Anti-Phishing Pilot at ACM IWSPA 2018

Evaluating Performance with New Metrics for Unbalanced Datasets

Ayman El Aassal
aelaassal@uh.edu

Luis Moraes
ltdemoraes@uh.edu

Shahryar Baki
sh.baki@gmail.com

Avisha Das
adas5@uh.edu

Rakesh Verma
rverma@uh.edu

Computer Science Department
University of Houston
Houston, TX 77204

Abstract

This paper provides a summary of the IWSPA Anti-Phishing shared task pilot. The pilot consisted of two subtasks: identifying phishing emails from a collection of legitimate and phishing email bodies, and separating phishing emails from legitimate emails when given full emails, i.e., with headers and bodies. For both subtasks, training datasets were made available approximately a month before the test data was released. Sixteen teams registered for the task and nine submitted models and predictions for the test data. We discuss the collection sources and preprocessing of the datasets, and the performance of the teams on the test data from several different perspectives. A unique aspect of the dataset was that it included synthetic attacks. Another emphasis in both subtasks was that the phishing class size was almost an order of magnitude smaller than the legitimate class size to reflect the real-world scenario. Hence, we introduce two evaluation metrics, called *balanced detection rate* and *normalized balanced detection rate*, which to our knowledge are new and more suitable for unbalanced datasets. We then evaluate the performance of the teams on the usual metrics as well as metrics for unbalanced datasets, including the new metrics. Two baseline methods, multinomial Naive Bayes and Logistic Regression, are also included for comparison.

In: R. Verma, A. Das. (eds.): Proceedings of the 1st Anti-Phishing Shared Pilot at 4th ACM International Workshop on Security and Privacy Analytics (IWSPA 2018), Tempe, Arizona, USA, 21-03-2018, published at http://ceur-ws.org

1 Introduction

With an increasing dependence on the Internet, there has been a growth in the number as well as variety of *social engineering attacks*. Phishing is a common social engineering attack, where attackers disguise themselves as legitimate entities to steal the digital identity of unsuspecting people, who often incur substantial financial losses. Because it targets people, phishing has also been the attack of choice for bringing otherwise well-protected organizations to their knees.[1]

According to the authors in [ST16], phishing attacks have been one of the oldest, yet effective, weapons used for exploitation. Over the last decade, there has been extensive research on detection and protection against phishing attacks, e.g., see [CNU06, VSH12, VH13, VR15, VA17, AZ17, DBA+18]. However, reports published by organizations like PhishLabs[2] and Anti-Phishing Working Group (APWG)[3] shed light on the seriousness of phishing as a growing threat to cybersecurity. According to APWG, approximately 300,000 unique phishing reports were submitted in the third quarter of 2017 alone. PhishLabs identified 170,000 unique phishing domains in 2017, an increase of 23% from 2016. These statistics corrobo-

[1] In phishing, the attacker does not have to do sophisticated reconnaissance to find vulnerable networks or applications to attack.

[2] https://pages.phishlabs.com/rs/130-BFB-942/images/2017+PhishLabs+Phishing+and+Threat+Intelligence+Report.pdf

[3] http://docs.apwg.org/reports/apwg_trends_report_q3_2017.pdf

rate the fact that despite being a widely researched topic, the phishing threat is far from being solved.

The common form of phishers' *modus operandi* starts with sending an email, usually the most common attack vector, embedded with a poisoned link or a malicious attachment.[4] If an unknowing victim clicks on the embedded URL, they can be directed to a malicious website. Similarly, clicking on the malicious attachment may cause malware to be downloaded on the victim's computer. To prevent such an attack, an ideal classifier should detect the threat instantaneously and take precautions by increasing distance between the victim and attacker.

In the first year of our Anti-Phishing Shared Pilot, we focus on detection of phishing emails. Apart from attachments, an email consists of two major parts - the *full header* and the *body*. Thus, we proposed two subtasks based on the type of email data provided - (i) *Subtask A:* Emails with only the body content, and (ii) *Subtask B:* Emails with full header information and body content. None of the tasks had emails with attachments. If a source email had an attachment, it was removed before inclusion. The justification for this decision is that malicious attachment detection techniques are orthogonal and hence out of scope of this pilot, which focuses on detection of phishing.

The pilot was announced in early January 2018 and 16 teams registered at the task site on Easy Chair. After the training and test datasets were released, nine teams submitted their best performing model based on the training datasets, and predictions for evaluation. The test datasets were released about a month after the training datasets were released and teams had approximately five days for submitting their predictions and their top models. All teams, which submitted materials for evaluation, were invited to present a poster at the 4th ACM International Workshop on Security and Privacy Analytics (IWSPA) 2018 in Tempe, Arizona.

In this overview of our Shared Task, we give a detailed explanation of our corpus collection (Section 2.1), preprocessing steps and challenges (Section 2.2). We describe the evaluation metrics followed by the evaluation of the system submissions in Section 4. Section 5 concludes with some insights and perspectives from the shared task and performance of the systems.

2 Datasets

Gathering datasets and preprocessing them for the two subtasks turned out to be quite challenging. This was especially critical for the Header Subtask. It required pristine and complete headers, which many datasets do not have, e.g., the Enron dataset emails have abbreviated and sanitized headers for privacy reasons. We took care to check the documentation for each dataset and the dataset itself for any signs of sanitization.

2.1 Dataset Sources

Our objective for the dataset was to make it as diverse as possible, so we gathered emails from as many sources as we could find. Legitimate emails were relatively easy to find compared to phishing ones. There are two reasons for this: (i) corporations are not inclined to share or make public the phishing emails they receive, and people generally delete them and move on, and (ii) quite a few accounts of public personas, and some companies, have been hacked and their emails have been released, generally to embarrass them and/or score political points.[5]

We gathered legitimate emails from different sources. These include email collections from Wikileaks archives, e.g., Democratic National Committee, Hacking Team, Sony emails, etc. We used selected emails from the Enron Dataset[6] and SpamAssassin[7] as well.

As for the phishing emails in our dataset, they were collected from the IT departments of different universities. We also included emails from the popular Nazario's phishing corpora[8], and synthetic emails created by organizers. Note that the emails collected from universities' IT departments usually do not have a full header, that is why we only used these sources for the No-header Subtask. The same is true for the synthetic emails.

The synthetic emails are emails artificially created by the organizers using Dada engine,[9] which is a system that generates text based on a pre-specified grammar. We based the grammar on phishing emails from Nazario's dataset. Dada has been used previously to create email masquerade attacks in [BVMG17]. A more detailed breakdown of the preprocessing of emails is to be found in the next subsection. As can be seen from Table 1, the ratio of phishing items to legitimate ones is approximately 1:9.

2.2 Preprocessing

Preprocessing of the dataset turned out to be a delicate task due to the highly diverse nature of emails, even

[4] Nowadays, there are also social networks and text messages as convenient vehicles for phishing, but emails still remain popular.

[5] The Enron email dataset, where emails became a matter of public record due to a court case, is an exception.

[6] https://www.cs.cmu.edu/~enron/

[7] http://www.csmining.org/index.php/spam-assassin-datasets.html

[8] https://monkey.org/~jose/phishing/

[9] http://dev.null.org/dadaengine/

Table 1: Dataset Statistics

	Legitimate	Phishing	Total
Train	5088	612	5700
Test	3825	475	4300

(a) No-header Dataset

	Legitimate	Phishing	Total
Train	4082	501	4583
Test	3699	496	4195

(b) Header Dataset

emails from the same source.

The phishing emails that we collected had different URL problems. The phishing emails from universities' IT departments did not include the phishing links in their reported emails, for obvious reasons, and the URLs from Nazario's dataset are old and link to dead websites.

The URLs in the emails from the legitimate sources were too revealing in the sense that they could lead participants and classifiers to recognize the sources.

To handle these issues we decided to normalize all the URLs in both datasets to ⟨⟨link⟩⟩. Another possible approach would be to replace all the URLs with live phishing links from Phishtank. The concern with this approach was the possibility of the classifiers picking up on some idiosyncrasies of such URLs, or participants noticing it and using it as a feature.

Another concern that we needed to address is the recognizability of the datasets. So we tried to remove, as much as was feasible, from the emails any signs that could hint at the origin of the datasets. For this purpose, we included in the preprocessing steps the normalization of organizations' or universities' names, recipients' names, domain names, signatures, threading and remove non-English emails.

We also made sure to remove emails that are too big (more than 1 MB) or too small (the threshold for removing smaller size emails varies with each dataset) and remove all base64 encoded text.

In order to remove as much noise as possible, we attempted to remove leftover HTML tags and empty spacing that resulted from parsing the body of the email using an HTML parser.

Although the preprocessing was not perfect, because a fraction of the emails were too noisy, we took considerable care to make the ensure that the quirks in the datasets did not make the tasks significantly easier than real-world scenario. As a final check before release, a logistic regression classifier was run on the datasets to check the hardness of the classification task.

3 Overview of Participating Systems

Nine teams submitted their best models on the training set and the predictions of their models on the test sets for the two tasks. However, during the review process it was discovered that some of the teams used techniques that had a high degree of overlap, so they were asked to combine their efforts into one paper for the proceedings. Hence, Table 2 shows the remaining team names and their alternative names that we use throughout the paper. In the rest of this section, we give a short description of the approach(es) taken by each team and a brief perspective on the results. The next section goes into more details on the performance.

Table 2: Participating teams. Every team participated in both subtasks.

Team name	Alt. name	Reference
CEN-SecureNLP	sys1	[RNU$^+$18]
CEN-AISecurity@IWSPA-2018	sys2	[RHP$^+$18]
CEN-DeepSpam	sys3	[MURS18]
CEN CryptCoyotes	sys4	[MRRK18]
Security-CEN@Amrita	sys5	[UBS$^+$18]
Amrita-NLP	sys6	[HRMP18]
CENSec@Amrita	sys7	[VNRK18]
TeamTripleN	sys8	[NNN18]
CEN-Security@IWSPA 2018	sys9	[NRK18]

CEN-SecureNLP (sys1), [RNU$^+$18] trained several classification techniques (Naive Bayes, SVM, Decision Tree, kNN, Logistic Regression, AdaBoost and Random Forest) on the TF-IDF and Doc2Vec representation of the emails. They did not use any header related features for the Header subtask. They did 10-fold cross validation on the training set with different classifier and representations and chose the best system to run on the test set. For both tasks, doc2vec outperformed the TF-IDF representation. SVM and Adaboost were the best classifiers for No-header and Header Subtasks respectively. They used accuracy as a metric for ranking their models, which is not recommended for an unbalanced dataset. In the No-header Subtask, they classified everything as legitimate and got 0 F_1-score. Their performance for the Header Subtask is slightly better than the No-header subtask and achieved 2% F_1.[10]

Team CEN-DeepSpam (sys3), [MURS18], uses Keras to build an embedding layer with word tokens from the email data. This embedding layer is used to train a Convolutional Neural Network, which acts as the email classifier. The authors propose a total of five models depending on the type of subtask (with headers and without headers) and number of training epochs (100, 500 and 1000 epochs for No-header Subtask and 100 and 500 epochs for Header Subtask).

[10]These results assume that phishing emails make up the positive class and legitimate emails the negative class.

CEN-DeepSpam came fourth on No-header SubTask and third on Header SubTask with F_1-scores of 63.8% and 92.4% respectively.

Team CEN-Security@IWSPA 2018 [NRK18], used *TF-IDF* to convert the email contents into the numeric feature vectors. Then they applied several machine learning techniques (Decision tree, K-NN, Naive Bayes, Random forest, SVM, and logistic regression) to classify the emails. In order to reduce the size of feature space they applied Singular Value Decomposition (SVD) and Non-Negative Matrix Factorization (NMF) to both TF-IDF vectors. Authors in [VNRK18] also built the same model but instead of using *TF-IDF*, they used *Term Document Matrix*(TDM) as a representation of the emails. For No-header Subtask, KNN with TDM and NMF outperformed the other combinations with F_1-score of 50.88%. For the Header Subtask, they did not extract any specific feature from email's header and they got F_1-score of 50.78% using KNN with TF-IDF and NMF.

Team TripleN (sys8), [NNN18], used deep learning with supervised attention. Their method for email body classification task has two layers: the word layer and the sentence layer. The word layer includes: a word embedding component, a bidirectional LSTM for the word level and an attention module. The sentence layer has a bidirection LSTM. For the attention module, they identified words that appeared more frequently in phishing emails and less frequently in legitimate emails. Their attention module idea is similar to the idea of [VH13], where bigrams were extracted based on their discrimination power. For the Header Subtask, they use another bidirectional LSTM. They also extract bodies of the emails from the Header Subtask to increase the size of the training corpus for the No-header Subtask. TripleN's deep learning approaches did quite well on both the tasks when the F_1 score is considered: 83.5% on the No-header Subtask and 93.0% on the Header Subtask.

Team AISecurity (sys2), [RHP$^+$18], captured syntactic and semantic features of emails in the dataset using word embedding (with word2vec method) and Neural Bag-of-Ngrams, which is a real-valued representation that captures the semantics of a text. These features are used to train four different deep learning algorithms: Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Multilayer Perceptron (MLP), and used a sigmoid function as an activation function. Four models were reported for each subtask (eight in total), with Word Embedding being used with CNN, RNN, and LSTM, and Neural bag-of-Ngrams used with MLP. Their highest F_1-score 57.09% was with the model combining Word embedding and LSTM on the No-header Subtask. For the Header Subtask, they were able to achieve an F_1-score 57.09% using MLP with Neural Bag-of-Ngrams.

Team CEN CryptCoyotes (sys4) [MRRK18] also employed word embedding with Word2vec to convert data, then compared the results of the following three classifiers: Multilayer Perception (MLP), Convolutional Neural Network (CNN), and Recursive Neural Network (RNN). All classifiers were used with a sigmoid function to classify phishing from legitimate emails. However, they used different parameters for Word2Vec compared to [RHP$^+$18]: Higher values for training sample (batch-size), word vector dimension, skip-window, number of negative samples, learning rate, but a lower number of iterations. CNN classifier gave the best results for No-Header Subtask with an F_1-score of 44.801%, whereas RNN gave the best results for Header Subtask with an F_1-score of 53.18%.

Team Amrita-NLP (sys 6), [HRMP18], also used word embeddings. However, the method used to obtain embeddings was fastText [BGJM16], which also takes into consideration subword information. In addition to training a model for each task using the data available for that task, the team also trained a model on the combination of both datasets. However, the combined method performed slightly worse than the individually trained models.

Team Security-CEN@Amrita (sys5), [UBS$^+$18], applied three different classifiers: Naive Bayes, logistic regression, and SVMs. The data was first converted into TF-IDF vectors then augmented with domain-level features (for instance, a list of common words in phishing emails). Support Vector Machines outperformed the other two methods in both subtasks.

We now take a deeper look into the performance of the teams, based on a number of metrics. Some of the metrics are more suitable for unbalanced datasets. We also propose two new metrics for unbalanced datasets, which we call Balanced Detection Rate and Normalized Balanced Detection Rate.

4 Evaluation

Due to the similar nature of the two subtasks (both are binary classification tasks), we use the same metrics to evaluate them. Precision, recall (sensitivity), specificity(true negative rate) accuracy and F_1-score are the most common metrics to quantify the performance of classifiers. Equations 1, 2, 3, 4 and 5 show the formulas for these metrics.

$$Considering,\ P = TP + FN\ and\ N = TN + FP \tag{1}$$

$$Accuracy = \frac{TP + TN}{P + N}$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall/Sensitivity = \frac{TP}{TP + FN} \quad (3)$$

$$TNR/Specificity = \frac{TN}{TN + FP} \quad (4)$$

$$F_1 = 2.\frac{Precision.Recall}{Precision + Recall} \quad (5)$$

Due to the imbalanced nature of our dataset and different cost of misclassification, these metrics do not reflect a realistic comparison. Suppose we have two systems A and B in the No-header Subtask and phishing is the positive class. If system A tags all the emails as legitimate (true negative = 3825 and false negative = 475), the resulting accuracy score is 89%. On the other hand, if system B can correctly identify say half of the phishing emails (TP = 237, FN = 238), and has some errors in legitimate emails (FP = 346, TN = 3479), despite its superior performance, its accuracy will be less than system A's performance (86%). So, we need some metrics that take into account the ratio of legitimate to phishing.

4.1 Metrics for Unbalanced Datasets

As discussed in [BDA13], geometric mean (G-mean) and Matthews correlation coefficient (MCC) are better metrics, since they balance the classification performance between minority and majority class. Equations 6 and 7 are the formulas for G-mean and MCC.

$$G - mean = \sqrt{sensitivity * specificity} \quad (6)$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TN + FN)(P)(N)}} \quad (7)$$

MCC ranges from One to -1. One stands for perfect prediction and -1 for worst prediction. For the G-mean, better prediction will lead to higher G-mean value and poor prediction will lead to lower G-mean value.

4.2 New Metrics for Unbalanced Datasets

We also propose new metrics, called "Balanced Detection Rate" and Normalized Balanced Detection Rate to rank systems. The idea for Balanced Detection Rate is to measure how many minority class instances were correctly identified and to charge appropriately using the incorrect instances of the majority class. So, we divide the number of correctly identified minority class instances by number of incorrectly classified majority class instances (Equation 8). Let $c = NEG/POS$. If $c > 1$, then the positive class is the minority class, so

$$BDR = \frac{TP}{1 + FP},\ BDR(\%) = 100 * \frac{DR}{TP + FN} \quad (8)$$

If $c < 1$, then the negative class is the minority class, so we replace TP by TN in the numerator and FP by FN and we take the NEG = TN + FP in the denominator for the DR% formula. If $c = 1$, then the dataset is balanced so both Detection Rates should be calculated and reported.

Observe that only a perfect classifier with FP = FN = 0 can have BDR% = 100%. We can generalize these definitions to take also cost of misclassification and benefit of minority class detection into account. For example, if $c > 1$ and α is the benefit of detecting a minority class instance, and β (γ) are respectively the cost of misclassifying the majority (minority) class instances, then we replace TP by $\alpha * TP$, FP by $\beta * FP$ and FN by $\gamma * FN$. Similarly, we can handle the case for $c < 1$ and again we should reported both generalized versions of Detection Rates when $c = 1$.

The 1:1 charging scheme may be too considered "too harsh" in some unbalanced situations. We also define a Normalized Balanced Detection Rate (NBDR), which normalizes the charge based on the size of the classes, as follows. Again, this assumes that positive class is the minority class. Note that the numerator is just the detection rate for the positive class DR(p) and the denominator is 2 - DR(n), where DR(n) is the detection rate for the negative class.

$$NBDR = \frac{\frac{TP}{TP+FN}}{1 + \frac{FP}{TN+FP}} = \frac{DR(p)}{2 - DR(n)} \quad (9)$$

For NBDR, we have NBDR% = NBDR * 100. Normalization may have another advantage, comparing across datasets that are similar in size and composition.

4.3 Baselines

We ran two different baselines on the data: Multinomial Naive Bayes and Logistic Regression. Both baselines score quite highly on most performance metrics. The data was preprocessed through tokenization and stop word removal. Simple word occurrence counts were used as features. For a word to be considered as a feature it must have appeared in at least five different emails. The models were trained for each subtask separately using only the training data for that specific subtask. For the Header Subtask, the email headers were tokenized just as the email bodies. We report their performance on the test set.

Table 3: Confusion matrices of top submissions for No-header Subtask based on BDR metric

Systems	TP	TN	FP	FN	BDR%
sys5	258	3807	18	217	2.85
sys7	115	3816	9	360	2.42
sys8	401	3741	84	74	0.99
sys6	347	3742	83	128	0.86
MNB	300	3721	104	175	0.60
LR	398	3665	160	77	0.52
sys3	287	3688	137	188	0.43
sys2	237	3479	346	238	0.14
sys4	237	3479	346	238	0.14
sys9	26	3726	99	449	0.05
sys1	0	3825	0	475	0

Table 4: Confusion matrices of top submissions for Header Subtask based on BDR metric

Systems	TP	TN	FP	FN	BDR%
sys6	480	3680	19	16	4.83
MNB	478	3671	28	18	3.32
sys8	458	3668	31	38	2.88
sys3	496	3618	81	0	1.21
sys5	490	3612	87	6	1.12
sys9	210	3578	121	286	0.34
LR	496	3348	351	0	0.28
sys2	346	3329	370	150	0.18
sys4	363	3193	506	133	0.14
sys7	237	2747	952	259	0.05
sys1	7	3593	106	489	0.01

4.4 Detailed Performance

We received 41 submissions for No-header Subtask and 40 submissions for Header Subtask from nine different teams. Figure 1 shows F_1-scores of best submission for each team. For No-header Subtask, system 8 has the best F_1 score of 83.54% and for Header Subtask, system 6 has the highest F_1 score of 96.8%. F_1 score of the top system and also average F_1 score in Header Subtask is better than No-header Subtask. In order to emphasis on importance of choosing proper evaluation metrics while dealing with unbalanced dataset, we also calculated the F_1 score by changing the legitimate emails to positive class (Figure 2). The scores are higher compared to Figure 1 in which we considered phishing as positive class. Even for sys1, that classified everything as legitimate in No-header Subtask, the F_1-score is better than sys2, sys4, and sys9 (94.1%). Table 3 shows the confusion matrices of teams' top submission considering phishing as positive class.

The remaining metrics that we report in the rest of this section, G-mean, MCC and BDR, are not sensitive to changing the positive and negative class which is more appropriate for our evaluation.

Table 5: Detection rate (%) of teams on synthesized and non-synthesized phishing emails separately

Teams	Detection Rate	
	Synthesized	Non-synthesized
sys1	0.0	0.0
sys2	59.3	48.7
sys3	66.7	63.0
sys4	59.3	48.7
sys5	90.7	68.4
sys6	92.6	75.8
sys7	53.7	44.2
sys8	100	82.4
sys9	50	46.3
Average	65.6	54.9
MNB	85.1	60.3
LR	96.3	82.18

Figure 3 and 5 show G-mean and MCC values of submissions. For No-header Subtask, sys8 proposed by [NNN18] performed the best among other systems followed by sys5 [UBS$^+$18] and sys6 [HRMP18]. We intentionally removed the sys1 results for No-header Subtask in Figure 5 since it classified everything as legitimate and got MCC score of -1.

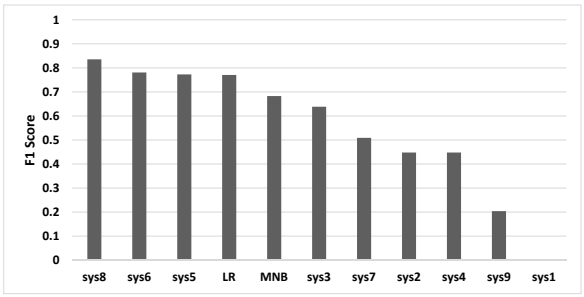
In Header Subtask, sys6 developed by [HRMP18] outperformed other systems. The next best performance belongs to sys3 [MURS18] and sys8 [NNN18].

4.5 Synthetic Data Evaluation

As mentioned earlier in Section 2, the phishing emails are gathered in two ways. One group is real phishing emails from different existing data sources, and the other group is computer generated based on a manually constructed grammar. Comparing the performance of the different systems on synthesized emails can give us an estimate of the similarity between these type of emails.

Here, we follow the same way that we used previously to report systems' performance on the whole dataset. First, we calculate systems' performance on synthesized and non-synthesized emails separately, then we chose the top performing submission of each team. Table 5 shows detection rates of top submissions among all systems.

Based on the detection rate result showed in Table 5, the average detection rate for the synthesized emails is higher than the non-synthesized ones which means synthesized email are detected correctly more than the others. Applying the t.test on these two groups shows a significant difference between them (p-value = 0.001).



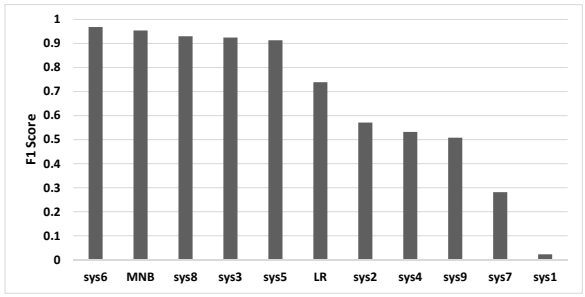
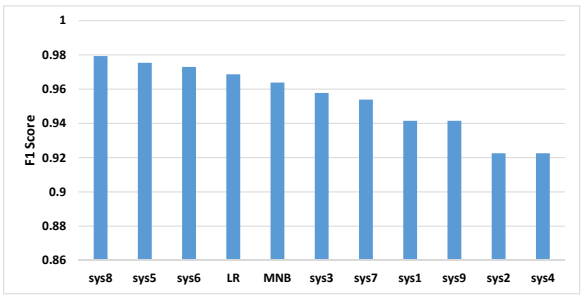


Figure 1: F_1-score of top submissions on No-header (left) and Header (right) Subtasks. Phishing is the positive class.



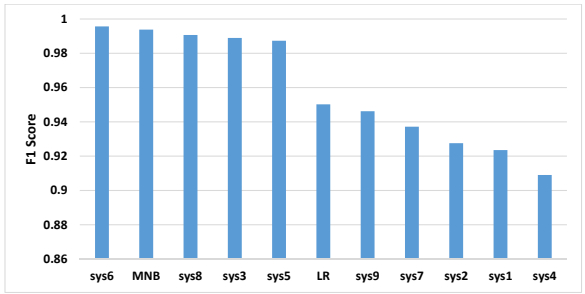
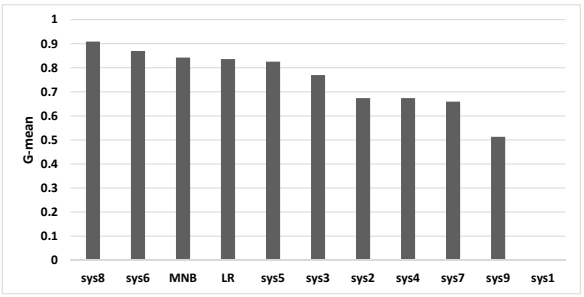


Figure 2: F_1-score of top submissions on No-header (left) and Header (right) Subtasks. Legitimate is the positive class.



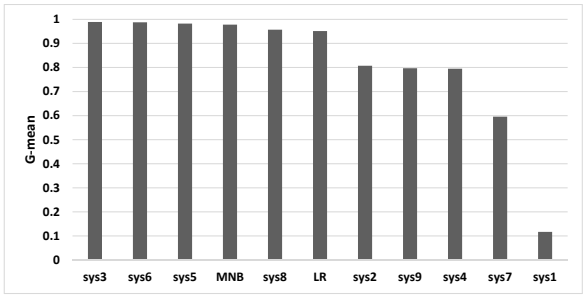
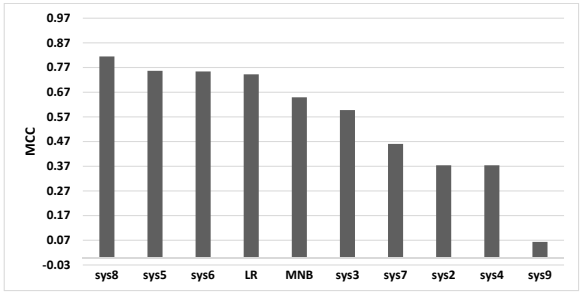


Figure 3: G-mean of top submissions on No-header (left) and Header (right) Subtasks



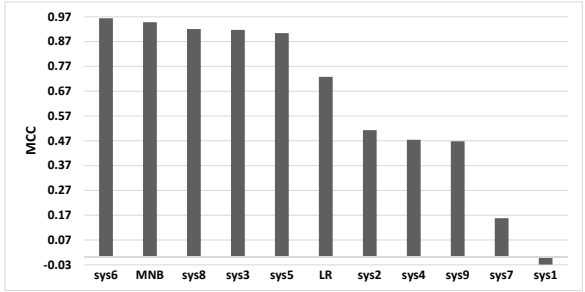
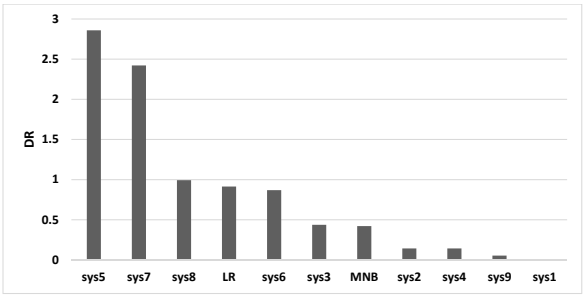


Figure 4: MCC of top submissions on No-header (left) and Header (right) Subtasks



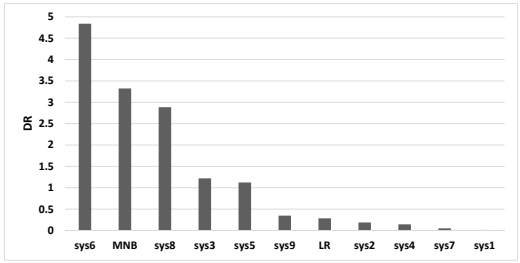


Figure 5: Balanced Detection Rate of top submissions on No-header (left) and Header (right) Subtasks

5 Conclusions

We introduced two new metrics for evaluating classifiers on unbalanced datasets, and examined the performance of the participating teams on both classical and new metrics. The first anti-phishing pilot at ACM IWSPA 2018 shows interesting correlations between the winning teams. In general, the deep learning models did quite well. With only the email bodies as input, logistic regression was a strong performer. However, the situation changed when headers were also provided.

The strong performance of Multinomial Naive Bayes (MNB) on the Header Subtask was surprising and needs closer investigation. It suggests that the header may be a rich source of useful features for phishing email detection. MNB's improvement over logistic regression on the Header SubTask could be because MNB is a generative model and logistic regression is discriminative [NJ01] and may need more data to achieve better performance. This hypothesis needs further investigation.

Acknowledgments

This research was supported in part by NSF grants CNS 1319212, DGE 1433817, DUE 1241772, and DUE 1356705. This material is also based upon work supported by, or in part by, the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-16-1-0422.

References

[AZ17] Ahmed Aleroud and Lina Zhou. Phishing environments, techniques, and countermeasures: A survey. *Computers & Security*, 68:160–196, 2017.

[BDA13] Mohamed Bekkar, Hassiba Kheliouane Djemaa, and Taklit Akrouf Alitouche. Evaluation measures for models assessment over imbalanced datasets. *J Inf Eng Appl*, 3(10), 2013.

[BGJM16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[BVMG17] Shahryar Baki, Rakesh M. Verma, Arjun Mukherjee, and Omprakash Gnawali. Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 469–482, 2017.

[CNU06] Madhusudhanan Chandrasekaran, Krishnan Narayanan, and Shambhu Upadhyaya. Phishing email detection based on structural properties. In *NYS Cyber Security Conference*, volume 3, 2006.

[DBA+18] Avisha Das, Shahryar Baki, Ayman El Aassal, Rakesh Verma, and Arthur Dunbar. SOK: Reconsidering phishing and spear phishing detection research from the security perspective, 2018. submitted for publication.

[HRMP18] Barathi Ganesh HB, Vinayakumar R, Anand Kumar M, and Soman K P. Distributed representation using target classes: Bag of tricks for security and privacy analytics. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[MRRK18] Vysakh S Mohan, Naveen J R, Vinayakumar R, and Soman KP. CEN CryptCoyotes: A.r.e.s : Automatic rogue email spotter. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[MURS18] Hiransha M, Nidhin A Unnithan, Vinayakumar R, and K P Soman. CEN-DeepSpam: Deep learning based spam detection. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[NJ01] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 841–848, 2001.

[NNN18] Minh Nguyen, Toan Nguyen, and Thien Huu Nguyen. A deep learning model with hierarchical lstms and supervised attention for anti-phishing. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[NRK18] Harikrishnan NB, Vinayakumar R, and Soman KP. A machine learning approach towards phishing email detection CEN-Security@IWSPA 2018. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[RHP+18] Vinayakumar R, Barathi Ganesh HB, Prabaharan Poornachandran, Anand Kumar M, and Soman K P. CEN-AISecurity: DeepAnti-PhishNet: Applying deep neural networks for e-mail spam detection. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[RNU+18] Vinayakumar R, Harikrishnan NB, Nidhin Unnithan, Soman KP, and Sai Sundarakrishna. CEN-SecureNLP: detecting e-mail spam using machine learning techniques. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[ST16] John Seymour and Philip Tully. Weaponizing data science for social engineering: Automated e2e spear phishing on twitter. 2016.

[UBS+18] Nidhin A Unnithan, Harikrishnan N B, Akrash S, Vinayakumar R, and Soman K P. Security-CEN@Amrita machine learning based spam e-mail detection. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[VA17] Rakesh M. Verma and Ayman El Aassal. Comprehensive method for detecting phishing emailsusing correlation-based analysis and user participation. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY 2017, Scottsdale, AZ, USA, March 22-24, 2017*, pages 155–157, 2017.

[VH13] Rakesh M. Verma and Nabil Hossain. Semantic feature selection for text with application to phishing email detection. In *Information Security and Cryptology - ICISC 2013 - 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers*, pages 455–468, 2013.

[VNRK18] Anu Vazhayil, Harikrishnan NB, Vinayakumar R, and Soman KP. Ped-ml: Phishing email detection using classical machine learning techniques CENSec@Amrita. In *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA_AP)*. CEUR-WS.org, 2018.

[VR15] Rakesh M. Verma and Nirmala Rai. Phishidetector: Message-id based automatic phishing detection. In *SECRYPT 2015 - Proceedings of the 12th International Conference on Security and Cryptography, Colmar, Alsace, France, 20-22 July,2015.*, pages 427–434, 2015.

[VSH12] Rakesh Verma, Narasimha Shashidhar, and Nabil Hossain. Detecting phishing emails the natural language way. *Computer Security–ESORICS 2012*, pages 824–841, 2012.