LA-UR-21-30256
Approved for public release; distribution is unlimited.

Title: Tensor Text-Mining Methods for Malware Identification and Detection, Malware Dynamics Characterization, and Hosts Ranking

Author(s): Alexandrov, Boian
Eren, Maksim Ekin

Intended for: Report

Issued: 2021-10-15

Tensor Text-Mining Methods for Malware Identification and Detection, Malware Dynamics Characterization, and Hosts Ranking

Maksim E. Eren and Boian S. Alexandrov

October 11, 2021

Abstract

Malware is one of the most persistent and costly cyber threats endangering reputation, confidentiality, integrity, and availability for organizations and national security. Consequently, many of the incident detection and prevention systems, and incident responders have begun to utilize machine learning as a helper in the fight against malware and other cyber threats. However, cyber defenders rely on interpretability and generalizability, yet the popular machine learning methods are black-box and often use traditional supervised solutions that do not generalize to novel malware. Therefore, there is a need to improve the existing solutions. At the same time, the majority of the prior research ignored essential evaluation criteria when reporting the results of their methods, which disables the safe reproducibility of the methods in a production environment. Tensor decomposition, on the other hand, enables interpretable unsupervised analysis of the large-scale data for the discovery of hidden patterns. Our findings, performed on real-world and large-scale experiments, show that tensor factorization-based methods yield performance results that surpasses or competes with existing supervised solutions with the added benefit of interpretability and generalizability. With the ability to analyse complex and large-scale data using tensors, we report results that reflect real-world production environments. We propose to develop new game-changing tools for malware identification and characterization that can trace malware evolution, rank the infected or malicious hosts, and streamline the work of incident response teams, malware analysts, and incident detection and prevention systems.

1 Motivation

Malware continues to be one of the most prominent threats to organizations, the public, and national security. The recent cyber reports point out that malware is one of the most frequent cyber threats with the highest risk factor and costly incident resolution [5]. A single malware attack, on average, costs $2.6 million, and average cost of a ransomware breach is $4.62 million in 2021 for the organizations in the USA [5, 13]. A data breach caused by a malware as the attack vector also places personally identifiable information (PII) including employee and customer data, intellectual property, and sensitive and classified information under risk. Manual analysis of malware by reverse engineers often does not scale in production systems due to the ever-growing amount of malware in the wild, and the increasing amount and complexity of the attacks. On average, 450,000 new malware is reported daily [20]. Total malware in the wild has increased by 1178% past 10 years, and malware attacks have increased by 11% in 2019 [5]. Basic incident detection and prevention systems rely on the signatures of the known malware, often obtained from open-source intelligence (OSINT) or malware reputation sources. However, signature-based security systems can be bypassed by malware authors that continuously modify their code [14]. Therefore, organization and anti-virus (AV) vendors have begun to utilize Machine learning (ML) and Artificial Intelligence (AI) based automated security systems to combat against malware [15, 19, 18, 14]. While AI and ML based automation save up to $3.81 million on data breaches, only about 25% of the organizations has fully-deployed automated incident detection and prevention systems [13]. Thus, there is a need to improve current solutions and their deployability in production systems.

The alerts generated from ML-based security systems need to be verified by human analysts; therefore, the interpretability of the results is essential for security systems. However, many popular

ML models used to identify malware are black-box. These black-box models are often supervised technologies with the need of an immense amount of labeled data during training to achieve good performance. A large amount of up-to-date, or production quality, labeled malware is expensive to obtain. Also, supervised methods need to be updated regularly as they poorly generalize to zero-day, or novel, malware that carries different characteristics than the model saw during the training time. To this end, supervised ML models suffer the same issue as the basic signature-based security systems where the malware authors obfuscate or modify the code and behaviour of the malware to bypass these systems [14], making the evasive actions performed by the adversaries part of the cyber kill chain [17]. Due to the growing importance, prior research has extensively studied the ML solutions to malware identification and characterization. However, the majority of the prior research for ML-based malware analysis had failed to include core evaluation criteria in their work for the past two decades [16]. For example, the majority of ML solutions for malware family characterization are unrealistically limited to identify, for instance, the top most populous families. This resulted in misleading high-performance metrics that do not generalize to the production environment, as they have been limited to the analysis of easiest malware. With the ever-growing quantity of malware, attacks, and their complexities there is an urgent need to improve existing solutions and their operational architectures, along with the core research evaluation to identify methods that can be deployed in production environments.

In comparison to the traditional ML models, tensor decomposition is a powerful unsupervised ML method, capable of extracting latent (previously unseen) information from complex data and produce interpretable results. Since tensor factorization is an unsupervised method, it produces results with good generalizability to novel malware and requires less amount labeled data to achieve production-level performance. Many cyber data including network traffic and malware are naturally multi-dimensional. Therefore, they can be naturally represented as tensors. The higher dimensional representation of these data allows the identification of extremely complex and hidden multi-faceted details that traditional ML models cannot recognize. Therefore, we can develop solutions to malware identification and characterization utilizing complex real-world data, allowing our results to be realistic and reflect how they would perform in production environments. We have so far reported solutions that surpass or compete with existing supervised methods for anomaly detection and malware classification, and have preliminary results for large-scale malware family classification. Our results that establish new benchmarks indicate a promising future for the utilization of tensors for cyberspace. More research in this field will allow developing powerful tools that can streamline the work of cyber defenders in the fight against the ever-growing malware threat.

2 Preliminary Results

2.1 Tensor Anomaly Detection and Malware/Benign-ware Classification

We have developed a general anomaly detection framework, based on non-negative tensor factorization, that is capable of performing precise detection of anomalous activities [8]. This framework is further expanded to include malicious activity identification on diverse set of tasks such as botnet traffic, users with compromised credentials, fraudulent credit card transactions, and spam e-mails [9]. Our unsupervised architecture has established state-of-the-art benchmarks on the datasets analysed, which surpass or compete with prior supervised solutions.

For the botnet detection, our prior anomaly detection framework utilized the network activity of the compromised devices. We have also analysed malware using static analysis based features to perform classification. In this work, a novel ensemble tensors algorithm, named Random Forest of Tensors (RFoT)[1], which exploits the philosophy "wisdom of crowds" was developed [10]. In this work, we show that RFoT is capable of performing accurate classification of malware and benign-ware in a semi-supervised setting, using the Windows Portable Executables PE Header information as features from the popular malware bench-marking dataset EMBER-2018 [4], with only a small quantity of labeled data.

[1] RFoT poster is available at https://www.maksimeren.com/poster/Random_Forest_of_Tensors_RFoT_MTEM.pdf

Table 1: EMBER-2018 dataset default train and test set split, and malware family and sample counts are displayed. Novel families for the know (or train) set are the families that only exist in the training set. The novel families for unknown (or test) set are the families that only exist in the test set (i.e. we do not see these families during training, or we do not have known specimens for reference). *Min Family* and *Max Family* columns show the minimum and maximum number of samples exist for a family in the dataset. For instance, there are malware families with single sample in both known and unknown sets. *Samples/Family* column shows the average number of samples per family. We used the entire dataset which contains the rare and novel families, making the classification task complex.

Set	Families	Samples	Novel Families	Novel Samples	Min Family	Max Family	Samples/Family
Known (Train)	2,730	289,026	1,982	11,157	1	16,689	105.87
Unknown (Test)	916	99,216	168	363	1	19,260	315.53

2.2 High-Quality Semi-Supervised Classification of Malware Families

Our prior work has shown that tensor factorization is a powerful unsupervised machine learning tool that can be effectively used to tackle prominent and challenging cyber-security problems. We have preliminary results showing the application of our tools to identification of malware families, and malware families that were not seen before.

In our preliminary work [7], we classify the entire malware families exist in the EMBER-2018 dataset. To do this, we have developed a novel semi-supervised framework, named *HNMFk Classifier*, which builds a hierarchical graph, using malware PE header as features, by taking non-negative matrix factorization with the automatic model determination. At the leaves of this graph, we perform semi-supervised classification where known samples are utilized to classify unknown specimens into their respective families. *HNMFk Classifier* utilizes the *NMFk* algorithm which was introduced by us [3, 6, 22, 2]. *NMFk* is a non-negative matrix factorization method that enables the determination of optimal number of latent topics. Using *NMFk*, we can determine the number of types of malware families in a dataset. By performing this in a hierarchical manner, at each depth we can analyse finer-grained details of the malware and archive better separability of different families even with extremely small number of labelled data.

This architecture could be understood if we look at another dataset of news articles as an example. Let us assume there are 3 higher-level topics of articles; sport, technology, and economy. If we cluster these new articles with *NMFk*, we should expect to obtain 3 optimal clusters. At the same time, however, *NMFk* might identify four clusters, where we have an additional topic that combines news articles about sport and technology. To further separate this new topic, we can apply *NMFk* again on this additional cluster of news articles on sport and technology. Similarly, we can further divide the cluster containing sports news articles into sub-topics such as soccer, football, tennis, and skiing by applying additional *NMFk* procedures. This is the idea behind the hierarchical approach, and consequently our new algorithm *HNMFk Classifier*. How do we separate the malware specimens further when we have a more heterogeneous cluster? We can continue applying *NMFk* and build a hierarchical graph where as we go deeper in the graph towards the leafs we begin to investigate finer-grained details of the features, and achieve better separability of the malware specimens.

We report preliminary results obtained when using *HNMFk Classifier* to identify malware families in EMBER-2018, [7]. In Table 1, we show the number of families and the corresponding number of samples that exist in the default split of the dataset. Note that the entire dataset is used in this experiment, which contains rare families, families that are only present in the known set (train), and the novel families that are only present in the unknown set (test). This distribution of the data makes this task complex, but at the same time makes our experiment realistic by resembling a production environment. As far as our knowledge, we are the first to perform such scale experiment with a realistic setting for malware family classification. In Table 2, we show the results of the *HNMFk Classifier* compared to other baseline models where prior research had reported benchmarks on.

HNMFk Classifier, a semi-supervised model, surpasses all the other models even though they are supervised. Another feature of *HNMFk Classifier* is its ability to perform abstaining predictions (i.e. it can say "I do not know" when it does not know instead of guessing). This allows *HNMFk Classifier* to do abstaining classification on the specimens that it is not sure about, or for the novel families (families that we did not see before, or was not part of the known/train set). In Table 2, we can

Table 2: HNMFk Classifier is compared against the state-of-the-art supervised classifiers. The ability of the HNMFk to discover novel families is also shown. HNMFk Classifier, a semi-supervised method, surpasses the previous state-of-the-art models, which are supervised, in malware family classification.

Model	F1	Abstaining Seen (%)	Abstaining Novel (%)
HNMFk Classifier (semi-supervised)	**0.796**	22.055	**42.699**
LightGBM (supervised)	0.115	NA	NA
XGBoost (supervised)	0.737	NA	NA
MLP (supervised)	0.652	NA	NA

also see that the 22% of the specimens that existed in the known set were classified as abstaining. Differently, 42% of the specimens from the novel families were classified as abstaining. This indicates the ability of the *HNMFk Classifier* to recognize zero-day specimens.

3 What we Propose to Develop:

3.1 New Malware Multi-phenomenological Signatures

We will utilize the latent features extracted by *HNMFk Classifier* for fast characterization of malware. *HNMFk Classifier* will allow us to find which malware families carry what features. These latent features will then be used to create YARA[2] rules to identify malware, types of malware, and malware families. Being able to discover the latent features, specifically in a hierarchical setting, could enable us to develop classifiers that surpass the capabilities of existing solutions. Such methodology can further gain strength by representing malware specimens as tensors. For instance, one example of such tensor could have the dimensions *(Family* $\times$ *Byte N-Grams* $\times$ *Location)*, with count of occurrence as the tensor entries. In another example, we can have more dimensions in a binary tensor, such as *(Family* $\times$ *Num. of Strings* $\times$ *Num. of Sections* $\times$ *Num. of Imports* $\times$ *Num. of Exports* $\times$ *File Size)*. These are only a few examples, but not a exhaustive list of possible tensor configurations for a such application. The new multi-phenomenological signatures will be used for malware detection in real time.

Data: There are several public static malware datasets that are available to use:

- EMBER-2018[3], a popular benchmarking dataset, contains static malware analysis based features extracted from 1.1 Windows executables [4]. Specifically, the contents of the PE Header are used as the feautures. The dataset contains both malware and benign-ware, and the malware specimens contains family labels obtained form AVClass.
- A well knowm malware repository, VirusShare[4] dataset [11], is also available to our use.
- Sophos Reversing Labs has recently released a new large scale malware dataset, SOREL-20M[5], consisting about 8TB of metadata, labels and features from 20 million Windows Portable Executable files [12]. The dataset also contains 10 million production scale malware specimens.
- A new dataset named MOTIF[6], which contains accurate malware family labels, compared to AVClass, will be published by DREAM Lab in the coming months.

3.2 Description of Malware Dynamics, Source Identification, and Hosts Ranking

Another open area of research involves the analysis of malware dynamics. We will use malware network traffic data and our *RESCALk* tensor factorization method [21] to build the tensors describing malware dynamics and hosts interaction. For instance, we will build a tensor with dimensions *(*IP_{Source} $\times$ $IP_{Destination}$ $\times$ *Time)* to identify latent communities and ranking via centrality measures the roles

[2]YARA: https://github.com/virustotal/yara
[3]EMBER-2018: https://github.com/elastic/ember
[4]VirusShare: https://virusshare.com/
[5]SOREL-20M: https://github.com/sophos-ai/SOREL-20M
[6]MOTIF (under review): https://openreview.net/forum?id=A1yBi6zg3_C

of malicious hosts (for example C&C servers and compromised users), the evolution of the bastion devices, and lateral movement of a threat actor or insider threat at an organization network in time. Our understanding to latent malicious communities and their dynamics in a network can shed a light into ways to reduce long and expensive recovery and identification times. This approach can further be extended to the analysis of malware dynamics in a host device, using dynamic malware analysis-based features instead.

Take, for example, a tensor with dimensions *(Family* $\times$ *API Call* $\times$ *Time)* to be used to characterize different malware behaviors in a host device, which can later be used as an anti-virus scheme. This can be further extended into a multi-modal setting for behaviour analysis, where we fuse information across multiple tensors such as *(Family* $\times$ *API Call* $\times$ *Time)*, *(Family* $\times$ *Network Activity* $\times$ *Time)*, and *(Family* $\times$ *Memory Activity* $\times$ *Time)*.

Similarly, the evolution of malware families can be analyzed, using static malware analysis-based features, via a tensor with dimensions *(Sample/family* $\times$ *Features* $\times$ *Timestamp)*. This can help us with identifying what features are modified for the malware families in time, as we did for scientific literature [1]. In addition, this can enable for us to understand authorship attribution of malware families, as we can see which families carry correlation or share similarities in time.

Finally, static malware analysis and dynamic malware analysis-based features can both be combined in a multi-modal approach. For example, take the tensors build using dynamic features; *(Family* $\times$ *API Call* $\times$ *Time)*, *(Family* $\times$ *Network Activity* $\times$ *Time)*, and *(Family* $\times$ *Memory Activity* $\times$ *Time)*, together with the tensor build using static features (such as using Windows Executable PE Header, or Android application based features); *(Family* $\times$ *Features)* (two example tensors which we may use; *(Family* $\times$ *Permissions* $\times$ *Activities* $\times$ *Services)* for Android, and *(Family* $\times$ *Num. of Strings* $\times$ *Num. of Sections* $\times$ *Num. of Imports* $\times$ *Num. of Exports* $\times$ *File Size* $\times$ *Timestamp)* for a Windows malware). Multi-modal-based approach could enable us to gain a deeper understanding of the evolution and behavioral patterns of malware characteristics, and identification of malware.

Data: Several malware network traffic based datasets are available by Canadian Institute for Cybersecurity by the University of New Brunswick[7]. Additionally, several malware traffic with background traffic records are available by Stratosphere Labs[8]. Below is a list of specific datasets that can be used:

- CCCS-CIC-AndMal-2020[9] contains static and dynamic analysis based features for Android malware. Alongside the family labels, this dataset also includes labels for 14 different malware categories such as adware, backdoor, ransomware, etc..
- CICMalDroid 2020[10] also includes static and dynamic analysis based features for Android malware.
- CIRA-CIC-DoHBrw-2020[11] is a dataset consist of DNS over HTTPS (DoH) flow records of malicious and benign traffic.

References

[1] Boian Alexandrov and James P. Smith. Scientific leadership identification and characterization. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2021. Presented at the NSARD, Washington, United States, LA-UR-21-23541.

[2] Boian Alexandrov, Velimir Vesselinov, and Kim Orskov Rasmussen. SmartTensors unsupervised ai platform for big-data analytics. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2021. LA-UR-21-25064.

[3] Boian S Alexandrov, Ludmil B Alexandrov, Valentin G Stanev, et al. Source identification by non-negative matrix factorization combined with semi-supervised clustering, September 15 2020. US Patent 10,776,718.

[7] UNB Datasets: https://www.unb.ca/cic/datasets/index.html
[8] Stratosphere Labs: https://www.stratosphereips.org/datasets-overview
[9] CCCS-CIC-AndMal-2020: https://www.unb.ca/cic/datasets/andmal2020.html
[10] CICMalDroid 2020: https://www.unb.ca/cic/datasets/maldroid-2020.html
[11] CIRA-CIC-DoHBrw-2020: https://www.unb.ca/cic/datasets/dohbrw-2020.html

[4] H. S. Anderson and P. Roth. EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. *ArXiv e-prints*, April 2018.

[5] K. Bissell and L. Ponemon. The cost of cybercrime. Technical report, Accenture, Ponemon Institute, 2019.

[6] Gopinath Chennupati, Raviteja Vangara, Erik Skau, Hristo Djidjev, and Boian Alexandrov. Distributed non-negative matrix factorization with determination of the number of latent features. *The Journal of Supercomputing*, pages 1–31, 2020.

[7] M. E. Eren, M. Bhattarai, R. J. Joyce, E. Raff, C. Nicholas, and B. Alexandrov. Semi-supervised classification of malware families via hierarchical non-negative matrix factorization with automatic model determination. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2021. LA-UR-21-29919.

[8] M. E. Eren, J. S. Moore, and B. S. Alexandrov. Multi-dimensional anomalous entity detection via poisson tensor factorization. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 1–6, 2020.

[9] M. E. Eren, J.S. Moore, E.W. Skau, M. Bhattarai, E.A. Moore, and B. Alexandrov. General-purpose unsupervised cyber anomaly detection via non-negative tensor factorization. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2021. LA-UR-21-29195.

[10] M. E. Eren, C. Nicholas, R. McDonald, and C. Hamer. Random forest of tensors (rfot), 2021. Presented at the 12th Annual Malware Technical Exchange Meeting, Online, 2021.

[11] External Data Source. Virusshare dataset, 2018.

[12] Richard Harang and Ethan M. Rudd. Sorel-20m: A large scale benchmark dataset for malicious pe detection, 2020.

[13] IBM. Cost of a data breach report. Technical report, IBM, 2021.

[14] Kaspersky. Machine learning methods for malware detection. Technical report, 2020.

[15] Microsoft 365 Defender Threat Intelligence Team. Microsoft researchers work with intel labs to explore new deep learning approaches for malware classification, 2020. https://www.microsoft.com/security/blog.

[16] Andre T Nguyen, Edward Raff, Charles Nicholas, and James Holt. Leveraging uncertainty for improved static malware detection under extreme false positive constraints. *arXiv preprint arXiv:2108.04081*, 2021.

[17] Tam N. Nguyen. Attacking machine learning models as part of a cyber kill chain. *ArXiv*, abs/1705.00564, 2017.

[18] Bernardo Quintero. Virustotal += bitdefender theta, 2019.

[19] Bernardo Quintero. Virustotal += sangfor engine zero, 2019.

[20] The Independent IT Security Institute. Malware statistics & trends report: Av-test, Oct 2021.

[21] Duc P Truong, Erik Skau, Vladimir I Valtchinov, and Boian S Alexandrov. Determination of latent dimensionality in international trade flow. *Machine Learning: Science and Technology*, 1(4):045017, 2020.

[22] Raviteja Vangara, Manish Bhattarai, Erik Skau, Gopinath Chennupati, Hristo Djidjev, Thomas Tierney, James P Smith, Valentin G Stanev, and Boian S Alexandrov. Finding the number of latent topics with semantic non-negative matrix factorization. *IEEE Access*, 2021.