eX^2: a framework for interactive anomaly detection

Ignacio Arnaldo
iarnaldo@patternex.com
PatternEx
San Jose, CA, USA

Mei Lam
mei@patternex.com
PatternEx
San Jose, CA, USA

Kalyan Veeramachaneni
kalyanv@mit.edu
LIDS, MIT
Cambridge, MA, USA

ABSTRACT

We introduce eX^2 (coined after explain and explore), a framework based on explainable outlier analysis and interactive recommendations that enables cybersecurity researchers to efficiently search for new attacks. We demonstrate the framework with both publicly available and real-world cybersecurity datasets, showing that eX^2 improves the detection capability of stand-alone outlier analysis methods, therefore improving the efficiency of so-called *threat hunting* activities.

CCS CONCEPTS

• **Security and privacy** → **Intrusion/anomaly detection and malware mitigation**; • **Human-centered computing** → **User interface management systems**; • **Information systems** → *Recommender systems*;

KEYWORDS

Anomaly detection; interactive machine learning; explainable machine learning; cybersecurity; recommender systems

ACM Reference Format:
Ignacio Arnaldo, Mei Lam, and Kalyan Veeramachaneni. 2019. eX^2: a framework for interactive anomaly detection. In *Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019* , 5 pages.

1 INTRODUCTION

The cybersecurity community is embracing machine learning to transition from a reactive to a predictive strategy for threat detection. At the same time, most research works at the intersection of cybersecurity and machine learning focus on building complex models for a specific detection problem [11], but rarely translate into real-world solutions. Arguably one of the biggest weakspots of these works is the use of datasets that lack generality, realism, and representativeness [3].

To break out of this situation, the first step is to devise efficient strategies to obtain representative datasets. To that end, intelligent tools and interfaces are needed to enable security researchers to carry out *threat hunting* activities, i.e., to search for attacks in real-world cybersecurity datasets. Threat hunting solutions remain vastly unexplored in the research community, and open challenges in combining the fields of outlier analysis, explainable machine learning, and recommendation systems.

In this paper, we introduce eX^2, a threat hunting framework based on interactive anomaly detection. The detection relies on outlier analysis, given that new attacks are expected to be rare and exhibit distinctive features. At the same time, special attention is dedicated to providing interpretable, actionable results for analyst consumption. Finally, the framework exploits human-data interactions to recommend the exploration of regions of the data deemed problematic by the analyst.

IUI Workshops'19, March 20, 2019, Los Angeles, USA

2 RELATED WORK

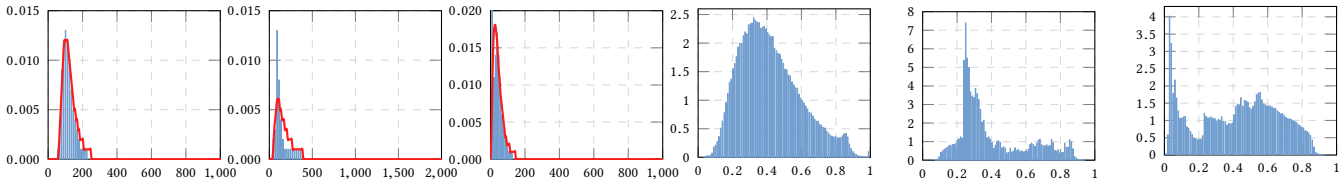
Anomaly detection methods have been extensively studied in the machine learning community [1, 6, 10]. The strategy based on Principal Component Analysis used in this work is inspired by [14], while the method introduced to retrieve feature contributions based on the analysis of feature projections into the principal components is closely related to [7].

Given the changing nature of cyber-attacks, many researchers resort to anomaly detection for threat detection. The majority of these works focus on building sophisticated models [13, 15], but do not exploit analyst interactions with the data to improve detection rates. Recent works explore a human-in-the-loop detection paradigm by leveraging a combination of outlier analysis, used to identify new threats, and supervised learning to improve detection rates over time [2, 8, 16]. However, these works do not consider two critical aspects in cybersecurity. First, they do not provide explanations for the anomalies (note that [2] provides predefined visualizations based on prior attack knowledge, but it does not account for new attacks exhibiting unique patterns). Second, neither of these works exploit interactive strategies upon the confirmation of a new attack by an analyst, therefore missing an opportunity to improve the detection recall and the label acquisition process.

3 FINDING ANOMALIES

We leverage Principal Component Analysis (PCA) to find cases that violate the correlation structure of the main bulk of the data. To detect these rare cases, we analyze the projection from original variables to the principal components' space, followed by the inverse projection (or *reconstruction*) from principal components to the original variables. If only the first principal components (the components that explain most of the variance in the data) are used for projection and reconstruction, we ensure that the reconstruction error will be low for the majority of the examples, while remaining high for outliers. This is because the first principal components explain the variance of normal cases, while last principal components explain outlier variance [1].

Let X be a p-dimensional dataset. Its covariance matrix Σ can be decomposed as: $\Sigma = P \times D \times P^T$, where P is an orthonormal matrix where the columns are the eigenvectors of Σ, and D is the diagonal matrix containing the corresponding eigenvalues $\lambda_1 \dots \lambda_p$, where the eigenvectors and their corresponding eigenvalues are sorted in



(a) Credit card (b) KDDCup (c) ATO (d) Credit card (e) KDDCup (f) ATO

Figure 1: Score distributions (a, b, c) and normalized scores (d, e, f) for three datasets obtained with the PCA method and the corresponding fitted distributions.

decreasing order of significance (the first eigenvector accounts for the most variance etc).

The projection of the dataset into the principal component space is given by $Y = XP$. Note that this projection can be performed with a reduced number of principal components. Let Y^j be the projected dataset using the top j principal components: $Y^j = X \times P^j$. In the same way, the reverse projection (from principal component space to original space) is given by $R^j = (P^j \times (Y^j)^T)^T$, where R^j is the reconstructed dataset using the top j principal components.

We define the outlier score of point $X_i = [x_{i1} \dots x_{ip}]$ as:

$$\begin{cases} score(X_i) = \sum_{j=1}^{p} (|X_i - R_i^j|) \times ev(j) \\ ev(j) = \frac{\sum_{k=1}^{j} \lambda_k}{\sum_{k=1}^{p} \lambda_k} \end{cases} \quad (1)$$

Note that $ev(j)$ represents the cumulative percentage of variance explained with the top j principal components. This means that, the higher is j, the most variance will be accounted for within the components from 1 to j. With this score definition, large deviations in the top principal components are not heavily weighted, while deviations in the last principal components are. Outliers present large deviations in the last principal components, and thus will receive high scores.

Normalizing outlier scores: As shown in Figure 1, the outlier detection method assigns a low score to most examples, and the distribution presents a long right tail. At the same time, the range of the scores depends on the datasets, which limits the method's interpretability. To overcome this situation, we project all scores into a same space, in such a way that scores can be interpreted as probabilities. To that end, we model PCA-based outlier scores with a Weibull distribution (overlaid in the figures in red). Note that the Weibull distribution is flexible and can model a wide variety of shapes. For a given score S, its outlier probability corresponds to the cumulative density function evaluated in S: $F(S) = P(X \leq S)$. Figure 1 shows the final scores F for each of the analyzed datasets. We can see that, with this technique, the final scores approximately follow a long-right tailed distribution in the $[0, 1]$ domain. Note that these scores can be interpreted as the *probability that a randomly picked example will present a lower or equal score.*

4 EXPLAINING AND EXPLORING ANOMALIES

Interpretability in machine learning can be achieved by explaining the *model* that generates the results, or by explaining each *model outcome* [9]. In this paper, we focus on the latter, given that the goal is to provide explanations for each individual anomaly. More formally, we consider an anomaly detection strategy given by $b(X^p) = S$ where b is a black-box detector, X^p is a dataset with p features, and S is the space of scores generated by the detector. The goal is to find an explanation $e \in \epsilon$ for each $x \in X^p$, where ϵ represents the domain of interpretable explanations. We approach this problem as finding a function f such that for each vector $x \in X^p$, the corresponding explanation is given by $e = f(x, b)$.

In this paper, we introduce a procedure f tailored to PCA that generates explanations $e = \{C, V\}$, where C contains the contribution of each feature to the score, and V is a set of visualizations that highlight the difference between the analyzed example and the bulk of the population.

Retrieving feature contributions: In this first step, we retrieve the contribution of each feature of the dataset to the final outlier score via *model inspection*. Note that we leverage matrix operations to simultaneously retrieve the feature contributions for all the examples; we proceed as follows:

(1) Project one feature at a time using all principal components. For feature i, the projected data is given by $Y_i = X_i \times P$, where the matrix P contains all p eigenvectors.
(2) Compute the feature contribution C_i of feature i as:

$$C_i = \sum_{j=1}^{p} Y_i^j \times ev(j) \quad (2)$$

where Y_i^j is the projected value of the i-th feature on the j-th principal component, and $ev(j)$ is the cumulative percentage of variance explained with the top j principal components given in Equation 1. In other words, the higher the absolute values projected with the last principal components, the higher the contribution of the feature to the outlier score.
(3) In a last step, we normalize the feature contributions to obtain a unit vector $\overline{C}$ for each sample:

$$\overline{C_i} = \frac{C_i}{\sum_{j=1}^{p} C_j} \quad (3)$$

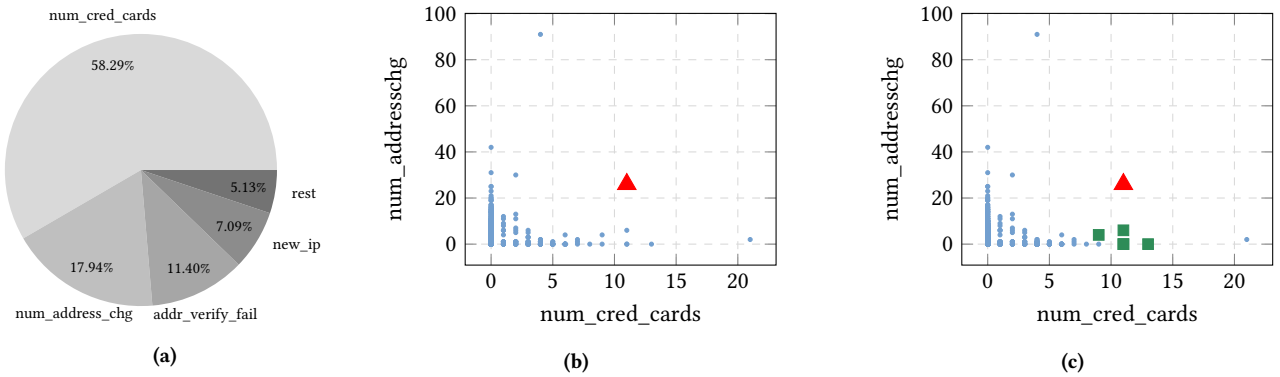


Figure 2: Explanation of an outlier (in red) of the account takeover dataset (ATO): (a) feature contributions; (b) distribution of the population in the subspace formed by the top 2 contributing features; (c) nearest neighbors (green) in the 2D subspace.

This way, for each outlier, we obtain a contribution score in the $[0, 1]$ domain for each feature in the dataset. To illustrate this step, we show in Figure 2a the feature contributions to the score of an outlier of the ATO dataset; we can see that *num_cred_cards* contributed the most to the example's score (58.29%), followed by *num_address_chg* and $addr\_verify\_fail$ (17.94% and 11.40% respectively).

Visualizing anomalies: Once the feature contributions are extracted, the system generates a series of visualizations to show each outlier in relation with the rest of the population. For ease of interpretation, these visualizations are generated in low dimensional feature subspaces as follows:

(1) Retrieve the top-m features ranked by contribution score
(2) For each pair of features (x_i, x_j) in the top-m, display the joint distribution of the population in a 2D-scatter plot as shown in Figure 2b. Note that in the example $m = 2$ and that the analyzed outlier is highlighted in red. In cases of large datasets, the visualizations are limited to 10K randomly picked samples.

With this approach, we obtain intuitive visualizations in low-dimension subspaces of the original features, in such a way that outliers are likely to *stand out* with respect to the rest of the population.

Exploring via recommendations in feature subspaces: As the analyst interacts with the visualizations and confirms relevant findings, the framework recommends the investigation of entities with similar characteristics. These recommendations are interactive and correspond to searching the top-k nearest neighbors in the feature subspaces used to visualize the data (as opposed to using all the features for distance computation). As shown in Figure 2c, the recommendations highlighted in green help narrow down the search of further anomalies.

This strategy, recommending based on similarities computed in feature subsets, exploits user interactions with the data. The intuition is that, upon confirmation of the relevance of an outlier with the provided visualizations, *the user identifies discriminant feature sets that are not known a priori*. Thus, points close to the identified anomaly in the resulting subspaces are likely to be in turn relevant.

5 EXPERIMENTAL WORK

Datasets: We evaluate the framework's capability to find, explain, and explore anomalies with four outlier detection datasets, out of which three are publicly available (WDBC, KDDCup, and Credit Card) and one is a real-world dataset built with logs generated by an online application:

- **WDBC dataset**: this dataset is composed of 367 rows, 30 numerical features, and includes 10 anomalies. We consider the version available at [5] introduced by Campos *et al.* [4]. Note that this is not a cybersecurity dataset, but has been included to cover a wider range of scenarios.
- **KDDCup 99 data (KDD)**: We consider the pre-processed version introduced in [4] in which categorical values are one-hot encoded and duplicates are eliminated. This version is composed of 48113 rows, 79 features, and counts 200 malicious anomalies.
- **Credit card dataset (CC)**: used in a Kaggle competition [12], the dataset is composed of 284807 rows, 29 numerical features, and counts 492 anomalies.
- **Account takeover dataset (ATO)**: this real-world dataset was built using web logs from an online application during three months. Each row corresponds to the summarized activity of a user during a 24 hour time window (midnight to midnight). It is composed of 317163 rows, 25 numerical features, and counts 318 identified anomalies.[1]

Detection rates and analysis of top outliers: Table 1 shows the detection metrics of the PCA-based method and Local Outlier Factor (LOF), a standard outlier analysis baseline, on each of the datasets. The detection performance of LOF is superior for the smaller dataset, WDBC. However, PCA-based outlier analysis outperforms LOF in the three cybersecurity datasets (KDD, CC, and ATO). This observation validates the choice of PCA, given that not only it outperforms LOF, but it also provides interpretability as explained in Section 4.

Despite improving the results of LOF in the cybersecurity datasets, we can see that the precision and recall metrics of the PCA-based method remain low. For instance, when looking at the top 100 outliers, the precision of our method (noted as P@100 in the table) is

[1] As most real-world datasets, ATO is not fully labeled, therefore the metrics presented in the following need to be interpreted accordingly.

Dataset	Method	AUROC	AUPR	Pr@10	R@10	P@50	R@50	P@100	R@100	P@200	R@200	P@500	R@500
WDBC	LOF	0.982	0.834	0.800	0.800	0.180	0.900	0.100	1.000	0.050	1.000	0.020	1.000
	PCA	0.899	0.219	0.300	0.300	0.160	0.800	0.090	0.900	0.050	1.000	0.020	1.000
KDDCup	LOF	0.606	0.029	0.000	0.000	0.240	0.060	0.170	0.085	0.105	0.105	0.054	0.135
	PCA	0.977	0.136	0.300	0.015	0.260	0.065	0.210	0.105	0.220	0.220	0.138	0.345
Credit card	LOF	0.654	0.015	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	PCA	0.954	0.255	0.400	0.008	0.620	0.063	0.480	0.098	0.500	0.203	0.282	0.287
Account takeover	LOF	0.568	0.004	0.000	0.000	0.020	0.003	0.010	0.003	0.005	0.003	0.004	0.006
	PCA	0.861	0.010	0.100	0.003	0.020	0.003	0.020	0.006	0.020	0.013	0.014	0.022

Table 1: Anomaly detection metrics of Local Outlier Factor (LOF) and the method based on PCA used in our framework.

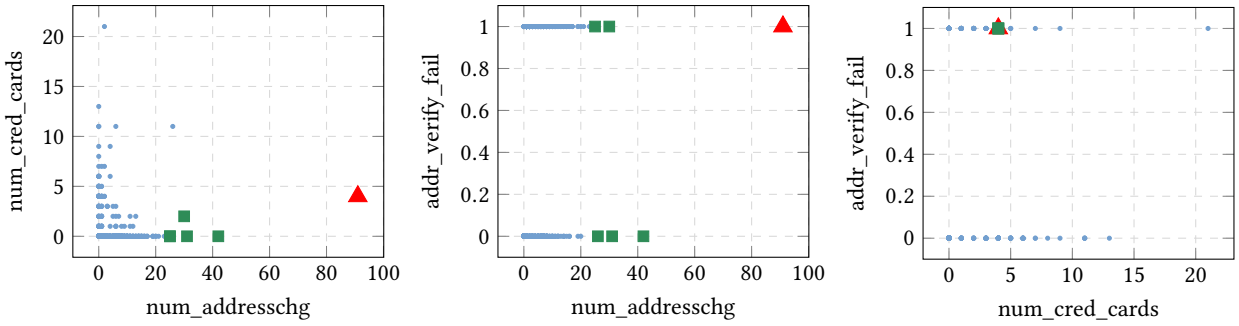


Figure 3: Visualization of the top ATO outlier with respect to the bulk of the population in 2D feature subspaces of interest. The recommendations performed by the system are shown in green.

0.210, 0.480, and 0.020 for KDDcup, CC, and ATO respectively. This observation indicates that *not all outliers are malicious*, and justifies the effort dedicated to providing interactive exploration of the data to increase anomaly detection rates.

Explain and explore: We show in Figure 3 the visualizations and recommendations generated for the top ATO outlier. The framework appropriately selects feature subsets such that the analyzed outlier (shown in red) stands out with respect to the population (blue), *ie* outliers fall in sparse regions of the selected subspaces. The top 3 contributing features retrieved by the framework are the number of address changes (`num_addresschg`), the number of credit cards used (`num_cred_cards`), and whether the user failed the address verification (`addr_verify_fail`). In the first plot (`num_addresschg` vs `num_cred_cards`), we can clearly see why the highlighted user is suspicious: he/she used four credit cards, and changed the delivery address more that 90 times. The plot also shows five additional users recommended by the system upon confirmation of the threat by an analyst. The recommended users present an elevated number of address changes, and used one or more credit cards.

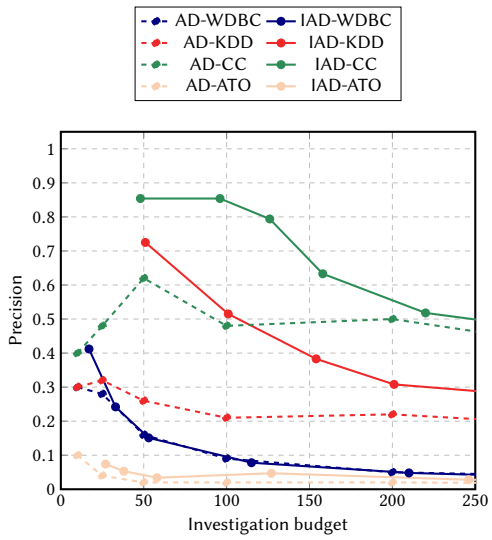


Figure 4: Precision versus investigation budget of anomaly detection alone based on PCA (AD) and interactive anomaly detection combining both PCA and recommendations (IAD).

To further evaluate the exploratory strategy based on recommendations, Figure 4 shows the detection rate obtained with PCA alone, versus the metrics obtained with the combination of PCA and recommendations. To obtain the latter metrics, we simulate investigations for the top-m ($m \in [10, 25, 50, 100, 200, 500]$) outliers (ie we reveal the ground truth) and consider the top-10 recommended entries for the confirmed threats. In all cases, interactive anomaly detection improves the precision. In particular, we can see a significant precision improvement for the KDD and CC datasets for investigation budgets in the 50-200 range.

6 CONCLUSION

We have introduced the eX^2 framework for *threat hunting* activities. The framework leverages principal component analysis to generate interpretable anomalies, and exploits analyst-data interaction to recommend the exploration of problematic regions of the data. The results presented in this work with three cybersecurity datasets show that eX^2 outperforms detection strategies based on standalone outlier analyis.

REFERENCES

[1] Charu C. Aggarwal. 2013. *Outlier Analysis*. Springer. https://doi.org/10.1007/978-1-4614-6396-2

[2] Anaël Beaugnon, Pierre Chifflier, and Francis Bach. 2017. ILAB: An Interactive Labelling Strategy for Intrusion Detection. In *RAID 2017: Research in Attacks, Intrusions and Defenses*. Atlanta, United States. https://hal.archives-ouvertes.fr/hal-01636299

[3] E. Biglar Beigi, H. Hadian Jazi, N. Stakhanova, and A. A. Ghorbani. 2014. Towards effective feature selection in machine learning-based botnet detection approaches. In *2014 IEEE Conference on Communications and Network Security*. 247–255.

[4] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery* 30, 4 (01 Jul 2016), 891–927.

[5] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. 2018. Datasets for the evaluation of unsupervised outlier detection. www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (July 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

[7] XuanHong Dang, Barbora MicenkovÃą, Ira Assent, and RaymondT. Ng. 2013. Local Outlier Detection with Interpretation. In *Machine Learning and Knowledge Discovery in Databases*, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Å¡elezný (Eds.). Lecture Notes in Computer Science, Vol. 8190. Springer Berlin Heidelberg, 304–320. https://doi.org/10.1007/978-3-642-40994-3_20

[8] S. Das, W. Wong, T. Dietterich, A. Fern, and A. Emmott. 2016. Incorporating Expert Feedback into Active Anomaly Discovery. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 853–858. https://doi.org/10.1109/ICDM.2016.0102

[9] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 5, Article 93 (Aug. 2018), 42 pages. https://doi.org/10.1145/3236009

[10] Victoria Hodge and Jim Austin. 2004. A Survey of Outlier Detection Methodologies. *Artif. Intell. Rev.* 22, 2 (Oct. 2004), 85–126. https://doi.org/10.1023/B:AIRE.0000045502.10941.a9

[11] Heju Jiang, Jasvir Nagra, and Parvez Ahammad. 2016. SoK: Applying Machine Learning in Security-A Survey. *arXiv preprint arXiv:1611.03186* (2016).

[12] Kaggle. 2018. Credit Card Fraud Detection Dataset. www.kaggle.com/isaikumar/creditcardfraud

[13] Benjamin J. Radford, Leonardo M. Apolonio, Antonio J. Trias, and Jim A. Simpson. 2018. Network Traffic Anomaly Detection Using Recurrent Neural Networks. *CoRR* abs/1803.10769 (2018). arXiv:1803.10769 http://arxiv.org/abs/1803.10769

[14] Mei-ling Shyu, Shu ching Chen, Kanoksri Sarinnapakorn, and Liwu Chang. 2003. A novel anomaly detection scheme based on principal component classifier. In *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDMâĂŹ03*. 172–179.

[15] Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson. 2017. Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams. *CoRR* abs/1710.00811 (2017). arXiv:1710.00811 http://arxiv.org/abs/1710.00811

[16] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li. 2016. AI^2: Training a Big Data Machine to Defend. In *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity)*. 49–54.