# Auto Semi-supervised Outlier Detection for Malicious Authentication Events

Kaiafas, Georgios ; Hammerschmidt, Christian; Lagraa, Sofiane; State, Radu

**DOI**
10.1007/978-3-030-43887-6_14

**Publication date**
2020

**Document Version**
Final published version

**Published in**
Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD 2019, Proceedings

# Auto Semi-supervised Outlier Detection for Malicious Authentication Events

Georgios Kaiafas[1(✉)], Christian Hammerschmidt[2], Sofiane Lagraa[1], and Radu State[1]

[1] SnT, University of Luxembourg, Luxembourg City, Luxembourg
{georgios.kaiafas,sofiane.lagraa,radu.state}@uni.lu
[2] TU Delft, Delft, The Netherlands
c.a.hammerschmidt@tudelft.nl

**Abstract.** Cyber-attacks become more sophisticated and complex especially when adversaries steal user credentials to traverse the network of an organization. Detecting a breach is extremely difficult and this is confirmed by the findings of studies related to cyber-attacks on organizations. A study conducted last year by IBM found that it takes 206 days on average to US companies to detect a data breach. As a consequence, the effectiveness of existing defensive tools is in question. In this work we deal with the detection of malicious authentication events, which are responsible for effective execution of the stealthy attack, called *lateral movement*. Authentication event logs produce a pure categorical feature space which creates methodological challenges for developing outlier detection algorithms. We propose an auto semi-supervised outlier ensemble detector that does not leverage the ground truth to learn the normal behavior. The automatic nature of our methodology is supported by established unsupervised outlier ensemble theory. We test the performance of our detector on a real-world cyber security dataset provided publicly by the Los Alamos National Lab. Overall, our experiments show that our proposed detector outperforms existing algorithms and produces a 0 *False Negative Rate* without missing any malicious login event and a *False Positive Rate* which improves the state-of-the-art. In addition, by detecting malicious authentication events, compared to the majority of the existing works which focus solely on detecting malicious users or computers, we are able to provide insights regarding when and at which systems malicious login events happened. Beyond the application on a public dataset we are working with our industry partner, POST Luxembourg, to employ the proposed detector on their network.

**Keywords:** Outlier detection · Ensemble learning · Cybersecurity · Embedding · Semi-supervised learning

## 1 Introduction

Lateral movement attack is a stealth and well orchestrated attack where the adversaries gain shell access without necessarily creating abnormal network traffic. They make use of legitimate credentials to log into systems, escalate privileges


P. Cellier and K. Driessens (Eds.): ECML PKDD 2019 Workshops, CCIS 1168, pp. 176–190, 2020.
https://doi.org/10.1007/978-3-030-43887-6_14

using lateral movements and subsequently manage to traverse a network without any detection. The JP Morgan Chase [36] and Target hacks [23] are two well known examples of attacks where the adversaries stayed undetected while they traversed network.

Researchers have addressed malicious logins detection by evaluating their methods on a real-world cyber security dataset provided freely by the Los Alamos National Lab [19]. Existing works focus on detecting malicious users or computers which leads to classifying all the generated events from a user or computer as malicious or legit. As a result, it fails to detect which specific events are malicious and does not provide any information regarding when the adversaries manage to impersonate benign users. Additionally, most of the existing approaches on this dataset are questionable and the authors in [32] provide further details of their study.

A common characteristic of *login logs* or *authentication events* is being comprised of multidimensional categorical variables. Categorical variables stem from discrete entities and their properties, e.g. source user, destination computer, or protocol type. The underlying values of this type of variables are inherently unordered and as a consequence it is often hard to define similarity between different values of the same variable. As such, detecting anomalies on discrete data is challenging and is not a well studied topic in academia; the primary focus is on continuous data. Moreover, the prominent challenge in the defensive cyber world is to develop effective approaches which are realistic.

A possible solution to this point comes from the semi-supervised approaches [22] that do not require anomalous instances in the training phase. These approaches model the normal class and identify anomalies as the instances that diverge from the normal model. In real-world problems where the amount of unlabeled data is immense, identifying events that are not suspicious needs a lot of manual work and underlies the risk of miss-labeling true anomalous events. Hence, our motivation to develop our auto approach is to alleviate analysts from time expensive and monotonous tasks that include a significant amount of uncertainty.

In this work, we analyze authentication events using the Los Alamos authentication dataset [19] and we aim at detecting unauthorized events to services or computers in contrast to the majority of the existing works. We propose an embedding based and automatic semi-supervised outlier detector to reduce the false positives produced by an unsupervised outlier ensemble. In particular, our approach is an ensemble approach where we develop an unsupervised outlier ensemble to identify the most confident normal data points which will feed the semi-supervised detector to ultimately detect outliers. Our technique could be considered as a sequential outlier ensemble approach where two dependent components are developed for an outlier detection task. We refer to the authors of [1] for the details of outlier ensembles categories.

The contributions of our proposed approach are:

– We produce an embedding space via the Logistic PCA [25] algorithm that has potentiality of better representing the normal behavior.

- We develop the Restricted Principal Bagging (*RPB*) technique, an improved variant of the well established feature bagging technique [27], that works on the principal components space.
- We introduce a new unsupervised combination function, *Vertical Horizontal Procedure* (*VHP*), that leverages gradually the predictions of individual and smaller scale ensemble members.
- We automatically build an automatic semi-supervised ensemble by combining the aforementioned novel components to effectively detect malicious events.

Overall, our approach improves current state-of-the-art by achieving a 0.0017 *FPR* and 0 *FNR*; without missing any malicious login event. It is tested on an extremely imbalanced data sample of the real-world authentication log dataset provided by Los Alamos. In this challenging data sample the percentage of malicious events is 0.0066% which is 1348 times lower than the average outlier percentage in datasets used for outlier detection [33].

These improvements enhance our understanding of anomalous patterns since existing state of the art methods fail to capture all the anomalous patterns. It is particularly important for the practical implementation to keep the base rate fallacy in mind: Reducing the number of the false positives by 150 compared to state of the art means that we enable cyber analysts spending less time on monotonous tasks of pruning false alerts.

Detecting malicious events instead of users or computers provides actionable insights to analysts by answering questions related to when exactly and at which systems a malicious event happened. Our work could also be used to extend existing methodologies which detect malicious users to further detect malicious events. To the best of our knowledge, this work is the first automatic semi-supervised attempt that aims at detecting anomalous authentication events.

The rest of the paper is organized as follows. We briefly review related work in Sect. 2. Then, we continue by describing extensively how we develop each component of our approach in Sect. 3. In Sect. 4 we explain in detail the dataset and we present the experimental settings and results. We close in Sect. 5, where we conclude with remarks and future research directions.

## 2 Related Work

**Anomaly Detection in Categorical Data.** In [17], the authors proposed a distance based semi-supervised anomaly detection method. In particular, the distance between two values of a categorical attribute is determined by the co-occurrence of the values of other attributes in the dataset. In [30], the authors proposed an unsupervised anomaly detector based on subspaces. It examines only a small number of low dimensional subspaces randomly selected to identify anomalies. In [7], the authors proposed an anomaly detection method on heterogeneous categorical event data. The method maximizes the likelihood of the data by embedding different events into a common latent space and then assessing the compatibility of events. Furthermore, approaches that are based on pattern

mining techniques have been developed. For instance, in [2], the authors proposed to identify anomalies using pattern-based compression, and [14] detects patterns in short sequences of categorical data.

**Malicious Logins Detection.** The Los Alamos National Lab provides a publicly available dataset [19] which is the most used and is related to authentication login events. There is a non-exhaustive list of papers analyzing this dataset for detecting abnormal authentication activities. The majority of the related work of this dataset focuses on detecting anomalous entities, users or computers [4,13,15,16,21,39]. On the other hand, only few works [18,29,35] detect anomalous events. The most used approach among all the existing works is the bipartite graph.

This work effectively detects malicious authentication events instead of malicious entities which gives the opportunity to analysts to correlate identified malicious authentication events with malicious events on other data sources. In addition, detecting anomalous entities could be considered as a subset of detecting malicious events because from the latter we can derive the former but not vice versa. Furthermore, our work is the first automatic semi-supervised outlier ensemble approach that is developed with the aid of established theory on outlier ensembles [1,44]. It is composed of novel and existed methods never tested for outlier detection on categorical data and especially on authentication logs.

## 3 Methodology

We propose a novel outlier ensemble detector for categorical data which automatically creates the "non-polluted" by outliers training set of a semi-supervised ensemble. More specifically, first it builds in an unsupervised way an outlier ensemble on all data points to identify with a relative confidence data points that are normal Secondly, it develops a semi-supervised ensemble detector which is trained only on the (normal) data points derived from the first phase. Finally, the semi-supervised ensemble classifies new observations (data points not in the training set) as belonging to the learned normal class or not. Figure 1 illustrates the sequential and automatic nature of our approach. Throughout this work we use *outliers* and *anomalies* interchangeably.

### 3.1 Phase 1

Unsupervised outier detection algorithms detect outliers based on their algorithmic design [45]. In this work, we reverse the problem of unsupervised outlier detection to unsupervised normal detection by using established oulier ensemble theory. The aim of the this phase is to create the training dataset of the semi-supervised model; normal data points. In particular, we independently employ two unsupervised detectors to build an outlier ensemble on bagged subspaces and finally identify the most confident normal data points.

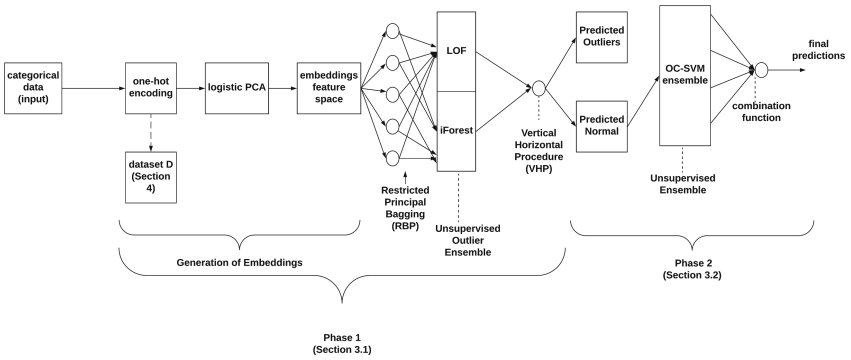


**Fig. 1.** Auto Semi-supervised Outlier Detector

**Generation of Embeddings.** Our dataset is a pure categorical dataset and we produce the embeddings of our proposed detector via the Logistic PCA algorithm [25]. This algorithm produces principal components and our aim is to find principal components that explain at least 90% of the total variance. We suggest a high percentage of explained variance because it means that we represent an amount of information very close to the information included in the original variables. We could have selected a different number of principal components that explain more than 90% of the total variance but we leave this sensitivity analysis for the future. Additionally, according to Theorem 2 of [25] we select columns to decrease the deviance the most. This Theorem states that for Logistic PCA the standard basis vector which decreases deviance the most is the one corresponding to column with mean closest to 1/2.

**Restricted Principal Bagging.** Our motivation for developing the *RPB* - *Restricted Principal Bagging* technique is to upper bound the sample space of the principal components and then add randomness in a similar way like the Feature Bagging technique [27]; randomness is a key ingredient of outlier ensemble techniques. Our technique aims at capturing the individual contribution of each principal component to the total explained variance. As such, we adjust the Feature Bagging technique [27] to work for principal components and find subspaces to detect ouliers more effectively. We explain in detail the *RPB* technique in Algorithm 1.

Firstly, *RPB* creates multiple random subsets of the first $p$ principal components and each of these subsets is denoted by $S_j$. We denote by $PCs$ the principal components that we keep after we have applied the Theorem 2 and we also call as $V$ the set of all the $S_j$. Hence, $V = \{S_1, S_2, S_3, S_3, S_4, S_5\} = \{0.04 * \mid PCs \mid, 0.1 * \mid PCs \mid, 0.2 * \mid PCs \mid, 0.3 * \mid PCs \mid, 0.4 * \mid PCs \mid, 1.0 * \mid PCs \mid\}$. Then for a $S_j$ and for $Iter$ iterations it samples from a uniform distribution U$(d/2, d-1)$ without replacement, where $d$ is the dimensionality of $S_j$. Hence, for each $Iter$ iteration $N_j$ principal components are sampled out and create a dataset $F_j$. Finally, an unsupervised outlier detector with random parameters is applied to $F_j$.

**Algorithm 1.** Restricted Principal Bagging

**Input:**

- **V** the set of all the $S_j$
- **OD** is an unsupervised Outlier Detection Algorithm which outputs numeric outlier scores for each data point
- **Iter** represents how many times we perform feature sampling

**Output**

- **E** is a vector composed of oulier scores for each data point

**Procedure:**

1: **for all** $S_j$ in $V$ **do**
2: **for** $i = 1, 2, 3, 4, ...Iter$ **do**
3: Randomly sample from a uniform distribution between $\lceil d/2 \rceil$ and $(d-1)$, where $d$ is the number of the principal components in $S$
4: Randomly pick, without replacement, $N_i$ principal components to create a subset $F_i$
5: Apply OD on $F_i$ feature space
6: **end for**
7: **end for**

**Unsupervised Outlier Detectors.** We employ two well performing and established unsupervised detectors to combine them and identify the most confident normal points that will feed afterwards the semi-supervised learner. We intentionally select heterogeneous detectors in order to increase the probability that they capture different patterns of anomalies. Also, we could have selected more than two heterogeneous unsupervised detectors to build the ensemble but for the current experiments we showcase the promising performance of the most straightforward version of our approach.

Firstly, we select iForest [28] which is a tree-based and state-of-the-art detector which performs the best across many datasets [11] and applications [9,41]. Secondly, we select LOF [5] which is a proximity-based method and designed to detect local outliers (see [1] for details in local and global outliers). It is also a state-of-the-art outlier detection algorithm and there is a large body of research on this detector [3,12,27,45].

The procedure that we follow at this phase is of running a detector over a range of parameters without leveraging the ground truth to tune the detectors. This procedure is interpreted as an ensemblar approach and we refer to [1] where the authors discuss the topic extensively. As such, we run LOF with different random values for the neighborhood parameter. Also, we run iForest with the Cartesian product of parameters $IF = \{(Number\ Of\ Estimators \times Maximum\ Samples \times Maximum\ Features)\}$.

LOF and iForest independently apply *RPB* on set $V$ to build the ensemble version of LOF and iForest. Henceforth, we call $LOF$ - $RPB\ scores_j$ and *iForest - RPB scores*$_j$ the outlier scores that are produced by applying the *RPB* technique on a subset $S_j$ and employing the LOF and iForest respectively. The final step is to combine these results in an unsupervised way to find the most

confident normal data points. We introduce later the *VHP* combination function to combine these results. Finally, we call as $W$ the most confident normal data points that will feed the semi-supervised algorithm to learn the normal behavior and as $O$ the least confident normal data points.

**VHP Combination Function.** As we discussed before, the *RPB* algorithm builds a couple of LOF and iForest ensembles on each subset $S_j$. Hence, we propose a strategy to effectively combine and gradually take advantage of these couples of ensembles instead of applying a global combination function across all the *LOF - RPB scores*$_j$ and *iForest - RPB scores*$_j$. The authors in [43] develop a novel local combination function and highlight the effectiveness of this type of combination functions.

In our strategy we utilize the *Averaging* combination function to calculate the average scores of ensemble members. The reason why we select this function is that the average score is the most widely used in outlier ensemble literature and performs the best in most cases [8]. It is worth noting that combining effectively outlier ensemble members without leveraging the ground truth is challenging and the authors in [1,24,44] extensively discuss the topic.

In particular, firstly we normalize all the *LOF - RPB scores*$_j$ and *iForest - RPB scores*$_j$ and then apply the *Averaging* function to get the average scores on each $S_j$. As such for each subset $S_j$ we build an ensemble produced by these combined outlier scores. We refer to this ensemble as *LOF Ens & iForest Ens*. Afterwards, we convert the numeric outlier scores of each *LOF Ens & iForest Ens* ensemble to binary values based on a threshold. Finally, we combine these binary values by utilizing the unweighted majority voting [40] technique to produce the output of *Phase 1*.

The conversion to binary values is referred as the *Vertical Strategy* and the combination of the binary values as the *Horizontal Strategy*. Henceforth, we call this combination function as *VHP*, Vertical Horizontal Procedure. All the outlier scores are normalized with the Z-score normalization scheme which is the most commonly used in outlier detection literature (see [1] for details in different normalization schemes).

### 3.2 Phase 2

At this phase we leverage the produced $W$ dataset of *Phase 1* to build the semi-supervised ensemble. The $W$ dataset is composed of the most confident normal class data points and via this dataset we learn the normal class patterns. As a result this procedure of our analysis makes our approach sequential and automatic at the same time. The desired outcome of this sequential approach is to reduce significantly the number of false positives of $O$ dataset after we have learnt the contour of the normal class.

Hence, we employ the OCSVM - One-Class SVM algorithm [34] which is a well performing algorithm that is applied to several problems such as, fraud detection [37] and network intrusion detection [26]. OCSVM is a boundary

method that attempts to define a boundary around the training data (normal class), such that new observations that fall outside of this boundary are classified as outliers [38].

Our proposed approach is developed on a pure unsupervised setup and as a result we do not seek for the best performing parameters. Hence, without any loss of generality we select as parameters of the OCSVM algorithm the Cartesian product $B = \{(Type\ of\ kernel \times Upper\ bound\ of\ training\ errors \times Kernel\ coefficient)\}$. The procedure that we follow at this phase is analogous to *Phase 1* where we execute each detector over a range of parameters without leveraging the ground truth to tune the performance.

In particular, we independently execute several training runs of the OCSVM on $W$ with different parameter values from set $B$. The number of training executions is equal to the carnality of set B. Next, for each execution of OCSVM an outlier score vector is produced which has length equal to the number of observations of $O$ dataset. Finally, we combine these outlier score vectors, without leveraging the ground truth, to ultimately produce the final outlier score for each data point. It is worth noting that we could have selected any other set of parameters as input for the OCSVM algorithm. The procedure of running a detector over a range of parameters without the use of labels is interpreted as an ensemblar approach (see [1] for details).

## 4 Experiments and Evaluation

The major objective of our experiments is to demonstrate the effectiveness of our proposed auto semi-supervised detector by comparing it with works which detect malicious login events. On the one hand, we do not leverage the ground truth to tune any component of our methodology on the other hand, we use the ground truth to present the performance of *Phase 1* as well as *Phase 2*.

### 4.1 Dataset

The Los Alamos National Laboratory provides a freely available and comprehensive dataset[1] [19]. It includes 58 consecutive days of credential-based login events, of which days the 3 to 29 are labelled as malicious or normal via a RedTeam table. This dataset consists of 1 billion events and is an excessively imbalanced dataset; the percentage of the malicious login events is 0.000071%.

Each authentication event contains the attributes: time, source user, destination user, identifier per domain, source computer, destination computer, authentication type, logon type, authentication orientation, and authentication result. In addition, the authentication events are Windows-based authentication events from both individual computers and centralised Active Directory domain controller servers [20]. We also create a new attribute for each authentication event based on if source computer and destination computer are the same or different.

[1] https://csr.lanl.gov/data/cyber1/.

This new boolean feature quantifies the Local or Remote rule respectively. In our analysis, the time variable is excluded and as a result a purely categorical feature space is produced.

Developing a data mining methodology on 1 billion events would require a big data infrastructure but our work is not on proposing a computer engineering tool. Hence, we use a data sample to develop and evaluate our methodology. Sampling from a such an excessively imbalanced dataset usually produces samples composed of zero malicious login events which makes the evaluation of both classes impossible.

Hence, we seek for a random sample of $150,000$ consecutive authentication events that contains at least 5 malicious events in order to thoroughly evaluate our approach. In other words, the percentage of malicious events has to be at least 0.0033%. Consequently, our randomly selected sample contains 10 malicious events and its percentage of malicious events is 0.0066%. Then, on the sampled categorical space we apply the one-hot technique to produce the input binary space of the Logistic PCA algorithm. The dimension of the this binary space is $150,000 \times 2700$ and we refer to this dataset as $D$.

### 4.2 Experiment Environment

We used the logisticPCA [25] R package for the implementation of the Logistic PCA algorithm and the data.table [10] R package for fast data manipulation. The iForest, LOF and OCSVM algorithms were executed using the Python *Scikit-learn* library [31].

### 4.3 Experimental Settings

**Phase 1.** We apply the Logistic PCA on the D dataset ($150,000 \times 2700$) and we keep 900 principal components which explain 93% of the total variance. Afterwards, we apply the Theorem 2 we explained in Sect. 3.1 and we return 500 principal components which will be the embeddings feature space denoted by $PCs$.

The exact parameters of LOF and iForest are presented in Table 1. LOF is employed with different number of neighbors as input whereas the input parameter set of iForest is the Cartesian product $IF = \{(Number\ Of\ Estimators \times Maximum\ Samples \times Maximum\ Features)\}$.

**Phase 2.** Table 2 presents the parameters at *Phase 2*. In Sect. 3.2 we defined the set $B$ which is the Cartesian product of the input parameter values of OCSVM. In addition, the *Averaging* combination function is utilized to unify the outlier scores of all OCSVM executions.

**Table 1.** Setting parameters

| | Subsets $\boldsymbol{S}$ | Parameters |
|---|---|---|
| LOF | $V$ = {4%, 10%, 20%, 30%, 40%, 100%} | $Neighbors$ = {5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100} |
| iForest | $V$ = {4%, 10%, 20%, 30%, 40%, 100%} | $NumberOfEstimators$ = {100, 200, 300, 400} |
| | | $MaximumFeatures$ = {10%, 20%, 40%, 60%} |
| | | $MaximumSamples$ = {10%, 30%, 50%} |

**Table 2.** Setting parameters

| | |
|---|---|
| *nu* | {0.0001, 0.0005, 0.001, 0.005} |
| *gamma* | {0.01, 0.05, 0.09, 0.001} |
| *kernel* | {"rbf", "sigmoid"} |

**Settings for Comparisons:** We develop the *VHP-Ensemble* with our proposed *VHP* combination function accompanied with the *RPB* algorithm by leveraging different subsets of principal components as we have discussed earlier. Also, we develop the *Vanilla-Ensemble* to compare our proposed ensemble with. It employs the iForest and LOF detector on the whole *PCs* embeddings space, the feature bagging technique by Lazarevic [27] and the *Averaging* combination function. The components of the developed ensembles and their corresponding names are presented in Table 3.

**Table 3.** Ensembles of *Phase 1*

| | Detector | | Principal components of subsets S | | | | | | | Combination | | Bagging | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ensmbles** | **LOF** | **iForest** | **20** | **50** | **100** | **150** | **150** | **200** | **500** | **VHP** | **Avg.** | **RPB** | **Lazarevic** |
| *VHP* | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** | **Yes** | No | **Yes** | No | **Yes** | No |
| *Vanilla* | **Yes** | **Yes** | No | No | No | No | No | No | **Yes** | No | **Yes** | No | **Yes** |

### 4.4 Evaluation

***Phase 1***

In Table 4, we summarize the performance of the ensembles discussed previously and presented in Table 3. Since the output of *Phase 1* is two sets, $W$ and $O$, we evaluate our detectors using the precision and recall measures. We also showcase the sensitivity of the ensembles by reporting the presicion and recall scores based

on different thresholds $m$; number of reported outliers. In our analysis $m$ plays the role of the confidence of finding normal data points.

We denote by $P_{@m}$ and $R_{@m}$ respectively, the precision and recall score produced with $m$ ranked data points which are considered as outliers. Table 4 is a typical example of the trade off between precision and recall. In our proposed approach the cost of higher precision is less than the cost of higher recall.

**Table 4.** Precision and Recall of the output of *Phase 1*

| Enembles | $P_{@1500}$ | $R_{@1500}$ | $P_{@5000}$ | $R_{@5000}$ | $P_{@7000}$ | $R_{@7000}$ |
|---|---|---|---|---|---|---|
| *VHP* | **0.015** | **1.0** | **0.008** | **1.0** | **0.007** | **1.0** |
| *Vanilla* | 0.005 | 0.8 | 0.0016 | 0.8 | 0.0011 | 0.8 |

***Phase 2.***
Since all the components of this work are developed in a pure unsupervised setup it is important to investigate the sensitivity of our approach, *VHP-Ensemble*. As such, we test multiple variants of this ensemble detector based on different numbers of reported outliers at *Phase 1*. In this way, we investigate the effect of *Phase 1* on building the semi-supervised ensemble detector.

Hence, we denote by *Detector-1500* the semi-supervised detector which is developed when a threshold rank $m = 1500$ is chosen for the *VHP-Ensemble*. The most outlier point among the $m = 1500$ reported outliers has a rank of 1. In the same fashion, we develop *Detector-5000* and *Detector-7000* where $m = 5000$ and $m = 7000$ respectively. Our motivation for selecting so large $m$ is that we want to feed the semi-supervised detector with the most confident normal data points. We identify them based on our intuition for the outliers percentage in our dataset. In our case, $m$ is at least 150 times greater than the number of true malicious authentication events.

We compare our methodology with works that are developed on the same level of granularity; detecting malicious authentication events. Detecting malicious users or computers means a huge amounts of events have to be further analyzed to identify which specific events are malicious. Since the existing works on malicious events is limited we compare our proposed detector with any kind of machine learning(supervised, semi-supervised, unsupervised) approach that is tested on authentication events. Hence, we evaluate all variants of our detector with (i) Siadati et al. [35], (ii) Lopez et al. [29], (iii) Kaiafas et al. [18].

In Fig. 2 we present a summary of the FPR and TPR scores of all the competitors. Amongst the competitors, Siadati et al. [35] achieves the lowest FPR whereas Kaiafas et al. and all the variants achieve the highest TPR; they do not miss any malicious login. In addition, *Detector-1500* achieves the lowest FPR among all the competitors. Ultimately, *Detector-1500* improves FPR of the Kaiafas et al. supervised detector by 10% (150 login events) and more than doubles Siadati's TPR. Siadati et al. detector is based on integrating security analysts knowledge into the detection system in the form of rules that define login

patterns. In other words, this detector does not improve the existing knowledge of the cyber analysts for anomalous patterns but instead relies on known rules to detect anomalies. As a consequence, the Siadati's rule based visualization detector misses 53% of the malicious logins.

In addition, each of the aforementioned approaches outperform the logisitic classifier of Lopez et al. [29] which achieves AUC 82.79%. We do not plot their reported FPR and TPR scores in Fig. 2 because their FPR scores are at least 5 times worse than the maximum FPR value in Fig. 2. Consequently, we avoid presenting a figure that is less readable and informative for the majority of the competitors.

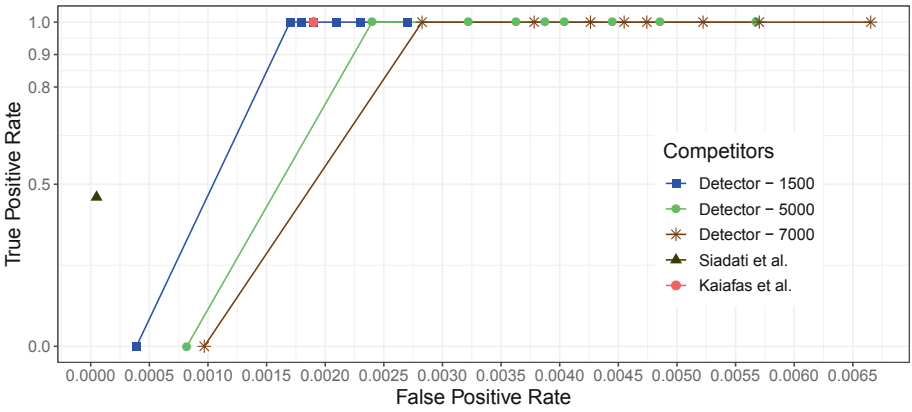


**Fig. 2.** Comparison of the Auto Semi-supervised Outlier Detector

## 5 Conclusion and Future Work

Our proposed automatic semi-supervised detector for malicious authentication detection outperforms the existent supervised algorithms and tools with the human in the loop. It is capable of capturing underlying mechanisms that produce anomalous authentication events. Our evaluation on a real-world authentication log dataset shows that we do not miss any malicious login events and improve the current state-of-the-art methods. Also, the sensitivity analysis showed that the rank threshold at *Phase 1* does not affect at all the TPR. On the other hand, the effect of the threshold on the FPR is not so noticeable. The semi-supervised ensemble detector improves the FPR of the unsupervised ensemble almost 9 times while all the developed variants did not miss any true malicious login events.

In the future we would like to extend this work by building an ensemble with multiple heterogeneous one-class classification algorithms [38]. Also, we want to model the authentication logs as graphs to produce embeddings with deep learning models [6]. Additionally, we intend to extend the existing work with

network representation learning techniques [42] instead of embeddings. Finally, an extensive comparative evaluation will follow based on the above improvements on many cyber-security datasets.

**Acknowledgement.** Georgios Kaiafas is supported by the National Research Fund of Luxembourg (AFR-PPP Project ID 11824564). Additionally, the authors would like to thank POST Luxembourg, the industrial partner of this project.

## References

1. Aggarwal, C.C., Sathe, S.: Outlier Ensembles. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54765-7
2. Akoglu, L., Tong, H., Vreeken, J., Faloutsos, C.: Fast and reliable anomaly detection in categorical data. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 415–424. ACM (2012)
3. Alshawabkeh, M., Jang, B., Kaeli, D.: Accelerating the local outlier factor algorithm on a GPU for intrusion detection systems. In: Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units, pp. 104–110. ACM (2010)
4. Bohara, A., Noureddine, M.A., Fawaz, A., Sanders, W.H.: An unsupervised multi-detector approach for identifying malicious lateral movement. In: 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), pp. 224–233. IEEE (2017)
5. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM SIGMOD Record, vol. 29, pp. 93–104. ACM (2000)
6. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans. Knowl. Data Eng. **30**(9), 1616–1637 (2018)
7. Chen, T., Tang, L.A., Sun, Y., Chen, Z., Zhang, K.: Entity embedding-based anomaly detection for heterogeneous categorical events. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, pp. 1396–1403 (2016)
8. Chiang, A., Yeh, Y.R.: Anomaly detection ensembles: In defense of the average. In: 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 3, pp. 207–210. IEEE (2015)
9. Ding, Z., Fei, M.: An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. IFAC Proc. Volumes **46**(20), 12–17 (2013)
10. Dowle, M., Srinivasan, A.: data.table: extension of 'data.frame' (2019). https://CRAN.R-project.org/package=data.table, r package version 1.12.2
11. Emmott, A.F., Das, S., Dietterich, T., Fern, A., Wong, W.K.: Systematic construction of anomaly detection benchmarks from real data. In: Proceedings of the ACM SIGKDD workshop on outlier detection and description, pp. 16–21. ACM (2013)
12. Goldstein, M., Uchida, S.: A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one **11**(4), e0152173 (2016)
13. Goodman, E., Ingram, J., Martin, S., Grunwald, D.: Using bipartite anomaly features for cyber security applications. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 301–306. IEEE (2015)
14. Hammerschmidt, C., Marchal, S., State, R., Verwer, S.: Behavioral clustering of non-stationary IP flow record data. In: 2016 12th International Conference on Network and Service Management (CNSM), pp. 297–301. IEEE (2016)

15. Heard, N., Rubin-Delanchy, P.: Network-wide anomaly detection via the dirichlet process. In: 2016 IEEE Conference on Intelligence and Security Informatics (ISI), pp. 220–224 (2016)
16. Heard, N., Rubin-Delanchy, P.: Network-wide anomaly detection via the dirichlet process. In: 2016 IEEE Conference on Intelligence and Security Informatics (ISI), pp. 220–224. IEEE (2016)
17. Ienco, D., Pensa, R.G., Meo, R.: A semisupervised approach to the detection and characterization of outliers in categorical data. IEEE Trans. Neural Netw. Learn. Syst. **28**(5), 1017–1029 (2017)
18. Kaiafas, G., Varisteas, G., Lagraa, S., State, R., Nguyen, C.D., Ries, T., Ourdane, M.: Detecting malicious authentication events trustfully. In: NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, April 2018
19. Kent, A.D.: Comprehensive, Multi-Source Cyber-Security Events. Los Alamos National Laboratory, New Mexico (2015). https://doi.org/10.17021/1179829
20. Kent, A.D.: Cyber security data sources for dynamic network research. In: Dynamic Networks and Cyber-Security, pp. 37–65. World Scientific, Singapore (2016)
21. Kent, A.D., Liebrock, L.M., Neil, J.C.: Authentication graphs: analyzing user behavior within an enterprise network. Comput. Secur. **48**, 150–166 (2015)
22. Khan, S.S., Madden, M.G.: One-class classification: taxonomy of study and review of techniques. Knowl. Eng. Rev. **29**(3), 345–374 (2014)
23. Krebs, B.: Target hackers broke in via HVAC company. Krebs on Security (2014)
24. Kriegel, H.P., Kroger, P., Schubert, E., Zimek, A.: Interpreting and unifying outlier scores. In: Proceedings of the 2011 SIAM International Conference on Data Mining, pp. 13–24. SIAM (2011)
25. Landgraf, A.J., Lee, Y.: Dimensionality reduction for binary data through the projection of natural parameters. arXiv preprint arXiv:1510.06112 (2015)
26. Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J.: A comparative study of anomaly detection schemes in network intrusion detection. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 25–36. SIAM (2003)
27. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 157–166. ACM (2005)
28. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE (2008)
29. Lopez, E., Sartipi, K.: Feature engineering in big data for detection of information systems misuse. In: Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering, pp. 145–156. IBM Corp. (2018)
30. Pang, G., Ting, K.M., Albrecht, D., Jin, H.: Zero++: harnessing the power of zero appearances to detect anomalies in large-scale data sets. J. Artif. Intell. Res. **57**, 593–620 (2016)
31. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
32. Pritom, M.M.A., Li, C., Chu, B., Niu, X.: A study on log analysis approaches using sandia dataset. In: 26th ICCCN, pp. 1–6 (2017)
33. Rayana, S.: Odds library. Stony Brook,-2016. Department of Computer Science, Stony Brook University, NY (2016). http://odds.cs.stonybrook.edu (2017)
34. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001)

35. Siadati, H., Saket, B., Memon, N.: Detecting malicious logins in enterprise networks using visualization. In: 2016 IEEE Symposium on Visualization for Cyber Security (VizSec), pp. 1–8. IEEE (2016)
36. Silver-Greenberg, J., Goldstein, M., Perlroth, N.: JPMorgan chase hack affects 76 million households (2014)
37. Sundarkumar, G.G., Ravi, V., Siddeshwar, V.: One-class support vector machine based undersampling: application to churn prediction and insurance fraud detection. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–7. IEEE (2015)
38. Swersky, L., Marques, H.O., Sander, J., Campello, R.J., Zimek, A.: On the evaluation of outlier detection and one-class classification methods. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–10. IEEE (2016)
39. Turcotte, M., Moore, J., Heard, N., McPhall, A.: Poisson factorization for peer-based anomaly detection. In: 2016 IEEE Conference on Intelligence and Security Informatics (ISI), pp. 208–210. IEEE (2016)
40. Van Erp, M., Vuurpijl, L., Schomaker, L.: An overview and comparison of voting methods for pattern recognition. In: Proceedings. Eighth International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 195–200. IEEE (2002)
41. Wu, K., Zhang, K., Fan, W., Edwards, A., Philip, S.Y.: RS-forest: a rapid density estimator for streaming anomaly detection. In: 2014 IEEE International Conference on Data Mining, pp. 600–609. IEEE (2014)
42. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Network representation learning: A survey. IEEE transactions on Big Data (2018)
43. Zhao, Y., Nasrullah, Z., Hryniewicki, M.K., Li, Z.: LSCP: Locally selective combination in parallel outlier ensembles. In: Proceedings of the 2019 SIAM International Conference on Data Mining, pp. 585–593. SIAM (2019)
44. Zimek, A., Campello, R.J., Sander, J.: Ensembles for unsupervised outlier detection: challenges and research questions a position paper. ACM SIGKDD Explor. Newsl. **15**(1), 11–22 (2014)
45. Zimek, A., Schubert, E.: Outlier detection. In: Liu, L., Özsu, M. (eds.) Encyclopedia of Database Systems, pp. 1–5. Springer, New York (2017). https://doi.org/10.1007/978-1-4899-7993-3