CAP6673 – Data Mining and Machine Learning

Review of Network Traffic Prediction Models for Near- and Long-Term Predictions

Written by:

Christopher Foley
Z15092976

Academic Year: 2015-2016

Summary

For network managers and operators a key problem is the prediction and detection of network intrusion from a potentially malicious source. To investigate this using the techniques of Machine Learning, artificial datasets have been created. However, due to their artificiality it is difficult to use them for realistic testing and predictions of real world data. To resolve this issue Song et al. [SONG] created a honeypot network at Kyoto University where data on network access and utilization was collected between November 01, 2006 through August 2009, where they published their results. The network consisted of different types of servers, operating systems and devices (Television and two printers). Based upon information on their web site (http://www.takakura.com/Kyoto_data/) the data collection continued through December 31, 2015.

In 2013, a subset of the data from this network was analyzed by Wald et al. [WALD] where they attempted to discern if past performance can indicate future results. The focus of this paper will be upon the work of Wald and its implications on the works of others.

In his research Wald investigated the possibility of using traffic statistics from one day (2008-01-01) and used this to train models which were then validated against datasets selected in July-December 2008 and August 2009. One of the goals was to create an algorithm that can monitor the network real-time and detect intrusions. To that goal, the feature set was reduced from 19 features provided in the Kyoto dataset, by eliminating those features which directly predicted the class label.

Surprisingly the training data proved to be very effective on the what they would call the near and long term sets. Their paper states that 4 learning algorithms were used: 5-Nearest neighbor, Naive Bayes and two forms of C4.5 Decision trees. The 5-Nearest Neighbor and both forms of the C.45 performed well on short and long term data.

Kyoto Dataset

Existing Datasets

Key to the investigation is the Kyoto dataset. Software to detect network intrusion is critical to the protection of the network. In 1998 MIT Labs created a data set for DARPA, this data set was based upon data created from a simulated USAF LAN which was attacked by the lab. A subset of this data set was then used in The Third International Knowledge Discovery and Data Mining Tools Competition which was held during KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining. The problem definition of the competition was to build an intrusion

detector and a predictive model which could determine which connections were valid and which were malicious. This dataset became known as the KDDCUP-99 dataset after the contest. While this is data set is useful, it clearly is artificial and is targeted for a military network.

Another dataset referred to by many authors is the ISCX 2012 which was developed for the same purposes as the other Network Intrusion datasets. Their paper describing this “Toward developing a systematic approach to generate benchmark datasets for intrusion detection” [SHIR] notes that their motivation was the same as others, create datasets that emulated network traffic for intrusion detection. Unlike the KDDCUP-99 data base they chose to create multiple datasets which reflected what they determined to be normal traffic in a network, then simulate attacks of different types. Thus their data could be used to distinguish and test different attack vectors. Again, this relied on a static styled network with simulated attacks.

Kyoto Dataset

A better approach was taken by Song, et al. [SONG] who created a honeypot data set designed to emulate a non military environment and monitor external real world connections and connection attempts. Unlike other datasets which created traffic in a laboratory and then artificially generated attacks the Kyoto team created an artificial network of servers with varied operating systems and security settings, peripherals (televisions and printers) and generated “normal traffic” in the hopes that the network would look attractive to an intruder. The network was then monitored for intrusions. When an intrusion was detected, the network was erased and restored to its original state. Unique to this dataset, other than the use of attacks from the real world, was the duration of their data collection. In their paper they announced the release of an evaluation database with nearly three years of actual data. Of note is that about 50% of the attacks were from China, United States and South Korea. Since the network had no real purpose all connections to the network were treated as potential threats. For each session they recorded attributes similar to the KDDCUP and additional attributes which cataloged security analysis and the IP/PORT for each session. In the three years they recorded 93,076,270 sessions of which 42,617,536 were considered known attacks and they recorded 4,420,971 source IP addresses. It is believed that many of the non attack sessions were by network routers attempting to build internal routing maps which would facilitate packet switching.

Issues

As stated before the objective is to use the attributes of network traffic to determine if a request is safe or malicious. A secondary concern is to train the software to reassess as data is collected. Looking at the Kyoto dataset it can be easily seen that a black/white list approach to classifying would be time consuming and counter productive. A classification algorithm or formula must be efficient with a minimum number of features to extract from the connection request, which would

allow on the fly checking. Due to the nature of networks and their traffic the number of characteristics of each call make it time consuming and complicated to detect on the fly and in a reasonable time such that end to end timeouts will not occur.

The approach taken in the Wald paper [WALD] was to take traffic already categorized as normal or anomalous, train a learner and determine if the trained learner could still correctly classify traffic whose attributes had been recorded 1 year or more from the training set date. The second concern is that once the model is built, how long can it be characterized as reasonably effective, given the fluid nature of the internet and the even more fluid nature of criminal expertise.

In his paper Wald discusses the datasets and their strengths and weaknesses.

Related work

Any student of machine learning will learn quickly that a goal of Machine Learning is to analyze data and detect patterns. Once patterns are detected, the pattern characteristics may be used to classify data of an unknown nature. In his paper Wald discusses the current state of work in Machine Learning with respect to Network Intrusion Detection. Referring to the work of Lappas et al. [LAPP]he notes that data mining can assist in intrusion detection. For example traffic meeting the characteristics of intrusion can be removed from the normal workflow which permits the network manager to focus on other traffic. Another author proposed a hybrid method which uses statistical data and frequency analysis to detect traffic deviating from the normal trend [NO01], [NO02].

Beaver et al. in their work “A learning system for discriminating variants of malicious network traffic” [BEAV] proposed a system that used the DARPA/KDDCUP-99 data sets as training data, then extracted certain features that could be used to characterize traffic as normal or anomalous. Key to their investigation is the use of the protocol header which determined routing and frequency of the connection requests indicated anomalous traffic. They state that their architecture will correctly classify 82% of the data and catch 89% of day-zero events.

Wald et al. - Approaches to Problem

In their paper Wald et al. [WALD] do not attempt to define a new solution to the problem of network intrusion detection. They attempt to determine a window during which an already existing model may be used. Their work reviewed datasets for network intrusion and then focused on the Kyoto dataset due the fact that it has data over a longer period of time. They note that many Intrusion Detection Systems are designed to run over longer periods of time, their engines do not use Machine Learning techniques to generate their rules during operation and may not be focused on the particular network/system in which they are deployed. The Wald team chose to examine the validity of classification techniques trained at a fixed point in time and then used data sets from 6 to

19 months away from the training time, thus giving a realistic model of the potential lifespan of a model.

Classification Algorithms

The team chose 4 learners, 5-Nearest Neighbor, two forms of C4.5 Decision Trees (C4.5D & C4.5N) and Naive Bayes. They state that these algorithms for classification were chosen specifically due to their dissimilarity to each other and the ease of their computation. They chose to classify the malicious sessions (as noted in the Kyoto dataset) as “positive results” and thus normal traffic was “negative”.

Feature Selection

Feature selection is key to any Machine Learning study. The team chose to focus on three features:

- Chi-Squared which uses the X^2 statistic to measure the relationship between a statistic and the independent variables of the class.
- Receiver Operating Characteristic which considers the feature with the class value and normalizes the class value which allows it to be used as a predictor
- Signal to Noise ratio which computes the ratio of the difference between the mean and the feature divided by the sum of the standard deviations for all the class values.

Performance Evaluation

The team chose Area Under the Receiver Operating Characteristic curve as the measure of effectiveness. This was selected because the curve is calculated based upon the ratio of True Positive vs. False Positive rates as the threshold is varied. Unlike the Receiver Operating Characteristic curve which uses one attribute as the normalized value from the classification model, the area curve uses the actual output from the classification model as its data. All models were trained on the same data set (Kyoto 2008-01-01) and therefore cross validation, which would reduce errors due to different training data sets, was not needed.

Kyoto Dataset considerations

The team chose the Kyoto dataset because it had two major characteristics they needed for their research:

- The data occupied a large period of time
- The attacks were not artificially generated which meant that they could be reasonably relied on to represent a real-world scenario.

The “near term” contains data from the 10th and 25th of the months from July through and including December 2008. The “far term” data began on July 1, 2009 and continue every 7 days until August 26 2009, with the exception of August 11, 2009 which had an abnormally small number of instances (10,716 versus 127,123 for August 12).

Attributes which would lead to a direct classification of the connection as an intrusion were removed from the analysis, such as:

- security analysis – which was generated after the intrusion was already classified as intrusion
- destination IP address – traffic generated internally would use a different destination IP address than an external source. Removal of this aids in simulating “real world” traffic and data.
- class label which was a result of the analysis

Results from Wald et al.

The Area Under Curve results for the near-term results from the Wald study are outlined in <<insertreferencehere>> and the Area Under Curve results from the long term are listed in <<othertablehere>>. As can be seen from the data 15 models were trained on the 2008-01-01 data using different combinations of the four learners and four feature selection options (including no feature selection). Following the convention of the Wald document, the highest performing learners are highlighted in **bold** and the lowest performing learners as *italicized.*

Beginning with the near term data it can be seen from the table that the results from all learners were very good until the end of the near term where their reliability began to drop and in 4 tests dropped below 0.9. While it is clear that the 5 Nearest Neighbors could produce the best results in a majority of the cases it is, as was noted before, computationally intensive. It is obvious from the data that the model built from the top learner using signal to noise feature selection will have a success rate greater than 0.999. When nine months away from the initial learner the data built using 5 nearest neighbor as the learner with consistently produces the best results. However, as has been noted many times, the 5 nearest neighbor learner requires larger computational overhead than other learners and may not be the best for a live environment. A reliable second best would be either of the C4.5 learners paired with signal to noise feature selection. As Wald noted, the 5 nearest neighbor will always result in an area under the curve of 0.999 and will always be one of the top 2 performers. A good second choice is the C4.5 learners coupled with Signal to noise feature selection which will give good results (area under curve of 0.99 or greater) with less computational impact. In the near term the Naive Bayes was the worst performer in most tests. Specifically, it was the best choice in only 2 of the 182 tests and worst in 171 of the 182 tests performed.

In the near term feature selection, no feature selection yielded the best results when paired with the C4.5D learner. However, the no feature selection did not perform the best with each data set when not paired with C4.5D. No feature was never the worst. Looking at the data, we can see that in the middle of the near term range (September – November, 2008) the 5 nearest neighbor learner performed well with all feature selection options. Surrounding the 5 nearest neighbor, in the July – August and late November – December, data the C4.5 learners coupled with either no feature selection or Signal To Noise gave the best performance.

Looking at Wald's long term data, in which he examined the similar months (July – August), however 1 year later (2008 vs. 2009), we see similar trends although not always the same learner and feature selection combinations and at slightly lower levels of efficiency (0.985 vs 0.973 overall). What is surprising is that in each of the 9 datasets we can find a combination of learner and feature selection that has an effectiveness greater than 0.99. Again the Naive Bayes learner was had the lowest effectiveness rate in 29 of the 144 tests and was the best choice in only 2 of the tests. Similarly the 5 nearest neighbor learner tended to produce the best results with the C4.5D learner usually second. Looking at the Signal to noise feature selection and the C4.5 learners we see that the C4.5N, coupled with the signal to noise feature selection would produce an area under the curve of 0.99 in all but one test.

Conclusions

As Wald noted, the far term data sets are chronologically closer together than the near term data sets, thus side by side comparisons make it difficult to determine any trends. The overall performance remained nearly the same for long term and short term and the trend for the 5 nearest neighbor to produce the best results in the middle of the datasets continued. What is unusual in the data is that it is not dissimilar. It would be expected that as time progressed further from the initial trained data set the attack data should change. The authors did not speculate on why the data did not change. However I would propose that since the Kyoto dataset is measuring network intrusions and reset the network after each detected intrusion, this indicates that the intrusions are not done as part of an intelligent attack, but an automated process to probe, explore and report. Since the explore option was not allowed to complete, the explore did not occur. I would suggest that there would be a significant increase in traffic, if the initial intrusion into the network had been permitted to establish a "beachhead" where data could be gathered and relayed back to the source servers.

In my view, the authors in their study were effective in demonstrating that a model created could remain valid for up to 19 months after the initial intrusion. Their results indicated that the learners that required the most extensive CPU time had the highest accuracy and the fastest learner the least accurate. Not surprising the "near term" results had an AUC of 0.999, while the "far term" had an AUC of 0.99. The Wald study determined that although not the highest every day, generally the C4.5D/N learners with the Signal to Noise feature selection would reliably produce an AUC of

greater than 0.99 for both “near term” and “far term”. The nearest neighbor learner was highly effective when the data was 10-12 weeks out, however even then the Signal to Noise feature selection was still effective.

What is interesting is that without additional training the model was effective 19 months after the initial date. It can be safely assumed that there were intrusions using a profile that did not exist prior to the start of the data set.

Future Work

Of interest in this paper would be to combine the results of Wald [WALD] and Beaver [BEAV] where a Machine Learning environment could be created in parallel with a detection environment to create a self adjusting network intrusion detection system. Additionally, as suggested by Wald, a study involving older data would be useful. Additional tests could be performed with additional feature selection options and different learners. The data from the Kyoto dataset continues through December 2015 and additional tests could be performed on the data longer term to determine model effectiveness over a the 9 year period. A slow decline in the models effectiveness could be an indication that most intrusions are simple probes to determine weakness generated by software and that more dedicated or focused attacks may indicate human intervention.

This study has implications in the realm of intrusion detection and protection, I believe that the Kyoto data set and Wald have shown that entities are continually attempting to probe systems in a mechanical method and when an weakness is found traffic profiles will change significantly.

Tables

Table 1: AUC Results for Near Term (6-12 month) Test Datasets

Test Dataset	Feature Selection	Learner			
		5-NN	C4.5D	C4.5N	NB
2008-07-10	No FS	0.99820	**0.99836**	0.99654	*0.98764*
	CS	**0.98956**	--	*0.98520*	0.98552
	ROC	0.99909	0.99863	**0.99914**	*0.98816*
	S2N	0.99947	0.99912	**0.99950**	*0.99649*
2008-07-25	No FS	0.99874	**0.99952**	0.99682	*0.98129*
	CS	**0.99926**	--	0.99583	*0.99233*
	ROC	**0.99939**	0.99638	0.99897	*0.96832*
	S2N	0.99960	0.99920	**0.99964**	*0.98927*
2008-08-10	No FS	0.99722	**0.99782**	0.99745	*0.98862*
	CS	*0.97955*	--	**0.99555**	0.99225
	ROC	**0.99928**	0.99739	0.99885	*0.98605*
	S2N	**0.99943**	0.99895	0.99938	*0.99645*
2008-08-25	No FS	0.99088	**0.99708**	0.99458	*0.98700*
	CS	**0.99893**	--	0.99660	*0.99452*
	ROC	0.99403	0.99762	**0.99913**	*0.98877*
	S2N	0.99962	0.99943	**0.99965**	*0.99692*
2008-09-10	No FS	**0.99508**	0.99187	0.99394	*0.98659*
	CS	0.99458	--	**0.99535**	*0.98608*
	ROC	0.99816	0.99709	**0.99898**	*0.98451*
	S2N	0.99807	0.99895	**0.99958**	*0.99757*
2008-09-23	No FS	**0.99809**	0.99252	0.98971	*0.97441*
	CS	0.93165	--	**0.98604**	*0.97542*
	ROC	**0.99881**	0.99554	0.99830	*0.97342*
	S2N	**0.99943**	*0.94104*	0.99645	0.99910
2008-10-10	No FS	**0.99715**	0.98938	0.98526	*0.97826*
	CS	**0.99316**	--	*0.97713*	0.98024
	ROC	**0.99922**	0.99200	0.99413	*0.98504*
	S2N	**0.99936**	*0.95132*	0.99501	0.99646
2008-10-25	No FS	0.99932	**0.99947**	0.99541	*0.98350*
	CS	**0.99042**	--	0.98990	*0.97289*
	ROC	**0.99969**	0.99292	0.99577	*0.97895*
	S2N	**0.99962**	0.99670	*0.99660*	0.99911
2008-11-10	No FS	**0.99469**	0.98841	*0.97940*	0.98238
	CS	*0.97970*	--	0.98477	**0.98543**
	ROC	**0.99687**	0.98847	0.99354	*0.98055*
	S2N	0.99578	*0.95695*	0.98977	**0.99670**
2008-11-25	No FS	0.98143	**0.99774**	0.99470	*0.94754*
	CS	**0.99127**	--	0.98993	*0.94808*
	ROC	**0.99432**	*0.87831*	0.99008	0.95099
	S2N	0.98762	0.99562	**0.99683**	*0.97578*
2008-12-10	No FS	0.99016	**0.99724**	0.99078	*0.97426*
	CS	**0.99750**	--	*0.98846*	0.98859
	ROC	**0.99252**	0.98390	0.98416	*0.96760*
	S2N	0.99257	*0.94983*	**0.99292**	0.98094
2008-12-25	No FS	0.96021	**0.99072**	0.98547	*0.88475*
	CS	*0.83107*	--	**0.99839**	0.96755
	ROC	**0.99121**	*0.73633*	0.95580	0.89229
	S2N	0.99633	0.99591	**0.99657**	*0.93357*

Table 2: AUC Results for Long-Term (18-19 months) Test Datasets

Test Dataset	Feature Selection	Learner			
		5-NN	C4.5D	C4.5N	NB
2009-07-01	No FS	0.98416	**0.99517**	0.98998	*0.9356*
	CS	**0.99594**	--	0.98776	*0.93972*
	ROC	**0.98813**	0.97659	0.975	*0.92236*
	S2N	0.98759	**0.99812**	0.99655	*0.99263*
2009-07-08	No FS	0.97484	*0.94669*	**0.98375**	0.96057
	CS	**0.99514**	--	*0.93943*	0.97557
	ROC	**0.99752**	0.99435	0.996	*0.97577*
	S2N	0.99762	0.99862	*0.99687*	**0.99864**
2009-07-15	No FS	**0.98066**	*0.92207*	0.9794	0.95124
	CS	**0.99815**	--	*0.94032*	0.95684
	ROC	**0.99792**	0.99	0.99032	*0.95876*
	S2N	0.99796	0.99631	*0.99208*	**0.99847**
2009-07-22	No FS	0.98984	**0.99158**	0.98621	*0.93282*
	CS	**0.99222**	--	0.97976	*0.94226*
	ROC	**0.99082**	0.98647	0.98659	*0.9277*
	S2N	0.98779	0.9891	**0.98974**	*0.97237*
2009-07-29	No FS	*0.87439*	**0.99909**	0.99869	0.93344
	CS	**0.99852**	--	0.99743	*0.94051*
	ROC	**0.92009**	*0.57019*	0.89387	0.90711
	S2N	*0.88255*	**0.99929**	0.99915	0.96439
2009-08-05	No FS	0.99175	**0.999**	0.99748	*0.94751*
	CS	0.98548	--	**0.9939**	*0.93008*
	ROC	0.99146	0.99204	**0.99251**	*0.94158*
	S2N	0.99717	**0.9988**	0.99847	*0.99142*
2009-08-11	No FS	0.99296	**0.99853**	0.99616	*0.97056*
	CS	**0.99575**	--	0.99084	*0.96532*
	ROC	**0.99908**	0.99399	0.99698	*0.97089*
	S2N	**0.99903**	0.99837	*0.99775*	0.99797
2009-08-19	No FS	0.93471	**0.99885**	0.99764	*0.91773*
	CS	0.95607	--	**0.99632**	*0.89347*
	ROC	0.94256	0.91848	**0.96257**	*0.90635*
	S2N	0.99234	0.99823	**0.99854**	*0.96936*
2009-08-26	No FS	0.99623	**0.99836**	0.99587	*0.96964*
	CS	**0.9977**	--	0.99369	*0.96028*
	ROC	**0.99928**	0.99578	0.99656	*0.94544*
	S2N	**0.99932**	0.9979	0.99744	*0.97674*

Bibliography

SON: J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nak, Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for nids detectio, 2011
WAL: R. Wald, T. Khoshgoftaar, R. Zuech, and A. Napolitan, Network Traffic Prediction Models for Near- and Long-Term Prediction, 2014
SHI: A. Shiravi, H. Shiravi , M. Tavallaee , A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detectio, 2012
LAP: T. Lappas and K. Pelechrini, Data Mining Techniques for (Network) IntrusionDetection System, 2007
NO0: Novakov, S, Combining Statistical and Spectral Analysis Techniques in Network Traffic Anomaly Detection, 2012
NO0: S. Novakov, C. H. Lung, I. Lambadaris and N. Seddig, Studies in applying PCA and wavelet algorithms for network traffic anomaly detection, 2013
BEA: J. Beaver, C. Symons, R. Gille, A learning system for discriminating variants of malicious network traffi, 2013