manner.

While distance based outliers have been shown to be of significance [18]–[20], this work proposes a novel indicator of possible anomalous behaviour. By monitoring cluster density over a time series, changes in underlying behaviour can be detected. Rather than only focusing on samples that are anomalous via distance based calculations, changes in the density of activity over time are also monitored.

Time series analysis of network streams has been shown to be effective in [17] and [21]. [17] demonstrates the use and effectiveness of utilizing a time series based approach over live data. The aggregation of traffic is highly beneficial in this case, as it reduces the computational sources needed. Furthermore, it also provides a tangible benefit when comparing results across clusters. This idea of time-aware analysis has been used elsewhere also [21], and has shown to be a valuable tool in mapping continuous behaviour.

This research extends such works by creating a novel method of categorizing changes in normality for specific time series over time. The approach is advantageous as it (a) abstracts heterogeneous network traffic types into a common view thereby increasing situational awareness throughout the network (b) actively monitors both persistent and intermittent

clusters (c) monitors clusters independently, capturing and predicating the unique behaviour within the clusters subspace.

III. TECHNOLOGY OVERVIEW

A. Cisco NetFlow

Cisco NetFlow is a system that amalgamates network traffic information into a format that successfully describes communications occurring on a network. Through a NetFlow enabled device (Fig. 1), packet traces are identified and stored as a single flow representation of a specific set of communications between two devices. These are used for multiple tasks such as network performance monitoring and used as a means of security evaluation when an incident has been detected. Visualization of the NetFlow has also proven to be of tangible benefit [22].

Using this aggregated data for anomaly detection has benefits, such as data size being reduced for processing purposes and storage. For the analysis in this paper, NetFlow was used as it was found that using NetFlow for network monitoring purposes was highly common in the area [23], as well, in this case, the current infrastructure of NPD allowed for the collection of NetFlow with relative ease.

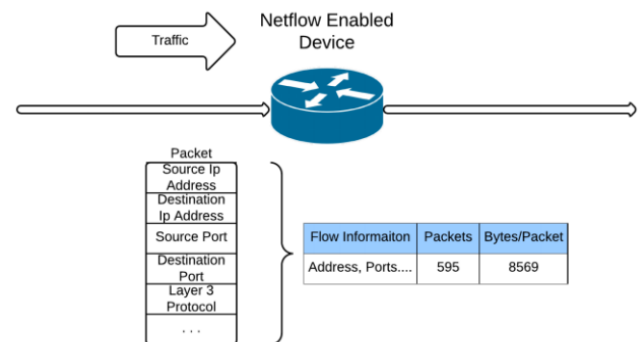


Figure 1. NetFlow Example

B. MCODE

MCOD is a distance based outlier detection mechanism that utilizes clustering to reduce the amount of distance based calculations needed to identify possible anomalies. This reduced performance cost of the algorithm makes it an ideal candidate for real-time anomaly detection over data streams when compared to other distance based algorithms such as Local Outlier Factor (LOF). As described in [7], MCODE uses a sliding window approach to identify outliers over a data stream. By using an expiring data set, the algorithm can be optimized to only use a data set that is large enough for satisfactory anomalies to be detected, while maintaining the low amount of calculations needed. Distance is calculated between cluster centroids and the NetFlow. If the point is within this range of a cluster centroid and the cluster has the specified density, k , then the point is determined to be a non-anomaly, or inlier.

IV. PROPOSED FRAMEWORK

The architecture is a two-step approach that involves monitoring traffic at different levels of abstraction. Firstly, an adaptation of the MCOD algorithm is applied on the NetFlow data in a series of time windows. Following this, results of the anomaly detection algorithm are passed into an abstraction layer that uses a polynomial regression algorithm to create a generalization of the cluster densities generated by the first step over time.

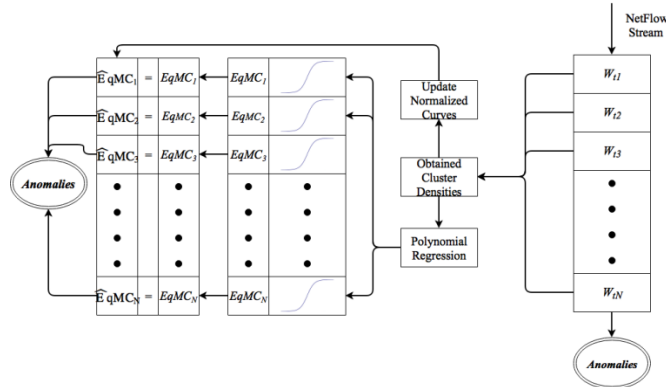


Figure 2. Framework Architecture

A. Anomaly Detection

When using algorithms such as k-Nearest Neighbour and MCOD, a common variable that must be tuned to the dataset is k . While its exact use may change, the principal remains consistent. k is used to describe the limit of normality within anomaly detection frameworks. It is a single variable used to tune the classification or detection rate within an algorithm. Within MCOD, it is the value at which a micro cluster is generated and an outlier is classified as an inlier. Within the context of NetFlow anomaly detection, this is a severe limitation, as it assumes that all network traffic activity has an equal distribution across the network, which is an extremely unlikely assumption (Discussed further in Section V).

To successfully mitigate this limitation, all cluster densities are monitored irrespective of the threshold denoted by k . This allows for the successful detection of abnormal increases and decreases in cluster densities. k is now only representative of a binary decision. To generate a cluster and monitor its density over time in the second layer of the architecture, or maintain independent points in layer one and identify them as anomalous. This method essentially transforms k from a measure of normality to a performance only metric.

B. Signature Generation

Within the proposed architecture this modified version of the MCOD algorithm is applied within various time windows (Fig.1, Fig.2). The cluster densities obtained from within each time window is polled at regular intervals and correlated over a three-hour period.

TABLE I. FREQUENTLY USED SYMBOLS

Symbol	Meaning
q_i	the i -th query point.
S_q	stream of query points.
W_t	window of Size t -time.
p_i	i -th point p ; $i = 1, \dots, n$.
n_{W_t}	number of non-expired points in window W_t .
$p_i.arr$	arrival time of point p .
$p_i.exp$	expiring time of point p .
p_{anom}	point p declared as anomalous.
now	current time.
r	query range, $q.r$ is the distance parameter for query q .
k	number of neighbours parameter $q.k$ is the neighbours parameter for query q .
nn_{p_i}	number of neighbours for point p .
MC_i	i -th micro cluster, $i = 1, \dots, n$.
D	micro cluster density, $MC_i.D$ is the density of the i -th microcluster.
$x.MC_i$	all time windows microcluster i is present in.
$y.MC_i$	all Density measurements for microcluster i

This allows for the creation of a polynomial equation that represents the density of a specific cluster throughout the time windows. Using this equation, it is then possible to generate a normalized curve that represents the activity of a cluster independently of all other clusters. Naturally, this leads to the automated generation of a value that can be used to assess the activity of a cluster without the need for manually solving a value for K , as traditional methods would necessitate. This method successfully enables the ability to generate a polynomial equation to successfully represent the activity of a cluster over time as a curve. In this architecture, two curves are generated, one that is continuously updated over time, and another that only represents activity over a three hour period.

Algorithm 1 MCOD in Time Series

Input: Query Point Stream S_q

Output: MC_i for all i , P_{anom}

```

1: while ( $S_q$ ) do
2:   Obtain point  $q$  from  $S_q$ 
3:    $q.arr = now$ 
4:   if  $q.r < |MC_i|$  then
5:      $q$  is added  $MC$ 
6:      $MC.D++$ 
7:   else if  $nn_q \geq k$  then
8:     new  $MC$ 
9:      $MC++q$ 
10:  else
11:     $p = p_{anom}$ 

```

Figure 3. Processing of new arrival.

Algorithm 2 Removal of Expiring Points

Input: W_t for all t

```

1: for All  $p$  in  $W_t$  do
2:   if  $p.exp > p.arr + now$  then
3:     if  $p.exp > W_t.t$  then
4:       remove  $p$  from  $W_t$ 

```

Figure 4. Removal of Expiring Points.

Algorithm 3 Generation of Polynomial Curve**Input:** $d.MC_i$ for all i **Output:** Eq_{MC_i} for all i

```

1: for  $MC_i$  for all  $i$  do
2:    $x = x.MC$ 
3:    $y = y.MC$ 
4:    $n = x.size$ 
5:    $k = n$ 
6:   while  $k \neq \text{rank}$  do
7:     Build Vanermounde Matrix
       
$$\begin{bmatrix} 1 & x_1 & \cdots & x_1^k \\ 1 & x_2 & \cdots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{bmatrix}$$

8:    $y = Xa$ 
9:   if  $y$  is not full rank then
10:     $k = k - 1$ 
11:    break;
12:   else
13:     $X^T y = X^T X a$ 
14:     $a = (X^T X)^{-1} X^T y$ 
15:     $y = a_0 + a_1 x + \cdots + a_k x^k$ 
16:    $Eq_{MC_i} = y$ 

```

Figure 5. Generation of Polynomial Curve**V. METHODOLOGY****A. Collected Data**

To successfully test the proposed method, NetFlow was collected in a 24-hour period from within NPD. This was live data, and no previous insight about this 24-hour period was held. It was unknown if it contained anomalies or not, simulating real-world conditions. The NetFlow contained all communications, both internally and externally, during this period. It contained a total of 158,125,828 NetFlow samples. From these samples, 8 attributes were extracted (Table II). And from these, 5 attributes were selected to be used in the anomaly detection calculations.

- Destination Port
- Source Port
- Destination Bytes
- Source Bytes
- Protocol ID

These attributes were normalized using theoretical maximums as well as observed maximums over a 3-month period (Table III). The remaining collected attributes were excluded from the anomaly calculations due to various reasons. The IP addresses were excluded due to the IP address leases allocated by DHCP servers were inconsistent in both maintaining the allocated IP's, and time-out periods for leased IP's. This would lead to inconsistent results within networks, as IP addresses could be re-allocated in as little as 30 minutes, drastically changing their perceived normal traffic pattern.

TABLE II. FREQUENTLY USED SYMBOLS

Destination Bytes	Volume of traffic sent from the Destination IP in bytes
Destination I.P.	The destination I.P address of the NetFlow
Destination Port	The Destination port used for the NetFlow
Protocol I.D.	The ID of the Protocol used. (6 = TCP, 17 = UDP)
Source Bytes	Volume of traffic sent from the Source IP in bytes
Source I.P.	The Source IP of the NetFlow
Source Port	The Source port of the NetFlow
Start Time	The time at which the represented communiqué was initiated.

TABLE III. MAXIMUM VALUES USED FOR NORMALIZATION

Variable	Maximum Value
Source/Destination Port	65535
Source/Destination Bytes	4294967270 (Bytes)
Protocol ID	255

These attributes, along with the Start Time attribute, were collected for the investigation of identified anomalies and clusters within the proposed architecture.

B. Program Configurations

For testing, windows of sizes of 5,15,30,60,180 minutes were configured, with polling of cluster densities occurring after every window cycle. This lead to a total of 58 points used to generate the polynomial equation to represent a clusters density though a 3-hour time window. Each window had the same variable configurations aside from the window size. A radius value of 0.025 was used for range calculations. K, now a performance metric used to denote when a cluster is dense enough to be monitored by the regression algorithm, was set to 25.

This value, when compared to the number of samples in a 3-hour period, may seem low. It was chosen to allow for the generation of clusters representing common activity at the earliest stage possible. This allows for the tracking of the density though the regression algorithm within the smallest available time window to compare activity within the results section (Section V).

VI. RESULTS

In this section, we discuss the results of the proposed method of anomaly detection. Anomalous samples that were identified at both stages of the proposed architecture will be outlined and analysed. Examples of normal activity of various types will also be presented.

A. Point Anomalies

Due to the two-stage architecture of the proposed method, anomalies may be detected in two different manners. Firstly, anomalies that are anomalous by the distanced based calculations are outlined at the end of every time window. An anomaly outlined in this window contains a point that never meets the required density irrespective of time. These are regarded as Point Anomalies.

Initial results indicated that the volume of these point anomalies was significant within short time windows.

Therefore, results for point anomalies were not investigated until a full window of 3 hours was completed. Table IV outlines example of such anomalies that were consistently active across the 24 hour period.

TABLE IV. DETECTED POINT ANOMALIES

Sample ID	Source Port	Destination Port	Source Bytes	Destination Bytes	Protocol ID
A	53134	40001	3400	5032	6 (TCP)
B	14553	1720	256	0	6 (TCP)

Sample A, is one such sample. Upon investigation, it was discovered that this belonged to an internal NPD IP address that was communicating with a Microsoft server. This was noteworthy however as it appeared to be a non-NPD device updating through external services. While this is not necessarily a security incident, as communication seemed legitimate (port 40001 is commonly used for Windows updates), it was still of interest. Sample B contains a sample that was flagged and was shown to be internal NPD Traffic. This sample proved to be a misconfigured VoIP system. This sample, again, was not of security concern, however still of interest to security personnel.

While both incidents were proved to be non-malicious, both are still very interesting finds. Either one of these incidents are typically correlated with a “rouge asset” incident. This is an asset connected to the network that has not been approved or cleared for use by the security operations team. If incidents such as these occurred on a sensitive subnet, it would be considered a security issue to be dealt with immediately. Both discoveries would not have been detected through normal monitoring and analysis tools.

B. Point Anomalies

Due to the low value of K selected for experimentation, a total of 422 micro clusters were formed throughout the 24 hours of testing. Each micro cluster was shown to be of a highly independent nature, with cluster density proving to have unique activity when compared against each other (Table VI).

Clusters were observed throughout the 24-hour period, with generation of polynomial representations every 3 hours. In parallel, a separate curve was also generated after the first 3-hour period that the cluster was first observed. This equation was constantly updated with new data to represent a regression that generalizes the activity of a cluster over a longer period. Comparison of both curves took place through the use of the Fréchet distance.

TABLE V. POLYNOMIAL REPRESENTATIONS OF CLUSTER DENSITIES

Reference ID	Polynomial Expression
1	$1413.7x^4 - 12063x^3 + 37870x^2 - 43326x + 19390$
2	$1848.7x^4 - 17661x^3 + 61331x^2 - 80081x + 38318$
3	$169.83x^4 - 1661.2x^3 + 5887.7x^2 - 8119.3x + 3963$
4	$-1.4583x^4 + 19.583x^3 - 78.542x^2 + 120.42x - 60$

By calculating the Fréchet distance between the two curves, it is possible to measure the deviance in behaviour from the normalized polynomial to the observed polynomial. Table V outlines clusters that were detected and tracked throughout the 24-hour period. These examples were chosen out of the 422 micro clusters generated with no preference. These clusters represent activity throughout NPD’s network. Again, it is important to note that this test was conducted on live data with no previous assumptions.

Table VI presents cluster locations and densities throughout a 3-hour window. Along with this, the Fréchet distance was calculated and normalized using the maximum cluster density observed with a 3-hour window over the 24-hour period. It is apparent from this window alone that the density values are indeed very independent of each other. From these values, a polynomial curve was generated and compared to the normalized curve using Fréchet distance as a measure of dissimilarity. At this stage, the normalized curve was generated over a period of 12 hours. Table VII presents the results as calculated.

Firstly, apparent is the absolute match of polynomials in Sample 4. For the analysis of results, a focus on maximum divergence was taken initially, however the total lack of divergence, when compared to the rest of the monitored clusters, was anomalous.

Upon investigation, it was found that a single asset was communicating along ports 7771 to a multicast IP address (224.5.6.7). While not malicious, requests to this IP address were extremely consistent across all time windows, showing a divergence of 0 constantly. It was found that an internal asset focused on development was trying to communicate with the IP address. The IP address, interestingly, is commonly found among multicast programming examples. Upon discovery, this activity was stopped immediately.

This example was of interest to the NPD network security team. A data leak would show very similar characteristics to sample 4, most notably, a “low and slow” type attack, that focuses on exfiltration of small amounts of data over extremely long periods of time. The ability for the proposed architecture to detect this kind of behaviour is of notable importance, as methods such as MCODE would have been unable to successfully detect this traffic through the normal tuning operation of the variable K.

Throughout testing, it was noted how consistent some of the network activity appears to be when clustering using the MCODE technique. Samples 11 to 14 show this by regressing into quadratic equations, with comparison metrics showing a non-significant amount of deviance between time windows.

Sample 15 displayed the greatest deviation among the clusters. Upon investigation, it was revealed that an internal asset connected to an external asset for a long period. This connection was the result of an employee contacting support for a piece of software, but simply left the application active on their machine. While this instance was not malicious, connecting and maintaining a connection to an external service for lengthy periods of time may be indicative of a possible attack

TABLE VI. TRAFFIC ACTIVITY EXAMPLES

Reference ID	Source Port	Destination Port	Source Bytes	Destination Bytes	Protocol ID	5*	15*	30*	60*	180
1	56582	135	2120	2320	6	3285	10336	19060	41903	125240
2	54132	135	12814	9268	6	3755	11767	22937	42216	118923
3	58329	443	1118	3698	6	240	703	1499	2851	9058
4	7771	7771	119584	0	17	N/A	N/A	5	45	115
5	1720	61444	320	0	6	1418	4018	7497	13860	43069
6	60408	7002	1443	0	6	250	751	1554	3113	8823
7	63080	7800	560	0	6	250	744	1408	2985	9541
8	0	0	2016	936	50	133	511	1119	2281	7021
9	55081	1521	4844	3628	6	48	165	318	634	1826
10	4354	443	1217	2191	6	1087	2766	5490	12003	34813
11	6249	80	3496	11096	6	29	89	195	356	1111
12	49507	5550	15368	55596	6	100	275	567	1126	3450
13	64176	2000	8096	244440	6	38	144	303	602	1740
14	10051	53505	438	0	6	39	147	298	648	1933
15	6979	5081	74	136	6	N/A	21	45	90	274

TABLE VII. CURVE COMPARISON OF SAMPLES

Reference ID	Normalized Polynomial	Observed Polynomial	Fréchet Distance
1	$-0.0128x^3 + 3.1385x^2 + 540.61x + 840.1$	$-0.0056x^3 + 0.9623x^2 + 710.06x + 117.59$	0.064685
2	$0.0126x^3 - 3.429x^2 + 873.36x - 551.9$	$0.0188x^3 - 4.2568x^2 + 861.22x - 386.93$	0.0838168
3	$0.0006x^3 - 0.1343x^2 + 54.09x - 43.974$	$0.0002x^3 + 0.0311x^2 + 48.587x - 12.715$	0.0969755
4	$-1.4583x^4 + 19.583x^3 - 78.542x^2 + 120.42x - 60$	$-1.4583x^4 + 19.583x^3 - 78.542x^2 + 120.42x - 60$	0
5	$0.0046x^3 - 1.0187x^2 + 274.16x + 87.072$	$-0.005x^3 + 1.38x^2 + 234.55x + 209.67$	0.0920324
6	$0.0002x^3 + 0.0232x^2 + 51.471x - 13.504$	$-0.0008x^3 + 0.2053x^2 + 41.025x + 56.416$	0.0925364
7	$-0.0004x^3 + 0.135x^2 + 42.298x + 49.549$	$-0.001x^3 + 0.2703x^2 + 45.385x + 13.104$	0.0812508
8	$0.0017x^2 + 39.06x - 65.127$	$0.0005x^3 - 0.1461x^2 + 41.377x - 72.919$	0.080559
9	$-0.0038x^2 + 10.847x - 2.4865$	$0.0039x^2 + 10.787x - 2.5642$	0.0838819
10	$-0.0051x^3 + 1.2061x^2 + 139.71x + 377.08$	$0.0027x^3 - 0.6516x^2 + 196.23x + 70.613$	0.0621606
11	$0.0013x^2 + 5.9133x + 3.1589$	$-0.0019x^2 + 6.6434x - 8.2158$	0.0835281
12	$0.0036x^2 + 18.5x + 4.0058$	$0.0069x^2 + 17.19x + 16.59$	0.0853373
13	$-0.0046x^2 + 10.559x - 13.175$	$0.0027x^2 + 10.663x - 15.135$	0.0840693
14	$-0.0015x^2 + 11.129x - 21.378$	$0.0033x^2 + 8.795x + 6.8986$	0.087594
15	$0.0008x^3 - 0.0175x^2 + 2.48x + 8.9333$	$-0.0015x^2 + 1.73x - 7.0056$	0.4041794

VII. CONCLUSIONS AND FUTURE WORK

This paper proposed an architecture designed to detect anomalies within NetFlow data. To achieve this, presented a method in which to eliminate the dependency a key variable used in previous anomaly detection methods. By utilizing polynomial regression techniques on cluster densities, it is possible to detect abnormal behavior on the network. In doing so, the systems remove the need for tuning the algorithm to a specific set and automates this process. By conducting this process, clusters of activity can now be monitored independently of each other.

This paper utilized real world data that was taken from The NPD Group's network over a period of 24 hours. Analysis of this data revealed anomalies within the network that were otherwise unnoticed by other tools monitoring the network, thus proving its ability to detect previously unknown anomalies.

To fully test the algorithms proficiency, future work will be focused on deploying the algorithm into a real-time environment. Research into the efficiency of the algorithm over longer periods of time will also be analyzed. While no malicious activity was detected, the architecture has shown promise in its ability to detect anomalous behavior after only 6 hours of training time. Longer periods of training may lead to

a comprehensive map of normal activity provided by the generation and storage of the normalized polynomial curves.

Also, the use of the regression system also provides an ability to retroactively generate curves representing anomalous behavior only. Events discovered after the fact can be represented by a series of cluster changes. Investigation into this possibility will also take place in the hopes of enabling the ability to activity monitor the network for specific events identifiable by a change in cluster densities.

REFERENCES

- [1] Z. Chen, C. K. Yeo, B. S. L. Francis, and C. T. Lau, "Combining MIC feature selection and feature-based MSPCA for network traffic anomaly detection," in *2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC)*, 2016, pp. 176–181.
- [2] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowl.-Based Syst.*, vol. 78, pp. 13–21, Apr. 2015.
- [3] G. Fernandes Jr., L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proença Jr., "Network anomaly detection using IP flows with Principal Component Analysis and Ant Colony Optimization," *J. Netw. Comput. Appl.*, vol. 64, pp. 1–11, Apr. 2016.
- [4] R. Ciplinskas and N. Paulauskas, "Outlier Detection Method Use for the Network Flow Anomaly Detection," *Moksl. - Liet. Ateitis*, vol. 8, no. 3, pp. 327–333, Jun. 2016.

- [5] R. Wankhede and V. Chole, "Intrusion Detection System using Classification Technique," *Int. J. Comput. Appl.*, vol. 139, no. 11, pp. 25–28, Apr. 2016.
- [6] M. Goldstein, "FastLOF: An Expectation-Maximization based Local Outlier detection algorithm," in *2012 21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 2282–2285.
- [7] M. Kontaki, A. Gounaris, A. N. Papadopoulos, K. Tsiachas, and Y. Manolopoulos, "Continuous monitoring of distance-based outliers over data streams," in *2011 IEEE 27th International Conference on Data Engineering*, 2011, pp. 135–146.
- [8] I. W. O. K. Putra, Y. Purwanto, and F. Y. Suratman, "Modified K-means algorithm using timestamp initialization in sliding window to detect anomaly traffic," in *2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2015, pp. 19–23.
- [9] K. Goeschel, "Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis," in *SoutheastCon 2016*, 2016, pp. 1–6.
- [10] J. Esmaily, R. Moradinezhad, and J. Ghasemi, "Intrusion detection system based on Multi-Layer Perceptron Neural Networks and Decision Tree," in *2015 7th Conference on Information and Knowledge Technology (IKT)*, 2015, pp. 1–5.
- [11] V. Vijayakumar, V. Neelanarayanan, E. V. Balan, M. K. Priyan, C. Gokulnath, and G. U. Devi, "Big Data, Cloud and Computing Challenges Fuzzy Based Intrusion Detection Systems in MANET," *Procedia Comput. Sci.*, vol. 50, pp. 109–114, Jan. 2015.
- [12] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests," *Inf. Sci.*, vol. 278, pp. 488–497, Sep. 2014.
- [13] V. Carela-Español, P. Barlet-Ros, A. Cabellos-Aparicio, and J. Solé-Pareta, "Analysis of the impact of sampling on NetFlow traffic classification," *Comput. Netw.*, vol. 55, no. 5, pp. 1083–1099, Apr. 2011.
- [14] R. Kumari, Sheetanshu, M. K. Singh, R. Jha, and N. K. Singh, "Anomaly detection in network traffic using K-mean clustering," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 387–393.
- [15] A. Alsayat and H. El-Sayed, "Social media analysis using optimized K-Means clustering," in *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2016, pp. 61–66.
- [16] D. T. Velmurugan, "Efficiency of k-Means and K-Medoids Algorithms for Clustering Arbitrary Data Points," *ResearchGate*, vol. 3, no. 5, pp. 1758–1764, Oct. 2012.
- [17] J. Asmuss and G. Lauks, "Network traffic classification for anomaly detection fuzzy clustering based approach," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 313–318.
- [18] A. Abid, A. Kachouri, and A. Mahfoudhi, "Anomaly detection through outlier and neighborhood data in Wireless Sensor Networks," in *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2016, pp. 26–30.
- [19] P. Fu and X. Hu, "Biased-sampling of density-based local outlier detection algorithm," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2016, pp. 1246–1253.
- [20] Z. Tsiatsikas, A. Fakis, D. Papamartzivanos, D. Geneiatakis, G. Kambourakis, and C. Kolias, "Battling against DDoS in SIP: Is Machine Learning-based detection an effective weapon?," in *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, 2015, vol. 4, pp. 301–308.
- [21] B. Gajic, S. Nováczki, and S. Mwanje, "An improved anomaly detection in mobile networks by using incremental time-aware clustering," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 1286–1291.
- [22] P. C. Wong et al., "A visual analytics paradigm enabling trillion-edge graph exploration," in *2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*, 2015, pp. 57–64.
- [23] B. Li, J. Springer, G. Bebis, and M. Hadi Gunes, "A survey of network flow applications," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 567–581, Mar. 2013.

Kieran Flanagan received his BSc degree (with honors) in Software Design in 2015. After this, he joined The NPD Group while studying as a postgraduate in Athlone Institute of Technology. He is currently perusing his MSc. His research activities including the creation of abstracted vulnerability metrics for network based assets and the detection of possible malicious network activity.

Dr. Enda Fallon joined Athlone Institute of Technology (AIT) from Ericsson in 2002. In 2003 he founded AIT's Software Research Institute (SRI). Since 2003, Enda has been a principal investigator on over 30 collaborative industry/academic research projects. His research interest focuses on service mediation and adaptation for heterogeneous networking environments. Enda holds a BSc in Computer Science and Mathematics from University College Galway and an MSc in Software Engineering from Athlone Institute of Technology and a PhD (Computer Science) from the Performance Engineering Laboratory at University College Dublin.

Dr. Abir Awad received her PhD degree in Computer Science from Polytech' Nantes, University of Nantes (Nantes, France). Her PhD was within the framework of the research project "ACSCOM", "Chaos based security for mobile communicating systems" supported by the ANRT. She developed her doctoral thesis on Cryptography from September 2006 to November 2009. Later, she joined the Operational Cryptology and Virology Laboratory ((C+V)^o), ESIEA and Le Mans University in Laval in 2010 and the computer science lab. (LIFO), University of Orleans in Orleans in 2011 as a lecturer-researcher. In 2012, she worked at the computer science dep, Ryerson University, Toronto, Canada. She is currently working as a researcher at the Software Research Institute (SRI), Athlone Institute or Technology (AIT) within the Irish Centre of Cloud Computing and Commerce IC4).

Paul Connolly started working in the NPD Group in 2014 as a Senior Security Analyst to evaluate and test security applications within the company. His current research focus is on the development of novel risk evaluation and visualization approaches.