Uncovering Unknown Unknowns in Financial Services Big Data by Unsupervised Methodologies: Present and Future trends

Gil Shabat GIL.SHABAT@THETARAY.COM
ThetaRay Ltd. [*]

David Segev DAVID.SEGEV@THETARAY.COM
ThetaRay Ltd.

Amir Averbuch AMIR.AVERBUCH@THETARAY.COM
ThetaRay Ltd.
School of Computer Science, Tel Aviv University

Abstract

Currently, unknown unknowns in high dimensional big data environments can go unnoticed for a long period of time. The failure to detect anomalies in critical infrastructure data can result in extensive financial, operational, reputational and life threatening consequences. In this paper, we describe algorithms for an automatic and unsupervised anomaly detection that do not necessitate domain expertise, signatures, rules, patterns or semantics understanding of the features. We propose several new methodologies for anomaly detection to protect critical infrastructures, with emphasis on finance, spanning from theory to actionable technology. Although anomalies can originate from several sources, we also show that cyber threat, financial and operational malfunction are converging into a single detection paradigm. Performance comparison between different algorithms (ours and others) is presented as well as examples from real use cases.

1. Introduction

In the last decade, with the overwhelming increase of interest in big data analytics, the demand for data driven methods for anomaly detection rose substantially. For example, anomaly detection that operates on high dimensional big data (HDBD) is of fundamental importance in cyber security for protecting critical energy infrastructure, transportation, financial institutions, or telecommunications corporations. Moreover, to find new and unseen types of threats and disturbances including advanced financial threats, unsupervised approaches are required. However, several factors make unsupervised anomaly detection in HDBD a challenging task: the need to learn the distributions of high dimensional data points, frequently loosely defined boundaries between normal and abnormal behavior, time evolving data normality, *i.e.* what is currently considered as a normal behavior might be abnormal in future and vice versa. In addition, unsupervised anomaly detection closely depends on well engineered features, which greatly relies on highly skilled domain experts,

[*] 8 Hanagar street, Hod Hasharon 4501309, Israel

on data integrity concern, null completion, codes vs. measurements handling and so on. Failure to correctly address these issues could result in high false alarm rate or potential financial lost.

In this paper, we focus on an automatic and unsupervised anomaly detection in a structured HDBD that do not necessitate domain expertise, signatures, rules, patterns or semantics understanding of the features. We propose several new methodologies for anomaly detection to protect critical infrastructures, with emphasis on finance, spanning from theory to actionable technology. Although anomalies can be originated from several sources, we also show that cyber threat, financial and operational malfunction are converging into a single detection paradigm.

The basic approach in securing critical infrastructures in the past 45 years, classified as "walls and gates", has failed. There is no reason to think that barriers between trusted and untrusted components with policy-mediated pass-through systems, will be more successful in the future. Rule based detection methodologies, which govern firewalls, signatures/patterns that govern IDS/IPS and antivirus, are irrelevant today for detection of new and sophisticated malware (virus, worms, back door, spyware, Trojans) masked as a legitimate stream and penetrate every state-of-the-art commercial barrier on the market. Traditional defense systems are ineffective. These systems do not catch zero-day attacks and Advanced Persistent Threats (APT), which do not have previously encountered signatures or play by known rules. In other words, traditional solutions cannot detect unknown unknowns; they are only able to detect yesterday's attacks - ones encountered in the past, and those they know they are looking for.

Currently, the time it takes to detect unknown unknowns in HDBD environments can go unnoticed for months and even more. The failure to detect anomalies in critical infrastructure data can result in extensive financial, operational, reputational and life threatening consequences.

In the flood of data (40 Trillion GB is an estimation of the size of the digital universe by 2020 up from 130 billion in 2005[1]) lie tremendous opportunities for us to understand, process, manipulate and extract intelligence from it, visualize it, connect the dots between pieces of information and turn HDBD into meaningful insight.

Anomaly detection is the process of identifying unexpected items or events in datasets which differ from the norm. When the dimension of the data is getting larger, anomaly detection is becoming more challenging as each feature can appear normal by itself, but abnormal when combined with other features. Having large datasets, with the appropriate algorithms, will enable to learn normal and abnormal combinations of feature values. Though deep learning plays a central role in our detection scheme, the scope of this paper is limited to machine learning algorithms that do not rely on the deep learning approach. Performance comparison between different algorithms is presented in section 4.1, as well as examples of two real use cases (sections 4.3 and 4.4).

2. Processing Ideologies of HDBD and Governing Principles for Unsupervised Anomaly Detection

Anomaly detection in HDBD can be classified into 4 modes-of-operation:

[1] IDC IVIEW https://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf

Unsupervised (unknown unknowns): There is no knowledge about the data and no labels to classify each member in the dataset into predefined categories. This is the focus of this paper.

Guided Unsupervised: We have some knowledge about the data and part of its labels. In this case, unsupervised approach becomes more tailored to this specific knowledge, which is incorporated in the learning process through label-based feature engineering.

Semi-Supervised: We have full knowledge on a small amount of labeled data (classified) but we do not know how the complementary data is classified.

Supervised: We have full knowledge of how each member of the dataset is classified into predefined categories.

Our unsupervised anomaly detection process relies on the following principles: Uncovering unknown unknowns without the need for domain expertise, signatures, rules, patterns, heuristics, supervision and semantics understanding. There is no limitation on the number of features (dimension) in the data. The same core technology, classified as a universal core, supports financial, cyber and industrial verticals without any modification/adaption to a specific vertical. Emphasis is on efficient processing to get fast and automatic detection in HDBD with low false positive rate.

3. Different Types of Algorithms for Unsupervised Processing

The algorithms are divided into three main categories: Preprocessing - deals mainly with data preparation, core algorithms - perform the genuine anomaly detection in the data, and post-processing - for organizing the results displayed to the user. Preprocessing algorithms include among others: Categorical to numerical conversion Udell et al. (2016), missing values replacement Shabat and Averbuch (2012); Udell et al. (2016), normalization, integrity check and automatic feature adaptation. "Core"-detection algorithms include among others: geometry extraction (Diffusion maps, Kernels), low rank approximation, density estimation, dictionary learning, neural nets and deep learning. Post-processing algorithms include among others: threshold calculation for anomaly detection, fusion of the results obtained from several algorithms, parameter rating, anomaly clustering and similarity search.

In this paper, two algorithms are described: Geometry-based extraction called Diffusion Maps (DM) Coifman and Lafon (2006a); Lafon (2004) or Nyström detector, and computationally efficient randomized low rank matrix decomposition. The algorithms do not require the data set to be completely anomaly free and can detect anomalies in the training set itself as long as they are rare.

3.1. Diffusion Maps (DM): Overview

DM methodology is based on constructing a Markovian diffusion process that quantifies the connectivity between multidimensional data points and follows the dominant geometric patterns in the data. DM embedding is obtained by spectral analysis of this process (i.e., eigenvalues and eigenvectors of its transition operator). This embedding usually provides a faithful low-dimensional representation of patterns and trends in the analyzed data. To classify each newly arrived multidimensional data point in online mode, we use the Nyström extension (see Lafon (2004); Coifman and Lafon (2006a)).

Let $M \subseteq \mathbb{R}^m$ be a dataset that is sampled from a manifold $\mathcal{M}$ that lies in the ambient space $\mathbb{R}^m$. Let $d \ll m$ be the intrinsic dimension of $\mathcal{M}$, thus, it has a d-dimensional tangent space $T_x(\mathcal{M})$, at every point $x \in M$. If the manifold is densely sampled, the tangent space $T_x(\mathcal{M})$ can be approximated by a small enough neighborhood around $x \in M$.

DM analyzes the dataset M by exploring the geometry of the manifold $\mathcal{M}$ from which it is sampled. This method is based on defining an isotropic kernel operator $Kf(x) = \int_M k(x,y)f(y)dy$ (for $f : M \to \mathbb{R}$) that consists of the affinities $k(x,y)$, $x,y \in M$. The affinities in this kernel represent similarity, or proximity, between data points in the dataset and on the manifold. The kernel K can be viewed as a construction of a weighted graph over the dataset M. The points in M are used as vertices in this graph and the weights of the edges are defined by the affinities in K.

These affinities in K are assumed to satisfy the following properties: Each data-point has a positive self-affinity, the affinities are non-negative and symmetric. The graph, which is defined by the weighted adjacencies in K, is connected.

The affinity kernel is the Gaussian kernel

$$k(x,y) \triangleq e^{-\frac{\|x-y\|^2}{\epsilon}}, \tag{1}$$

where $x,y \in M$ and $\varepsilon > 0$. This kernel is also used in other dimensionality reduction methods (e.g., Laplacian Eigenmaps Belkin and Niyogi (2003)) as well as out-of-sample extension methods (e.g., Geometric Harmonics Coifman and Lafon (2006b) and MSE Bermanis et al. (2013)).

The graph K represents the intrinsic structure of the manifold and is used by DM to construct a Markovian (random-walk) diffusion process that follows it. The degree of each data point (i.e., vertex) $x \in M$ in this graph is defined as $q(x) \triangleq \int_M k(x,y)dy$. The diffusion process is defined by normalizing the kernel K with $q(x)$ to obtain the transition probabilities $p(x,y) \triangleq k(x,y)/q(x),\ x \in M,\ y \in M$. These probabilities constitute the row stochastic transition operator $Pf = \int_M p(x,y)f(y)dy$ of the diffusion process.

DM computes an embedding of data points on the manifold into a Euclidean space whose dimensionality is usually significantly lower than the original data dimensionality. This embedding is a result of spectral analysis of the diffusion kernel.

Under mild conditions on the kernel K the resulting diffusion affinity kernel has a discrete decaying spectrum of eigenvalues $1 = \lambda_0 \geq |\lambda_1| \geq |\lambda_2| \geq \ldots$ (see Lafon (2004)). These eigenvalues are used in DM together with their corresponding eigenvectors $\mathbf{1} = \phi_0, \phi_1, \phi_2, \ldots$ to define the DM embedding of the data. Each data point $x \in M$ is embedded by DM to the diffusion coordinates that are given by $\varPhi(x) \triangleq (q^{-1}(x)\lambda_1^t\phi_1(x), \ldots, q^{-1}(x)\lambda_\delta^t\phi_\delta(x))$, where t is the diffusion time parameter and the exact value of δ depends on the kernel's spectrum. In most cases, δ is significantly smaller than the original dimensionality of the observable data.

As a result of the spectral theorem, the Euclidean distances in the embedded space correspond to the diffusion distance metric of the diffusion process defined by P (Coifman and Lafon (2006a); Lafon (2004)). This metric quantifies the connectivity between data points in the diffusion process by considering the several possible diffusion paths between them within a time step t. Therefore, the resulting embedded space of DM follows the geometry that is defined by the underlying diffusion process of DM. When the data points are sampled from a low-dimensional manifold, this diffusion geometry reveals the intrinsic structure of the underlying manifold and the DM embedding provides a meaningful representation of the data. Additional analysis techniques can then be applied to the embedded space to perform common learning tasks, such as clustering, classification and anomaly detection (see David (2009); Rabin (2010)).

3.1.1. Extension schemes for newly arrived multidimensional data points

A newly arrived multidimensional data point is embedded by several possible techniques into a small subspace such as manifold, as proposed in Kaymaz (2005); Rabin and Fishelov (2017), or using a multi-scale scheme Bermanis et al. (2013), Interpolative Decomposition Cheng et al. (2005) or Nyström extension.

Nyström extension finds a numerical approximation for the continuous eigenfunction problem

$$\int_a^b G(x,y)\phi(y)dy = \lambda\phi(x).$$

Discretization is achieved by $\frac{1}{n}\sum_{j=1}^{n} G(x_i,x_j)\phi(x_j) = \lambda\phi(x_i)$. The Nyström extension of ϕ to a new data point x is given by

$$\hat{\phi}(x) \triangleq \frac{1}{n\lambda}\sum_{j=1}^{n} G(x,x_j)\phi(x_j).$$

In our setup, G is a Gaussian kernel matrix, $g_{ij} = e^{\frac{-\|x_i-x_j\|^2}{\epsilon}}$. The eigenfunctions of G constitute an orthonormal basis for $\mathbb{R}^n$. Any vector $f^T = (f(x_1), f(x_2), \dots, f(x_n))$ can be decomposed according to $f = \sum_{i=1}^{n}\langle f,\phi_i\rangle\phi_i$. f is extended to x by $\hat{f}(x) = \sum_{i=1}^{n}\langle f,\phi_i\rangle\hat{\phi}_i(x)$. It is important to mention that the direct computation of the kernel for n data points in $\mathbb{R}^d$ takes $\mathcal{O}(n^2 d)$ operations, and an additional $\mathcal{O}(n^2 k)$ operations for computing its rank k SVD. It takes also $\mathcal{O}(n^2)$ in memory for storing the kernel. These computational requirements make it impractical for processing large data. In order to cope with this problem, random Fourier features Rahimi and Recht (2008) are used with batch computing to reduce memory requirements and computational complexity. As described in Yu et al. (2016), by using structured random projections based on fast Hadamard transform, the computational complexity can be reduced to $\mathcal{O}(nd\log d)$. Incorporating the fast structured random Fourier features, reduces the computational complexity to linear in the number of data points and enable the algorithm to process large data on a GPU with a limited memory amount.

Eventually, the anomalies are detected in the following way: The training data is projected into a manifold according to the DM methodology. Newly arrived multidimensional data points are projected onto the manifold using out-of-sample extension methodologies. If the newly arrived multidimensional data point falls inside the manifold it is classified as normal, otherwise it is considered an anomaly. The decision of point is an anomaly is defined by computing its score, based on a nearest-neighbor-based density estimation. The threshold is estimated by performing a model fitting to a probability density function.

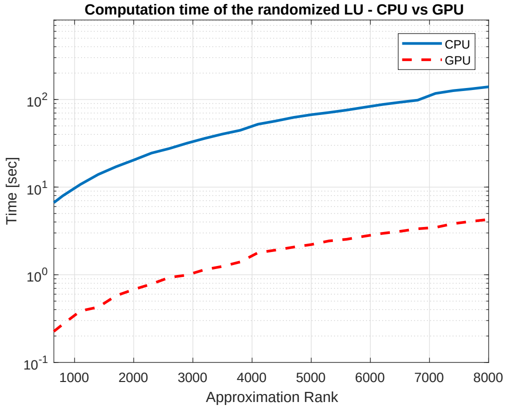


Figure 1: Running time comparison between CPU and GPU

3.2. Low Rank Approximations for Anomaly Detection

Low rank approximations play a central role in machine learning and specifically in anomaly detection. In this paper we propose an approach for low rank approximations in anomaly detection by finding a linear subspace that captures only the normal data points and not the anomalies. Each data point gets a score that reflects how well it is spanned by the subspace. In order to compute efficiently a low rank approximation in an LU form of a given matrix, it is possible to use the Randomized LU decomposition Shabat et al. (2016), which is a fast matrix factorization method that uses random projections. This approach enables the derivation of theoretical error bounds, and can be fully parallelized to run on a GPU without any CPU-GPU data transfer. The method can be further accelerated by using sparse random projections Clarkson and Woodruff (2013), as described in Aizenbud et al. (2016).

Given an input $m \times n$ matrix A, a desired rank k and the number of random projections l, the randomized LU returns the matrices L, U, P, Q, where L and U are lower and upper trapezoidal matrices respectively, and P, Q are orthogonal permutation matrices such that

$$\|LU - PAQ\|_2 \leq C(m, n, l, k)\sigma_{k+1},$$

where σ_{k+1} is the $k+1$ largest singular value of A and the factor $C(m, n, l, k)$ appears explicitly in Shabat et al. (2016) along with the corresponding success probability. Estimating the rank k of the data can be performed using a variety of models (see, for example Gavish and Donoho (2014); Kritchman and Nadler (2008)). Algorithm 1 describes the randomized LU decomposition. Figure 1 shows a comparison between the CPU and GPU running time on a $20,000 \times 20,000$ matrix for different low rank approximations (indicated in the algorithm by "k") in single precision. The number of random projections was $l = k + 5$. The GPU used was GTX-1080 and the CPU was Intel 6950X running with 10 cores.

3.2.1. Building Dictionaries for Classification/Detection

Following algorithm 1 it can be shown that $D := P^T L$ approximates the range of A and can therefore be used as a dictionary. The approximation error for the range is given by

Algorithm 1 Randomized LU Decomposition

```
Input: A matrix to decompose of size $m \times n$; $k$ desired rank; $l$ number of columns to use.
Output: Matrices $P, Q, L, U$ such that $PAQ = LU$ where $P, Q$ are permutation matrices and $L, U$ are lower and upper triangular matrices, respectively.
Create a Gaussian random matrix $G$ of size $n \times l$.
$Y \leftarrow AG$
Perform RRLU decomposition Pan (2000) to $Y$, such that $PYQ_y = L_y U_y$.
Choose a basis from $L_y$ by picking the first $k$ columns, $L_y \leftarrow L_y(:, 1 : k)$.
$B \leftarrow L_y^\dagger PA$ /*$\dagger$ is the pseudo inverse.
Perform LU decomposition to $B$ with column pivoting $BQ = L_b U_b$.
$L \leftarrow L_y L_b$.
$U \leftarrow U_b$.
```

$$\|DD^\dagger A - A\| \leq C(m, n, l, k)\sigma_{k+1}.$$

When high accuracy is achieved, for example by using power iterations Martinsson et al. (2010); Li et al. (2017), the error becomes $\|DD^\dagger A - A\| \approx \sigma_{k+1}(A)$.

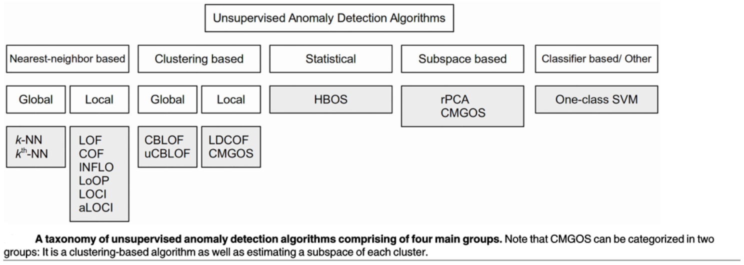
During the training phase, the dictionary D is computed according to algorithm 1. The score of a newly arrived measurement x is then determined by the approximation error $\|DD^\dagger x - x\|$. Therefore, points with low score are by definition well represented by the dictionary D and are similar to normal data, while anomalies are susceptible to have high scores. In practice, due to the inherent random nature of the algorithm, the dictionary learning procedure is performed several times and a representative span of range A is derived.

4. Results

4.1. Accuracy comparison with other algorithms

In this section we present the detection performance of the two algorithms (geometry and dictionary) described above and compare it to the results obtained from other unsupervised anomaly detection algorithms reported in Goldstein and Uchida (2016). The algorithms described in Fig. 2 were applied to 10 datasets (both algorithms and datasets were adapted from Goldstein and Uchida (2016)): Breast cancer Wisconsin (Diagnostic), Pen-Based recognition of handwritten text (global), Pen-Based recognition of handwritten text (local), letter recognition, speech accent data, landsat satellitem, thyroid disease, Statlog shuttle, Object images (ALOI) and KDD-Cup00 HTTP. All the datasets contain different features for a variety of anomaly detection tasks.

We adopt the AUC (area under the curve) as a measure to assess the accuracy of our anomaly detection algorithms. We choose then to compute the average AUC on all the 10 datasets considered to address consistency of the performance across all the datasets. Figure 3 depicts the average accuracy over all the data sets reported for our algorithms and for some of the most accurate algorithms reported in Goldstein and Uchida (2016). It



A taxonomy of unsupervised anomaly detection algorithms comprising of four main groups. Note that CMGOS can be categorized in two groups: It is a clustering-based algorithm as well as estimating a subspace of each cluster.

Figure 2: Unsupervised anomaly detection algorithms (taken from Goldstein and Uchida (2016))

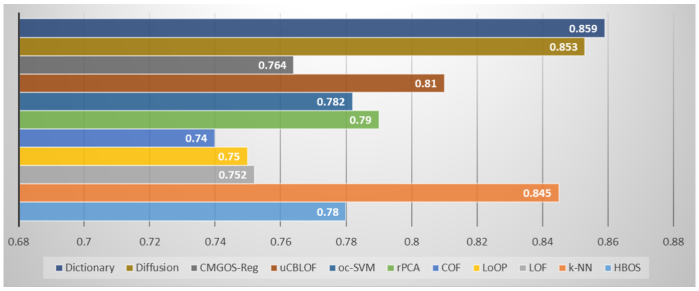


Figure 3: Average accuracy of a variety of algorithms applied to several datasets described in the text.

can be seen that the nearest neighbor-based (k-NN) and the clustering-based (uCBLOF) algorithms are the two most accurate algorithms reported in Goldstein and Uchida (2016). However, both the geometry and dictionary detectors we present are superior to all the algorithms that appear in Fig. 3.

4.2. Computation time comparison with other algorithms

Besides the high anomaly detection accuracy required, algorithms should also comply with good computational performance. We compile in Fig. 4 the computation time (in log scale) of our diffusion and dictionary algorithms on all the datasets described in the precedent section. It can be seen that it takes less than 1 sec for our dictionary-based algorithm to find anomalies in most of the datasets considered, with a very favorable speed scaling when moving from small datasets (thyroid for example) to large ones (aloi or kdd99 for example). A very good computation time is also reached by our diffusion algorithm, which is a kernel based method that usually leads to very poor speed performance. Even the largest dataset (kdd99) that contains $620,098$ rows and 29 columns is analyzed within only 41 sec on a 4-core CPU laptop with a i7-7700HQ processor.

Although a direct comparison of our speed performance and that of the algorithms in Goldstein and Uchida (2016) is obviously hazardous and greatly depends on the hardware used, we can still present a comparison of the computation speed scale up when moving from the thyroid dataset made of 6916 rows, to the aloi dataset with its 50000 rows up to the kdd99 dataset. All these are composed of close to 25 features, which enables us to directly

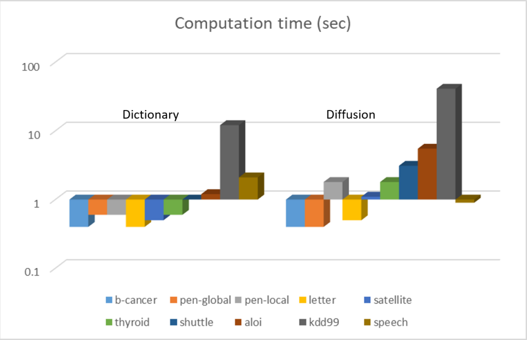


Figure 4: Computation time (in sec) for our two algorithms applied to all datasets described in the text.

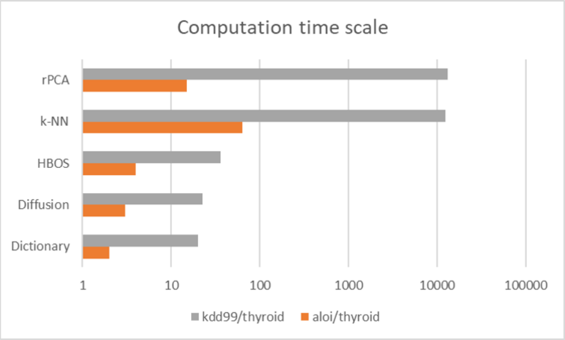


Figure 5: Computation time scale for five different algorithms applied to aloi and kdd99 relative to the thyroid dataset.

compare the increase in the computation speed as a function of the number of rows in the dataset. The results are shown in Fig. 5. The computation time scale presented in Fig 5 for aloi and kdd99 are relative to the thyroid dataset. Only the fastest three algorithms appearing in Goldstein and Uchida (2016) are shown for comparison. It should be noted that the computation time of the HBOS and rPCA algorithms for the thyroid dataset is reported to be less than 0.1 sec. We therefore set it to 0.1 sec, which results in a comparison which is not favorable to us. It can be seen that with increasing dataset size our algorithms perform much better than even the fastest algorithms we compare with that are reported in Fig. 2.

4.3. ATM Security: A real case

In this section, we briefly present a real use case encountered in our joint work with one of our customers. The customer is the ATM monitoring and security department of a top international bank, headquartered in the US. The bank suffers from high losses due to new fraud scenarios undetected by legacy technology and from an increasing instances of security breaches. The data was collected from 2, 520 ATM machines in north America and consisted of 14, 000, 000 ATM transactions logs and account information. The project was performed within 9 days on site. The algorithms detected unknown events, including suspected fraud, abnormal activity and operational deficiencies. Our approach provided also useful pin-point forensic information and transformed weeks of investigation into hours. In addition to unknown fraudulent activity, operational malfunctions and potential security breaches that were detected, we also discovered data integrity issues which had a great impact on the customer operations.

4.4. Anti-Money Laundering: A real case

In another real scenario, the goal was to detect cases of anti-money laundering (AML). The customer was the compliance and AML department of a European multinational bank. The customer had to challenge late detection of criminal activity, low accuracy and analyst alert fatigue. The data contain 50, 000, 000 transactions gathered during 18 months, in addition to account information such as balance and transfers, and socio-demographic profiles. Our algorithms detected ML and terror funding (TF) cases even before the case file was opened. We were also able to cluster all the alerts into 23 clusters and 20 single anomalies, substantially reducing the number of false alarms and of events to investigate. In addition, we discovered new instances of money laundering 11 months prior to the regulatory notification, and which were not detected by existing controls. We also uncovered known money laundering cases on average 70 days prior to existing controls, unknown ML and TF events that were not detected by existing systems and regulators. Finally, we also increased forensic and detection efficiency by reducing by 40% the number of alerts and by establishing a priority on clustered alerts.

5. Conclusions and Future Work

Currently, the time it takes to detect unknown unknowns in HDBD environments can long, in some cases more than months. The failure to detect anomalies in critical infrastructure data can result in extensive financial, operational, reputational and life threatening consequences. The paper describes automatic unsupervised anomaly detection algorithms that uncover unknown unknowns without the need to have domain expertise, signatures, rules, patterns, heuristics, supervision and semantics understanding. The same core technology, classified as a universal core, supports financial, cyber and industrial verticals without any modification/adaption to a specific vertical. Emphasis was channeled to efficient processing to get fast detection in HDBD with low false positive rate.

Anomaly detection methodology is divided into three main categories: Preprocessing - deals mainly with data preparation, core algorithms - perform the genuine anomaly detection in the data, and post-processing - for organizing the results displayed to the user. Each category has open research and developments challenges for the future such as: automatic feature selection, adaptation and manipulation in preprocessing category, neural nets and deep learning with low false positive in “core” detection algorithms, threshold calculation for anomaly detection, fusion of the results obtained from several algorithms, anomaly clustering and visualization to support fast forensic in post-processing algorithms.

Acknowledgement The authors would like to thank Sergiu Lupu and Eyal Segev for fruitful and helpful discussions.

References

Y. Aizenbud, G. Shabat, and A. Averbuch. Randomized LU decomposition using sparse projections. *Computers & Mathematics with Applications*, 72(9):2525–2534, 2016.

M. Belkin and P. Niyogi. Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

A. Bermanis, A. Averbuch, and R.R. Coifman. Multiscale data sampling and function extension. *Applied and Computational Harmonic Analysis*, 34:15 – 29, 2013.

H. Cheng, Z. Gimbutas, P.G. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM Journal on Scientific Computing*, 26(4):1389–1404, 2005.

K.L. Clarkson and D. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 81–90. ACM, 2013.

R.R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006a.

R.R. Coifman and S. Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis*, 21(1): 31–52, 2006b.

G. David. *Anomaly Detection and Classification via Diffusion Processes in Hyper-Networks.* PhD thesis, School of Computer Science, Tel Aviv University, March 2009.

M. Gavish and D.L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

I. Kaymaz. Application of Kriging method to structural reliability problems. *Structural Safety*, 27(2):133–151, 2005.

S. Kritchman and B. Nadler. Determining the number of components in a factor model from limited noisy data. *Chemometrics and Intelligent Laboratory Systems*, 94(1):19–32, 2008.

S. Lafon. *Diffusion Maps and Geometric Harmonics.* PhD thesis, Yale University, May 2004.

H. Li, G.C. Linderman, A. Szlam, K.P. Stanton, Y. Kluger, and M. Tygert. Algorithm 971: an implementation of a randomized algorithm for principal component analysis. *ACM Transactions on Mathematical Software (TOMS)*, 43(3):28, 2017.

P.G. Martinsson, A. Szlam, and M. Tygert. Normalized power iterations for the computation of SVD. *Manuscript., Nov*, 2010.

C-T Pan. On the existence and computation of rank-revealing lu factorizations. *Linear Algebra and its Applications*, 316(1-3):199–222, 2000.

N. Rabin. *Data Mining Dynamically Evolving Systems via Diffusion Methodologies.* PhD thesis, School of Computer Science, Tel Aviv University, April 2010.

N. Rabin and D. Fishelov. Multi-scale kernels for Nyström based extension schemes. *Applied Mathematics and Computation*, 2017.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

G. Shabat and A. Averbuch. Interest zone matrix approximation. *Electronic Journal of Linear Algebra*, 23(1):50, 2012.

G. Shabat, Y. Shmueli, Y. Aizenbud, and A. Averbuch. Randomized LU decomposition. *Applied and Computational Harmonic Analysis*, 2016.

M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016.

F.X. Yu, A.T. Suresh, K.M. Choromanski, D.N. Holtmann-Rice, and S. Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016.