Big Data and Deep Models Applied to Cyber Security Data Analysis

Ying Zhao
Naval Postgraduate School
yzhao@nps.edu

Andrew Polk
UC Santa Barbara
polk@umail.ucsb.edu

Shaun Kallis
Cal State University Monterey Bay
shaunlantzkallis@gmail.com

Lauren Jones
Naval Postgraduate School
lmjones@nps.edu

Riqui Schwamm
Naval Postgraduate School
rschwamm@nps.edu

Tony Kendall
Naval Postgraduate School
wakendal@nps.edu

Abstract

We present initial work that applies big data and deep models to a cyber security data analysis with a use case approach. We explored new technologies such as BDP (Big Data Platform) as a service on the Amazon AWS system and Lexical Link Analysis (LLA). BDP provides various analytics in near real-time to help decision makers respond to threats and in a timely manner. We also used LLA as an example of deep models and a data-driven unsupervised ML method that can improve cyber decision making.

DoD networks require strong Cyber Situational Awareness Analytic Capabilities (CSAAC) because adversaries deploy increasingly sophisticated malicious activities against DoD networks and therefore requires the capture and inspection of packets transmitted within the network to assess the cyber security questions of who, what, where and when.

New big data analytical tools and technologies can dramatically improve CSAAC by effectively and efficiently aggregating the ever-increasing volume of data from disparate sources that could provide early detection of network vulnerabilities, threats, and attacks.Big data and deep models could provide significant opportunities to perform better analysis of real-time data and potentially:

- Prevent expensive and damaging distributed denial of service (DDOS) attacks
- Maintain a competitive advantage of the military or businesses by protecting expensive research
- Prevent blackmail from email or ransomware
- Better secure vital networked infrastructure

Data Set Description

The cyber data was taken from multiple routers in the Los Alamos National Laboratorys internal network (LANL 2017). The data set contains windows authentication events and processes, domain name lookups, network flow data, and hacking events. The data contains 58 days and total 12 gigabytes of network information and 1.6 billion events. There were known malicious activities (identified as Red Team Actions) conducted within this network during this time period.

Some of the information contained within the dataset was anonymized or deidentified. While this removes significant amounts of information from the data set, there is still valuable information to be gleaned about the behavior of the network due to unity of identification across the five different files (i.e. User 1 or U1 is the same user across all data sets and Computer 1 or C1 is the same computer across all data sets).

Some of the well-known ports (e.g. http port 80, 443, etc.), protocols (e.g. 6 for Transmission Control Protocol), and system users (e.g. SYSTEM or Local Service) were left identified within the datasets. Time was captured in one-second intervals, starting with a time epoch of (1). In order to illustrate the methodologies studied in this paper, we started with the Domain Name Service (DNS) data set. Figure 1 shows a snapshot of the LANL-DNS data. Time, source computer, and computer resolved are the attributes.

The LANL cyber data set was chosen for a number of different reasons over other popular open source data sets (e.g., DARPA (DARPA 2000) or KDD data (KDD 1999) sources). The LANL cyber data set is from 2015, one of the more recent data sets of this size and complexity, so it contains the activities of some newer malicious attack methodologies. The goal is to classify and predict the hacked or hacking computers using big data and deep models.

Methods

In order to incrementally test cyber data sets using potential big data and deep models including ML/AI methods, the LANL-DNS data file was initially pre-processed, analyzed, and interpreted to understand the output results shown in this paper before testing on other more complex data sets. The steps for understanding the data:

```
31,C161,C2109
35,C5642,C528
38,C3380,C22841
```

Figure 1: The LANL-DNS log data[2]

- Perform data visualization and exploration: display and visualize data initially and check data quality.
- Perform unsupervised machine learning to discover interesting patterns and anomalies.
- Apply supervised learning to generate more precise classification or prediction models.

Data Visualization and Exploration Using Big Data Platforms (BDP)

Defense Information Systems Agency (DISA) 's BDP is on Amazon Web Services (AWS) and a mix of big data standard tools and customization including tools for ingestion, data management, security, data exploration, and data analysis. These functions are supported by open source tools including PostgreSQL, Apache Maven, Apache Spark, Apache Storm (Kronos), Elastic Search, GEM prospector, Hadoop, Map/Reduce, Kafka, Accumulo, Unity, IronHide (Kibana), Zookeeper, Kryolibrary, NodeJS, R-Shiny.

BDP can process large-scale real time data feeds to provide useful visualizations of the data for initial data exploration to discover anomalous events. Ingestion of the LANL-DNS data into the BDP cluster included the following steps:

- Customized and formatted a rapid deployment archive (RDA) for parsing the csv file data
- Connected a puppet server to upload data to the Kronos server which ingested and parsed the data

For the data visualization and exploration, we used Unity and Kibana/Iron Hide. Unity uses queries to visualize time series, histograms, and pie charts for the initial examinations of the data. Iron Hide creates Data-driven documents (D3) visualizations including heat maps, graphs, and charts which could indicate threats. Figure 2 shows the Unity histogram of the event counts (i.e., each line in the LANL-DNS data is an event associated with a timestamp) for all the computers. Figure 3 shows a Kibana heat map of number of connections made for each computer (y-axis) over time (x-axis). These tools could show big data in a near real-time to provide rapid updates for a focused segment.

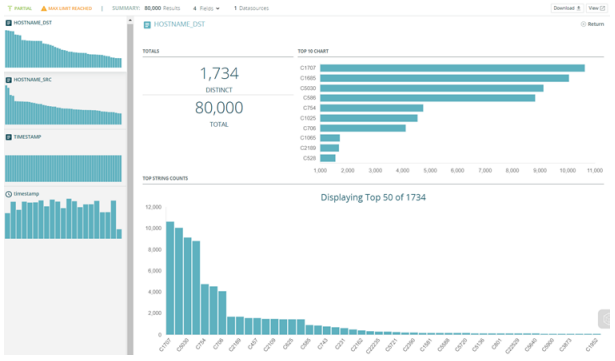

Figure 2: BDP Unity histogram of the event counts (i.e., each line in the LANL-DNS data is an event associated with a timestamp)

Visualization/Exploration Using Gephi and Plotly For the data exploration, we also used a open source network display program Gephi (Gephi 2018) as a way to visualize the LANL cyber data that shows the connections between points in data sets. Gephi uses Source and Target fields to draw the network graphs. Gephi also includes a timeline function to allow a user to view the connections between nodes at specific times or in a range of times.

The LANL flow data was displayed with Gephi. Since the red team created hacking events such as teal colored computer nodes in Figure 5, the hacking or hacked computer nodes resulted from the red teams actions. Each node is a computer. Figure 4 shows the hacking events during a 24-hour period. One teal node is hacking, the orange nodes are being hacked, purple nodes are neither hacking nor being hacked. The color of the edges between nodes represents the protocols used for the connections. Purple edges are most likely TCP. Green connections are protocol-1 which may be related to the hacked computers.

The shape of the graph provides clues as to the nature of the nodes. Nodes that are highly connected to other nodes may be name servers or popular web servers. The hacked nodes seem in the area of the nodes with higher numbers of

Figure 3: BDP data exploration: A heat map showing the number of connections made for each computer over time from Kibana

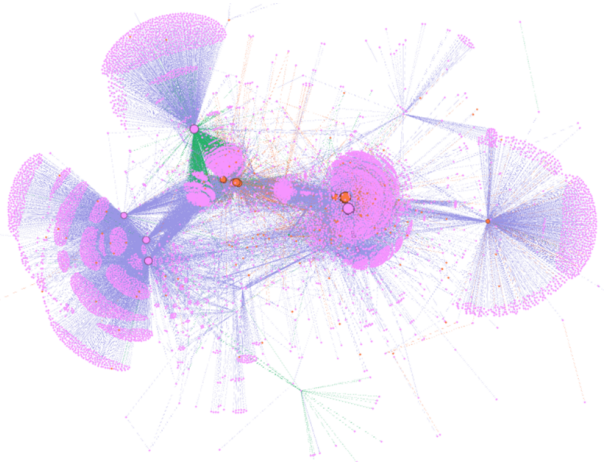

Figure 4: Gephi network visualization of computers

connections (high centralities).

We also explored the Sankey graph with Python Plotly (Sankey 2018). Figure 5 shows a Sankey graph to categorize how different parameters such as protocols, port numbers, and packets connected to each other in the LANL flow data. For example, protocol-6 is mostly associated with port ranges 1025-65536 and then port ranges 0-1024.

Unsupervised Learning Using Lexical Link Analysis (LLA)

In a LLA (Zhao, MacKinnon, and Gallup 2015), describes the characteristics of a complex system using a list of attributes or features with specific vocabularies or lexical terms. Because number of lexical terms can be potentially very large from big data, the model can be viewed as a deep model for big data. For example, we can describe a system using word pairs or bi-grams as lexical terms extracted from text data. LLA automatically discovers word pairs, and displays them as word pair networks. Bi-grams allow LLA to be extended to numerical or categorical data. For example, for structured data such as attributes from databases, we discretize and then categorize attributes and their values to word-like features. The word pair model can further be extended to a context-concept-cluster model (Zhao and Zhou 2014). A context can represent a location, a time point or an object (e.g. file name) shared across data sources. For example, in information assurance, information is the context, assurance is the concept. The timestamp, computer name are the contexts to link different data sources.

Figure 6 shows an example of such a word network discovered from text data. Clean energy, renewable energy are two bi-gram word pairs. For a text document, words are represented as nodes and word pairs as the links between nodes. A word center (e.g., energy in Figure 6) is formed around a word node connected with a list of other words to form more word pairs with the center word energy.

We computed associations and links as pairs of a source computer and a resolve computer from the LANL-DNS data set. The strength of the associations and links are defined as how many time points or events that the two computers are linked via "source" or "resolve".

The output from LLA for the LANL-DNS data processing identified 15237 unique active devices (computers). There are no identifying features differentiating an end user device such as a personal computer versus a DNS server; all are identified as anonymous devices, such as C123. Figure 7 shows an example of a LLA network discovered from the LANL-DNS data. Each node is a computer. The links represent how likely two computers are linked as a "source" and "resolve" pair in the events (timestamps).A correlation measure is computed using Equation (1). Colored nodes (computers) are grouped into one clusters based on their link patterns using LLA.

$$r_{ij} = \frac{(Linked\ Events\ Computer\ i\ and\ Computer\ j)}{\sqrt{(Events\ Computer\ i)(Events\ Computer\ j)}} \quad (1)$$

One can filter the nodes based on the strength of the links in LLA as shown in Figure 8.

The detail LLA outputs for the LANL-DNS data set are listed as follows:

Output 1: The list of words representing the computers in the data set and nodes in the network with the following characteristics computed as shown in Figure 2.

- Group: what group a node belongs. A node or a word is a computer.
- Type: group type from LLA.
- Degree: how many connections each node has.
- Betweenness: how many connections belong to the different groups.
- Degree in: how many connections a computer (word) as resolve.

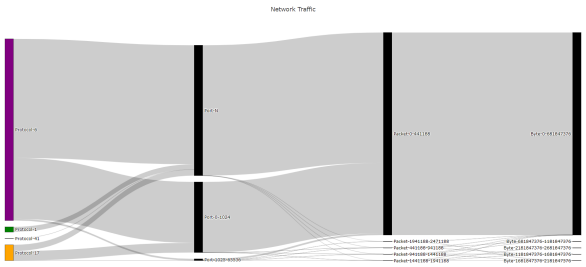

Figure 5: Sankey for showing the LANL flow data. Number of packets and bytes are split into five groups each with proportional ranges of one of the 5th of their maximum value

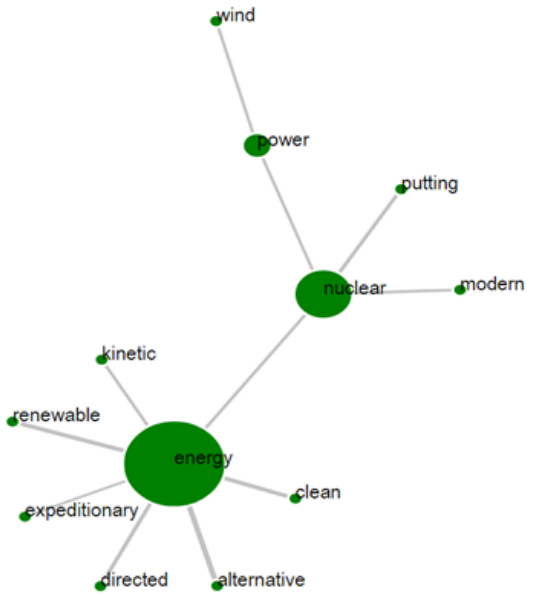

Figure 6: An example of word network from a text data by LLA

- Degree out: how many connections a computer (word) as source.

Output 2: The list of associations of computer associations.

After the initial data exploration, the question of the research is that how to predict hacking and hacked computers from these data sets. We computed additional metrics based on the Output 1 of LLA as follows:

- Multi: degree_in*degree_out;
- DIV: degree_in/degree_out if degree_out not 0;else 0;
- SUM: degree_in+degree_out;
- DIFF: degree_in-degree_out

Figure 11 show a gains chart for predicting the hacked and hacking computers. The x-axis shows the computer sequence number ranked by the four metrics. The y-axis shows percentage of hacked or hacking computer nodes. 1.75% out of 15237 total computers are either hacked or hacking as the ground truth, therefore, if there is a perfect prediction algorithm, the top 1.75% of the sorted nodes (based on the perfect scores) should predict 100% of the hacked or hacking computers as shown in the leftmost curve (two straight lines). The two results are interesting:

- The best performed prediction metric is Multi (degree_in*degree_out) where the top 2160 nodes (14%) include 62% of the total hacked or hacking nodes. This is the best gain over other scores: For example, if sorted by the degree in scores, the top 14% contains 56% of the total hacked or hacking nodes. If sorted by the random scores, 14% contains 14% of the total hacked or hacking nodes, which is the worst performing prediction.
- The bottom ranked 40% of the nodes (from 9112 to 15237) are normal. This is also significant since we can eliminate the 40% nodes when examining hacked or hacking nodes, which is a big labor saving for cyber security analysts.

The metric "degree_in*degree_out" indicates highly active devices are more likely to be hacked. The highly active devices do not mean they are anomalous, however, a common behavior seen in malicious actions is increased activity of devices that may be participating involved in the unauthorized action. We later computed an activity metric by counting the number of event (i.e. timestamps) a computer is associated in the data set. This is a much simpler metric to compute than the associations in LLA. The activity metric shows a similar gain to the best LLA metric. We also appended other node characteristics of in the flows data such as the number of source ports, number of destination ports, total duration of a nodes connections, total packets of a nodes connections, total bytes of a nodes connections as shown in Figure 12, and then apply supervised machine learning methods using the tool (Hall et al. 2009), in an at-

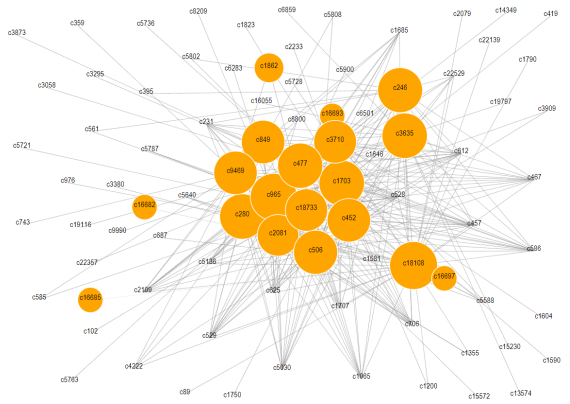

Figure 7: An example of feature network from the LANL-DNS data by LLA

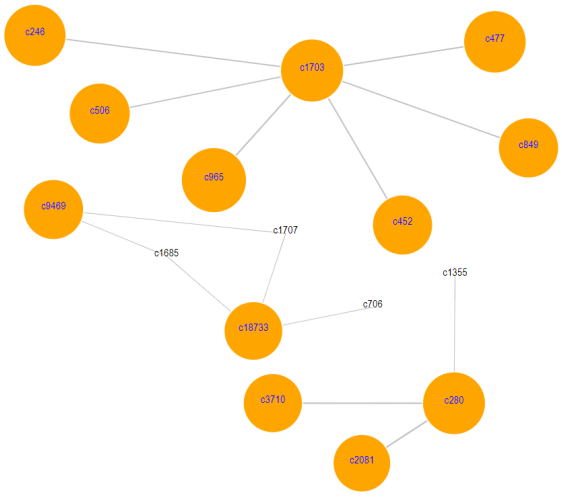

Figure 8: The links of computer nodes filtered from Figure 7

word	group	type(color)	degree(size)	betweenness(size)	degree_in	degree_out	metric1
c1685	5	Popularity	14049	10061	14033	16	14017
c1707	5	Popularity	13952	9975	13931	21	13910
c586	21	Emerging	13545	10387	13456	89	13367
c457	21	Emerging	13347	10285	13319	28	13291
c1065	21	Emerging	13423	10342	13342	81	13261
c529	21	Emerging	13421	10350	13339	82	13257
c467	21	Emerging	13393	10332	13310	83	13227
c612	21	Emerging	13392	10307	13302	90	13212
c528	21	Emerging	13323	10247	13253	70	13183
c2109	21	Emerging	13386	10301	13274	112	13162
c625	21	Emerging	13419	10318	13272	147	13125
c706	23	Popularity	12810	8118	12806	4	12802
c1025	23	Popularity	12205	7747	12197	8	12189
c231	10	Emerging	12589	11213	12337	252	12085
c5030	23	Popularity	12108	7690	12085	23	12062
c754	23	Popularity	13345	8534	12252	1093	11159
c5136	23	Popularity	11207	7007	11143	64	11079
c22529	23	Popularity	10618	6614	10554	64	10490
c5640	23	Popularity	10482	6590	10442	40	10402
c5588	23	Popularity	10466	6571	10423	43	10380
c4222	23	Popularity	10464	6566	10403	61	10342
c561	9	Anomaly	13233	13145	10720	2513	8207
c585	10	Emerging	7668	6219	7264	404	6860
c743	10	Emerging	7624	6190	7202	422	6780
c2174	9	Anomaly	6354	6180	6185	169	6016
c395	9	Anomaly	6353	6279	5888	465	5423
c18268	10	Emerging	5395	4388	5378	17	5361
c5721	10	Emerging	5272	4187	4917	355	4562
c16135	10	Emerging	4405	3487	4384	21	4363
c5720	10	Emerging	4205	3218	3942	263	3679
c3349	10	Emerging	3334	2904	3251	83	3168

Figure 9: LLA outputs of the node characteristics https://v2.overleaf.com/project/5b9ae8266dbe242220b55f42

tempt to generate better gains charts. So far, the metric "degree_in*degree_out" from unsupervised LLA shows a slight edge over other methods.

Conclusion

We applied big data and deep analytics methods to the LANL cyber data used to detect the hacked or hacking computers in a network where DNS, flows, services and login information are collected. We showed how big data visualization and exploration tools such as BDP, Gephi, and Python Plotly can explore ig data to provide meaningful information to decision makers. Gephi and Plotly are good for prototyping. The BDP has security advantages and shows potential for finding anomalies in near real time through various metrics. LLA computes the associations, statistics and centralities for nodes (computers) and derived metrics are significantly useful to predict hacked or hacking nodes in the gains chart evidently. The best performing metric shows the top 14% of the nodes include 62% hacked or hacking nodes and the bottom 40% of the nodes are 100% normal, therefore can be eliminated from examination.

word	degree_in	degree_out	Hack	MULT	DIV	SUM	DIF
c477	2	40	Hacked	80	0.05	42	-38
c849	3	29	Hacked	87	0.103448276	32	-26
c506	3	34	Hacked	102	0.088235294	37	-31
c452	4	29	Hacked	116	0.137931034	33	-25
c246	4	30	Hacked	120	0.133333333	34	-26
c1096	4	33	Hacked	132	0.121212121	37	-29
c1944	9	25	Hacked	225	0.36	34	-16
c965	14	38	Hacked	532	0.368421053	52	-24
c1936	21	32	Hacked	672	0.65625	53	-11
c2079	29	27	Hacked	783	1.074074074	56	2
c3635	44	22	Hacked	968	2	66	22
c16693	1	0	Normal	0	0	1	1
c16697	2	0	Normal	0	0	2	2
c18108	0	52	Normal	0	0	52	-52
c1862	5	0	Normal	0	0	5	5
c9455	10	0	Normal	0	0	10	10
c16682	2	6	Normal	12	0.333333333	8	-4
c9469	1	41	Normal	41	0.024390244	42	-40
c5700	12	8	Normal	96	1.5	20	4
c2081	4	28	Normal	112	0.142857143	32	-24

Figure 10: Derived metrics from the output of LLA

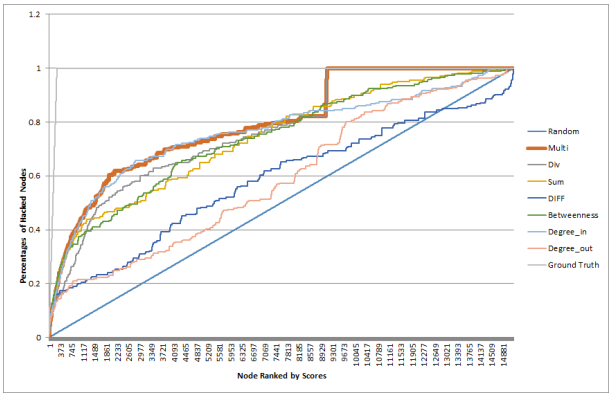

Figure 11: Gains chart using centrality node scores computed from LLA and derived metrics

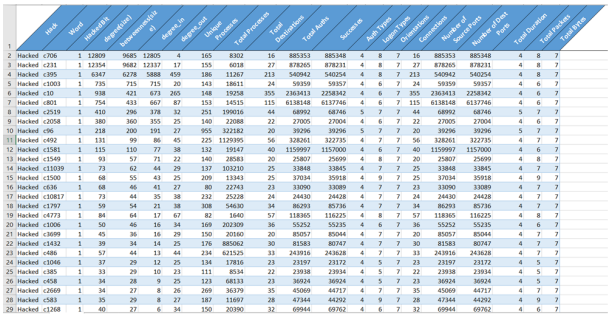

Figure 12: Combined data for computer nodes

Acknowledgements

Authors would like to thank the Naval Research Program at the Naval Postgraduate School and the Naval Research Enterprise Internship Program at the Office of Naval Research for the research support. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the U.S. Government.

References

DARPA. 2000. DARPA intrusion detection scenario specific data sets, retrieved from https://www.ll.mit.edu/r-d/datasets.

Gephi. 2018. The open graph viz platform, retrieved from https://gephi.org/.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1):10–18.

KDD. 1999. The UCI KDD archive, information and computer science, university of california, irvin,retrieved from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

LANL. 2017. The LANL cyber data, retrieved from https://csr.lanl.gov/data/cyber1/.

Sankey. 2018. Sankey graph with python plotly, retrieved from https://plot.ly/python/sankey-diagram/.

Zhao, Y., and Zhou, C. 2014. US patent 8,903,756: System and method for knowledge pattern search from networked agents. retrieved from https://www.google.com/patents/us8903756.

Zhao, Y.; MacKinnon, D.; and Gallup, S. 2015. Big data and deep learning for understanding dod data. In *Journal of Defense Software Engineering, Special Issue: Data Mining and Metrics*.