Unsupervised Anomaly Detection in Online Game

Trung Thanh Nguyen
Le Quy Don Technical University
and GRD - VNG Corporation
thanhnt@vng.com.vn

Anh Tuan Nguyen,
GRD - VNG Corporation
anhnt7@vng.com.vn

Tuan Anh Ha Nguyen
GRD - VNG Corporation
tuannha@vng.com.vn

Ly Thi Vu,
Le Quy Don Technical University
lyvt@mta.edu.vn

Quang Uy Nguyen
Le Quy Don Technical University
quanguyhn@gmail.com

Long Dao Hai
GRD - VNG Corporation
longdh@vng.com.vn

ABSTRACT

Online game is one of the most successful business on the Internet. As online game business grows, cheating in game becomes popular and is the biggest challenge of online game systems. In this paper, we investigate the application of anomaly detection techniques to cheating detection in an online game (JX2) of VNG company. A method to evaluate the performance of unsupervised anomaly detection techniques was proposed. Six unsupervised anomaly detection algorithms were tested. The experimental results show that the kernel density based technique and ensemble techniques performed best on this game data. Our post analysis helped to identify and eliminate some cheating players in the game.

CCS Concepts

•Information systems → Massively multiplayer online games; Data mining; Clustering; *Social networking sites; Decision support systems;* **•Human-centered computing →** Social networks;

Keywords

Anomaly Detection; Online Game; Unsupervised Learning

1. INTRODUCTION

Online game is one of the most profitable businesses on the Internet nowadays. As the border between online and real economies are mixed, cheating in games has grown rapidly [12]. Due to cheating in online game, the user's sensitive information is at the risk of leakage. Players might get unfair in game sets, receive unexpected advertisements or even lose money [13]. Thus, detecting and preventing cheating in online game is of great vital to create fair games and protect players.

By analysing and detecting abnormal behavior of players, one can identify the players who are likely cheaters. In online games, abnormal users are game players who illegally use hacking tools or cheating methods to get higher advantage than their competitors in terms of win ratio, money, items, upping levels [25]. This decreases the willingness of users to play games since they can not win against someone with hidden weapons. Therefore, detecting abnormal users is an important way to protect kind users [12].

In machine learning, detecting anomaly has received great attention from the research community [5]. Anomaly detection aims to find samples in data that do not follow the expected behavior. These samples are often referred to as anomalies or outliers. Two these terms are often used interchangeably. Anomaly detection techniques have extensively been applied to a wide variety of applications such as fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security [17].

Detecting anomalies in data has been studied in the early 19th century by statisticians. Overtime, a variety of anomaly detection techniques have been developed for diverse application domains [5]. However, applying anomaly detection methods to online games was still under-examined [12]. In this paper, we investigate the effectiveness of various unsupervised anomaly detection techniques in an online game (JX2). We proposed a method for comparing the performance of different unsupervised anomaly detection techniques. Since, labeled data is not available, evaluating unsupervised anomaly detection methods has been a challenging task until recently [30]. Our proposed method for evaluating the effectiveness of anomaly detection approaches is based on applying a classification algorithm to the output of anomaly detection. The main contributions of the paper are:

- A method for evaluating the performance of unsupervised anomaly detection techniques was proposed.
- Six anomaly detection techniques was applied to an online game. Their effectiveness was evaluated based on the proposed measure. The best techniques were determined and used in building a real anomaly detection system at VNG company.

The remaining of this paper is organized as follows. In the next section, we introduce some popular anomaly detection techniques. Our method for measuring the performance of

SoICT 2015, December 03-04, 2015, Hue City, Viet Nam
© 2015 ACM. ISBN 978-1-4503-3843-1/15/12...$15.00
DOI: http://dx.doi.org/10.1145/2833258.2833305

unsupervised anomaly detection algorithms and six anomaly detection techniques used in the paper are presented in Section 3. Section 4 describes the experimental setup. The results of applying anomaly detection techniques to the online game are presented in Section 5. Section 6 concludes the paper and highlights some future work.

2. BACKGROUNDS

Anomaly detection has been the topic of a great deal of research. Many anomaly detection techniques have been developed for various application domains. For a comprehensive review of the research on anomaly detection, the readers are recommended to read [5]. In this section, some popular techniques are described. Regarding to the availability of labeled data, anomaly detection techniques are classified into the following three classes [5].

Supervised anomaly detection: These methods assume that labeled instances for both normal and anomaly class are available during the training process. The objective is to build a predictive model for normal vs. abnormal classes. The model is then used to determine which class an unseen data instance belong to [26, 22]. Although, the performance of supervised methods are often robust compared to others, obtaining labeled data is strenuous and expensive. Consequently, supervised anomaly detection techniques were not used as frequently as unsupervised methods.

Semi-Supervised anomaly detection: Semi-supervised techniques reply on the assumption that there is only one class of instances (often normal class) in the training data. These methods are more widely applicable than supervised techniques since abnormal instances are not required in the training phase. The typical approach is to construct a model for normal behavior, and use the model to determine anomalies in the test data. Popular anomaly detection techniques based on one class learning include one class support vector machine [19], one-class Kernel Fisher Discriminants [20] and artificial immune system [23].

Unsupervised anomaly detection: These are the most widely applicable techniques since labeled samples are not required for both classes. A large number of unsupervised anomaly detection approaches have been proposed. Among them, nearest neighbor based techniques, clustering based techniques and statistical techniques have been widely applied.

Nearest neighbor based techniques assume that the density in normal region is higher than in abnormal region. Usually, the distance of a data instance to its k^{th} nearest neighbor is considered as the anomaly score. If this distance is too far (greater than a threshold) then the data sample is considered anomaly [4, 29]. The advantage of nearest neighbor based techniques is that they are fully unsupervised. Furthermore, nearest neighbor based techniques do not make any assumptions regarding the generative distribution for the data. Nevertheless, the computational complexity of the techniques is often high ($O(N^2)$, N is the number of samples). This hinders the application of nearest neighbor based techniques to some real world applications where time constraint is critical.

Clustering based techniques reply on the assumption that normal data instances belong to large and dense clusters, while anomalies belong to small or spare clusters. The techniques use clustering algorithms to divide the dataset and report any data instance that does not belong to any cluster or belongs to the clusters with a small number of samples as anomalous [28, 15]. Similar to nearest neighbor based techniques, clustering based techniques can operate in a fully unsupervised mode. However, the computational complexity for clustering the data is challenging especially for the algorithms such as hierarchical clustering [24] where the complexity is $O(N^2)$.

Statistical anomaly detection assumes that normal data instances are generated by a stochastic model, any declare any sample with low probability of generated from the model as anomalous. The techniques estimate a statistical model from the given data and then apply the learnt model to test if an unseen instance belongs to this model or not. Two classes of statistical models: parametric and non-parametric have been developed for anomaly detection [8, 3]. The advantage of statistical techniques is that the complexity of fitting data is low (often linear). Consequently, statistical techniques have extensively been used in variety of real world problems. However, statistical approaches rely on the assumption that the data is generated from a particular distribution. If this assumption does not hold, the performance of statistical methods might suffer.

In the next section, we present six unsupervised anomaly detection algorithms that were implemented in our experiments. Unsupervised techniques were selected since in online game applications, labeled samples are unavailable.

3. METHODS

In this section, we present the proposed method for evaluating the performance of unsupervised anomaly detection techniques. After that, six anomaly detection techniques used in the experiments are described.

3.1 Performance Evaluation of Unsupervised Anomaly Detection

Evaluating the performance of unsupervised anomaly detection has been a challenging task in the research community [30]. Our proposed method is based on the assumption that if an anomaly detection technique performs efficiently on a dataset, then the abnormal samples detected by this method must be well separated from the normal samples. Consequently, a relevant classification algorithm should perform efficiently on this dataset after the abnormal and normal instances are labeled with the corresponding labels. In other words, we can measure the performance of supervised algorithms and use this measure to compare the performance of different unsupervised anomaly detection.

Precisely, assume that the dataset includes N samples and two anomaly detection approaches namely A and B are applied to this dataset. Na_1 and Na_2 are the abnormal and normal samples detected by algorithms A. Similarly we have Nb_1 and Nb_2 for algorithm B. Assume that algorithm A performs better than algorithm B on this dataset, then the performance of a classification algorithm C on the dataset of $Na_1 + Na_2$ (combining $Na1$ and $Na2$ into one after labeling them) will be higher than the performance of C on $Nb_1 + Nb_2$.

The selection of the classification technique is important for the efficiency of this method. The selected technique should be able to process non-linear separation data. Moreover, the capability of handling imbalance data is also required. In this paper we used logistic regression to classify the results of anomaly detection algorithms. Logistic regression has successfully been applied to many non-linear separation classification problems [6]. Sampling with replacement method was used to handle imbalanced data [2]. The objective is to replicate abnormal samples so that the abnormal class has the same number of samples as normal class. The performance of logistic regression is measured by F-score defined as:

$$F - score = 2.\frac{Precision.Recall}{Precision + Recall} \qquad (1)$$

where

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \qquad (2)$$

and

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \qquad (3)$$

3.2 Random Walks

Random Walks (RW) algorithm is a popular method using in information retrieval problems[10]. In RW, data are represented as a stochastic graph where the nodes are objects and the edge represents the relation between two objects. In interactive games, assuming that there are N ($N \geq 2$) players competing in the game, then an user i is represented as a node and a directed edge from user i to user j represents the interaction between user i and j. This connection could be the events such as winning games, money exchanges, items exchanges etc.. The weight of the edge would be the ratio of money exchanges, items exchanges or winning/losing ratio.

After the graph G of interactions in the game is constructed, we compute the transition probability matrix (M) for graph G using above information. In matrix M, each element $r_{i,j}$ is the probability that there will be an interaction between user i and j. Using Power Iteration method [27], the principle eigenvector of matrix M is obtained. The value of the principle eigenvector is used to determine the abnormal degree of objects. Moonesinghe et al. [9] proposed Outrank algorithm to detect outliers based on the random walk model. Outrank algorithm is detailed in Algorithm 1. In Algorithm 1, M is the transition matrix and β is the parameter used to avoid dead-end problem in the graph [9].

Algorithm 1 Outlier-Ranking; Input: information transformation matrix called as $transInf$ in size $n \times n$, error tolerance ϵ

```
for each integer $i$ in $n$ objects do
    $sumInf = 0.0$
    for each integer $i$ in $n$ objects do
        $sumInf = sumInf + transInf(i, j)$
    end for
    for each integer $j$ in $n$ objects do
        $M_{ij} = transInf(i, j)/sumInf$
    end for
end for
$\beta = 0.85$, $t = 0$, $r_0 = (\frac{1}{n})_1$
while ($\sigma \leq \epsilon$) do
    $r_{t+1} = \beta * M * r_t + \frac{1-\beta}{n}$
    $\sigma = \|r_{t+1} - r_t\|$
    $t = t + 1$
end while
Rank $r_t$ from $\max r_t$ to $\min r_t$
```

3.3 Local Outlier Factor

The Local Outlier Factor (LOF) algorithm proposed by Markus et al. [4] is a method for finding outliers in a multidimensional dataset based on object density. The idea of LOF is to calculate the density of a sample locally instead of globally. This local density is then considered as the degree to which an object is abnormal. The detail of LOF algorithm is presented in Algorithm 2.

Algorithm 2 Process of LOF algorithm

```
1. Calculate the $k - distance$ of object $p$ ($k - distance(p)$) as distance between $p$ and its $k^{th}$ nearest neighbour.
2. Find the set of k-nearest neighbours of $p$ as
       $N_k(p) = q \subseteq D \setminus p \mid d(p, q) \leqslant k - distance(q)$
3. For each object $o \subseteq D$, calculate the reachability distance as
       $rd_k(p, o) = max\{k - distance(o), d(p, o)\}$
4. Compute the local reachability density of $p$ as
       $l_k(p) = \left( \frac{\sum_{o \subseteq N_k(p)} rd_k(p, o)}{\mid N_k(p) \mid} \right)^{-1}$
5. Calculate the local outlier factor (LOF) value
       $LOF_k(p) = \frac{\sum_{o \subseteq N_k(p)} \frac{l_k(o)}{l_k(p)}}{\mid N_k(p) \mid}$
6. Sort objects $p$ in decreasing order of the $LOF$ value.
```

In Algorithm 2, LOF value of object p is computed as the average ratio of local reach density $l_k(p)$ and k-nearest neighbors. The ratio between $l_k(p)$ of p to those of p's k-nearest neighborsis lower meaning that the point p is far away from its nearest cluster and the higher the LOF value of p is. Therefore, the LOF value represents the degree of object being an outlier.

3.4 Weighted Rank based detection algorithm

Weighted Rank based detection algorithm (RADA) was recently proposed by Huang et al. [11]. The motivation of RADA is to combing a ranked based techniques with the density methods like LOF. RADA was tested on some outlier detection problems and its performance was robust [11]. RADA algorithm is described in Algorithm 3.

3.5 Kernel Density Estimation

Kernel Density Estimation (KDE) [14], is a non-parametric method to estimate the density of data samples in a dataset. A sample with low density indicates its rarity in the dataset and can be abnormal. The density of data x is calculated by the following equation:

$$KDE_h(x) = \frac{1}{n} \sum_{p} K_h(x - p) \qquad (4)$$

where p are data points around x that are determined by the Nearest Neighbour method and K_h is a kernel function.

Choosing a suitable kernel is important for the performance of KDE. The chosen kernel will determine the shape of the joint distribution. In this paper, we selected Gaussian kernel to detect abnormal users in online game. The

Algorithm 3 Description of RADA algorithm

```
1. Calculate the $k-distance$ of object $p$ ($k-distance(p)$) as distance between $p$ and its $k^{th}$ nearest neighbour.
2. For $p \subseteq D$, Find the set of $k$ nearest neighbors of the object $p$ as $N_k(p)$
       $N_k(p) = q \subseteq D \setminus p \mid d(p,q) \leqslant k - distance(q)$
3. Calculate $d(q,o)$ for all $o \in D - \{q\}$ and find the rank of $d(q,p)$ in this set. Let this be $r_q(p)$.
4. Compute the outlierness of $p$, as follows:
       $O_k(p) = \frac{\sum_{q \subseteq N_k(p)} r_q(p)}{\mid N_k(p) \mid}$
5. Measure the outlineness of $p$ with weight
       $W_k(p) = O_k(p) \times \frac{\sum_{q \subseteq N_k(p)} d(p,q)}{\mid N_k(p) \mid}$
6. Rank p based on $W_k(p)$.
```

Gaussian kernel is defined as:

$$K_{gaussian,h}(u) = \frac{1}{(2\pi)^{\frac{d}{2}}\, h^d} e^{-\frac{u^2}{2h^2}} \qquad (5)$$

where h is the bandwidth of kernel; d is the dimension of data.

3.6 LOF-Ensemble and KDE-Ensemble

This section presents two ensemble methods for anomaly detection that are based on LOF and DKE. These methods are similar to the ensemble method proposed by Gao et al. [7]. The first method called LOF-Ensemble (LOF-E) is the combination of LOF and RADA algorithm for outlier detection. In this method, LOF was applied to the whole dataset. A sub-dataset of LOF output with the highest score was extracted. This sub-dataset was then be re-ranked by RADA to detect final outliers. Algorithm 4 presents the LOF-Ensemble method in detail.

Algorithm 4 Description of LOF-Ensemble method

```
1. Apply LOF for whole sample dataset $D$.
2. Based on the results of LOF, extract t samples with highest probability of being abnormal to the sub-set, $D'$
3. Compute the outlierness of samples in $D'$ set by the Algorithm 3.
```

The second ensemble method called KDE-Ensemble (KDE-E) is the combination of KDE and RADA algorithm. The algorithm for KDE-Ensemble is similar to LOF-Ensemble with the difference is LOF algorithm be replaced by KDE algorithm in the first step of Algorithm 4.

4. EXPERIMENTAL SETTINGS

This section presents the experimental settings in this paper. Subsection 4.1 presents the architecture of the anomaly detection system. The system was designed to run and achieve good performance in a big data environment. Subsection 4.2 presents the way to collect data and two sets of experiment.

4.1 Computing System Architecture

Achieving high performance is critical in a real anomaly detection system. We used map-reduced framework [1] to increase the performance of the system. Moreover, an in-house storage system was implemented to store historical actions of game players. The in-house storage is a distributed B+Tree based on a key-value storage called ZDB [16]. Each user has a long list of actions that can be queried by its position or time range. This long list was stored in multiple servers and managed by a distributed B+Tree. The computing architecture is shown in Fig 1.

In this system, raw logs from game server was daily pushed to *Log Processor* component. The log data was processed and information about the interaction between users was extracted. Then, user information was saved into *Log Store Service*. Log Store Service is built based on a distributed structured storage called Big Set that was built on ZDB key-value store [16]. This component increased the availability and scalability of the structured storage system. The core component of the architecture is *Anomaly Detection Service*. This service applied anomaly detection techniques to find abnormal users. Anomaly detection component was designed so that new detection techniques could comfortably be added to the system. Finally, the detected results were stored into a database and visualized for post analysis.

4.2 Data Collection

Anomaly detection methods were applied to detect cheating users in JX2 game. JX2 is the code name of "Vo Lam Truyen Ki 2" game. This is one of the top game of VNG Corporation with 20 millions registered users and more than 5 millions active users. This is a massive multiplayer online role-playing game. For this game, the number of items bought and sold and the amount of money gained and lost by users are important characteristics. Thus, four attributes (number of items bought and sold, the amount of money gained and lost) were extracted from game log system for each user. These values were then normalized into range of [0,1] by applying the following equation:

$$V_{new} = \frac{V_{old} - min(V)}{max(V) - min(V)} \qquad (6)$$

where V_{new} is the value of V_{old} after normalized and max and min are the maximum and minimum values of the attribute V in the dataset.

The setting for the parameters is as follows. Euclidean distance is used for calculating distance between data samples in all tested algorithms. The number of neighnors (k) in LOF and RADA is 20. The bandwidth of Gaussian kernel (h) in KDE is automatically determined using the technique by Shimazaki et al. [21]. In LOF-Ensemble and KDE-Ensemble, 20% samples with the highest rank by the first algorithm (LOF or KDE) was selected for the second algorithm (RADA). We divided the experiments into two sets. The first set aims to verify the reliability of proposed method for evaluating unsupervised anomaly detection described in Subsection 3.1. For this we selected three datasets of classification problems where labeled data is available. Logistic Regression was used to classify these datasets and F-score was reported. The second experiment was used to evaluate the performance of six anomaly detection algorithms on JX2 game.

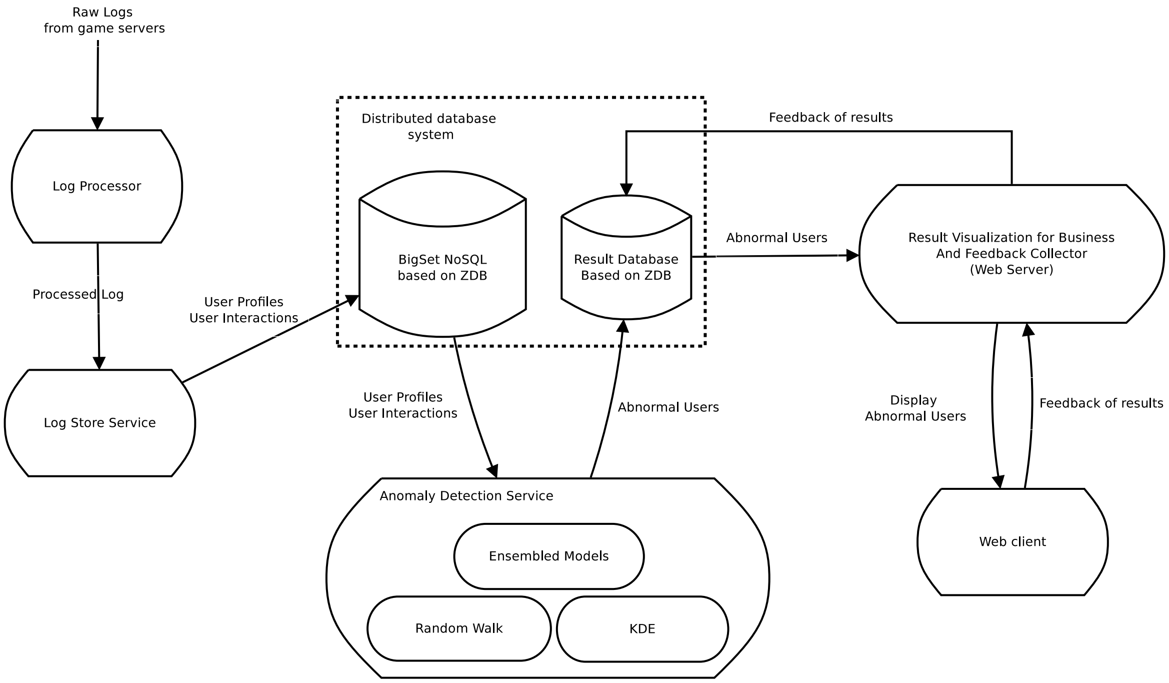

Figure 1: Computing System Architecture.

5. RESULTS AND DISCUSSION

This section presents the results of our experiments. First, the proposed method to evaluate the performance of unsupervised detection was verified. Then, the efficiency of six detection techniques was analysed. Post analysis on the output of anomaly detection is presented at the end of this section

5.1 Verifying the measure

In order to evaluate the reliability of the method proposed in Subsection 3.1, we tested this method on some supervised anomaly detection problems. The tested problems include three real world datasets:

- Packed Executable dataset (PEC): This dataset was collected from the Malfease Project [18] to classify the non-packed executable (viruses) from packed executable (normal files). In our experiments, we selected 2300 packed executable as normal observations, and added 100 non-packed executable as anomalies. Each sample in this dataset has 8 features.

- Wisconsin Dataset (WBC): The Wisconsin breast cancer dataset was downloaded from UCI machine learning dataset. WBC contains 699 instances and has 9 attributes. Among 699 samples, there are 458 benign and 241 malignant samples. In the experiment, we kept only 50 malignant samples to simulate imbalanced problem.

- Wisconsin Diagnostic Breast Cancer (WDBC): This dataset describes nuclear characteristics for breast cancer diagnosis. The dataset includes 569 samples (357 benign, 212 malignant) and has 31 attributes. We kept 40 malignant samples to create imbalanced issue.

Table 1: F-score with different datasets of increasing difficulty

Problems	Swap-0	Swap-10	Swap-20	Swap-30
PEC	0.990	0.986	0.972	0.964
WBC	0.949	0.875	0.850	0.782
WDBC	0.946	0.917	0.897	0.880

After selecting the datasets, logistic regression was used for classifying. Over-sampling technique was applied to balance the number of samples in two class. The performance of logistic regression on each problem was measured by F-score. We then swapped 10 samples from normal to abnormal set and 10 samples from abnormal to normal set. Similarly, 20 and 30 samples in each set was swapped to increase the difficulty of the classification problem. The result of classifying these datasets (measured by F-score) using logistic regression is presented in Table 1

It can be seen from Table 1 that using classification method is a reliable way for measuring the separability of two datasets. Apparently, when the datasets were well separated (the original dataset without swapping, Swap-0), the performance of logistic regression is highest. This is presented by the greatest value of F-score on Swap-0.

When the number of samples swapped from a class to another increased, the performance of the classification algorithm decreased. We can see that F-score of Swap-10 is smaller than Swap-0, Swap-20 is smaller than Swap-10 and Swap-30 is smaller than Swap-20. Overall, the results in this subsection show that the performance of classification algorithms on the output of anomaly detection is a reliable measure for the accuracy of anomaly detection algorithms. In the following subsection, this method is used to evaluate the performance of anomaly detection techniques in online game.

5.2 Comparing anomaly detection techniques

This subsection presents the results of applying six anomaly detection techniques to an online game (JX2). We collected data from log database of JX2 over seven days (from June 25th 2015 to June 31st 2015). For each day, about 20000 samples (users) were collected. Six anomaly detection techniques were used to detect abnormal users in each day. The data sample scored by anomaly detection methods were ranked and 1% users with the highest probability of anomaly was reported as the potential cheating users. The abnormal portion (1%) and the rest (99%) were then labeled. Over-sampling technique was applied and logistic regression was used to classify this dataset. F-score was calculated and they are presented in Table 2. In this table, Day-1 to Day-7 are shorted for days from June 25th 2015 to June 31st 2015, respectively. Additionally, the best result is printed bold faced.

Table 2: Result of six anomaly detection techniques measured by F-score using logistic regression

Data	RW	KDE	LOF	RADA	LOF-E	KDE-E
Day-1	0.154	**0.610**	0.013	0.019	0.528	0.512
Day-2	0.162	**0.682**	0.184	0.0132	0.600	0.542
Day-3	0.244	**0.700**	0.187	0.009	0.600	0.628
Day-4	0.266	**0.607**	0.243	0.000	0.521	0.559
Day-5	0.136	**0.650**	0.104	0.030	0.517	0.547
Day-6	0.256	0.643	0.068	0.037	**0.695**	0.674
Day-7	0.183	0.735	0.080	0.005	0.689	**0.800**

It can be observed from Table 2 that KDE is often the best technique among six tested methods. Obviously, KDE achieved the best results in five out of seven days. Perhaps, the ability to handle various distributions aids KDE in achieving good performance on this problem. Conversely, the performance of three single techniques (RW, LOF and RADA) was much worse than KDE. Particularly, F-score of these method are often close to zero. This means that most of the abnormal samples detected by these methods is misclassified by logistic regression.

Comparing single methods with ensemble methods, the table shows that two ensemble methods achieved convincing results. Although, ensemble methods was mostly equal to KDE, they were often by far better compared to other single approaches. Interestingly, though both LOF and RARA did not perform well on this problem, the combination of LOF and RADA (LOF-E) performed much better than single algorithms. These results motivate the future research in developing ensemble techniques in anomaly detection applications.

5.3 Post Analysis

After applying anomaly detection algorithms to JX2, we conducted a post analysis to verify the results of detection techniques. A brief result is presented in Table 3. In this table, top five anomalous users marked by KDE in Day-1 (25/06/2015) and their features are reported. For the sake of comparison, the average values of each feature over all active users on that day are also presented in the last row. Note that, all the features are normalized into $[0, 1]$ and m. and i. are shorted for money and item respectively. It can be seen that KDE has successfully filtered out the anomalous users. For instance, User 1 received more money than every active users on the selected day. Comparing to the average income of the whole dataset, we found that User 1 received 480 times higher than the expected amount of a normal user.

Table 3: Top five users filtered by KDE on Day-1

UID	m. spent	m. received	i. sold	i. bought
1	0.15963	**1.00000**	0.00111	0.00043
2	0.01780	0.00000	0.00041	**1.00000**
3	0.01228	0.00000	**1.00000**	0.00275
4	**1.00000**	0.00000	0.00062	0.00000
5	**0.94169**	0.02551	0.00103	0.00168
Average	0.00634	0.00209	0.00089	0.00093

Similarly, User 2 and 3 were marked because they bought and sold more items than anyone else in the collected dataset reaching more than 1000 times higher than the average. There is a high probability that these patterns are the results of a "boosting" situation [1] in online game. Although User 4 and 5 do not look as suspicious as the others, they were reported since they spent too much money in one day.

6. CONCLUSIONS AND FUTURE WORK

The problem of detecting abnormal users in an online game (JX2) was examined in this paper. We proposed a method for evaluating the performance of different unsupervised anomaly detection techniques. After that, the efficiency of various anomaly detection algorithms was investigated. The experimental results aided the selection of the most appropriate methods in constructing a real anomaly detection system for game at VNG Corporation.

There are some research areas which arise from this paper. First, we would like to apply the tested anomaly detection methods to other games to better understand their performance. Second, we want to combine clustering methods with the techniques in this paper to determine exact number of abnormal users for each dataset. In this paper, the number of suspicious users was reported as 1% users with the highest probability of anomaly. However, this might not be accurate since the number of abnormal users might be various overtime. At the theoretical level, we would like to further examine more sophisticated methods to evaluate the performance of unsupervised anomaly detection algorithms.

Acknowledgements

This research was funded by VNG Corporation.

7. REFERENCES

[1] F. N. Afrati and J. D. Ullman. Optimizing multiway joins in a map-reduce environment. *IEEE Trans. Knowl. Data Eng*, 23(9):1282–1298, 2011.

[2] M. Bekkar and T. A. Alitouche. Imbalanced data learning approaches review. 2013.

[1] The situation where a player has multiple accounts to support the main account in getting achievements

[3] M. Bouguessa. A mixture model-based combination approach for outlier detection. *International Journal on Artificial Intelligence Tools*, 23(4), 2014.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proc. ACM SIGMOD 2000 International Conference on Management of Data, Dalles, TX*, 2000.

[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computating Surveys*, 41(3), 2009.

[6] R. Christensen. *Log-Linear Models and Logistic Regression*. Statistics. Springer Verlag, 1997.

[7] J. Gao, W. Hu, Z. Zhang, and O. Wu. Unsupervised ensemble learning for mining top-n outliers. In *Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, Proceedings, Part I*, volume 7301 of *Lecture Notes in Computer Science*, pages 418–430. Springer, 2012.

[8] M. Gebski and R. K. Wong. An efficient histogram method for outlier detection. In *Advances in Databases: Concepts, Systems and Applications, 12th International Conference on Database Systems for Advanced Applications, DASFAA 2007, Bangkok, Thailand, April 9-12, 2007, Proceedings*, volume 4443 of *Lecture Notes in Computer Science*, pages 176–187. Springer, 2007.

[9] P. N. T. H. D. K. Moonesinghe. Outrank: A graph-based outlier detection framework using random walk. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.

[10] J. He, H. Tong, M. Li, W.-Y. Ma, and C. Zhang. Multiple random walk and its application in content-based image retrieval. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR 2005, November 10-11, 2005, Singapore*, pages 151–158. ACM, 2005.

[11] H. Huang, K. Mehrotra, and C. K. Mohan. Rank-based outlier detection. In *Electrical Engineering and Computer Science Technical Reports*, 4 2011.

[12] A. B. Jeng and C. L. Lee. A study on online game cheating and the effective defense. In *Recent Trends in Applied Artificial Intelligence, 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2013, Amsterdam, The Netherlands, June 17-21, 2013. Proceedings*, volume 7906, pages 518–527. Springer, 2013.

[13] A. R. Kang, J. Woo, J. Park, and H. K. Kim. Online game bot detection based on party-play log analysis. *Computers & Mathematics with Applications*, 65(9):1384–1395, 2013.

[14] J. Kim and C. D. Scott. Robust kernel density estimation. *The Journal of Machine Learning Research*, 13(1):2529–2565, 2012.

[15] E. Leon, O. Nasraoui, and J. Gomez. Anomaly detection based on unsupervised niche clustering with application to network intrusion detection. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 502–508, Portland, Oregon, 20-23 June 2004. IEEE Press.

[16] T. Nguyen and M. Nguyen. Zing Database: high-performance key-value store for large-scale storage service. *Vietnam Journal of Computer Science*, 2(1):13–23, 2015.

[17] A. Patcha and J.-M. P. 0001. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[18] R. Perdisci, A. Lanzi, and W. Lee. Classification of packed executables for accurate computer virus detection. *Pattern Recognition Letters*, 29(14):1941–1946, 2008.

[19] G. Ratsch, S. Mika, B. Scholkopf, and K. R. Muller. Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(9):1184–1199, Sept. 2002.

[20] V. Roth. Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4):942–960, 2006.

[21] H. Shimazaki and S. Shinomoto. Kernel bandwidth optimization in spike rate estimation. *Journal of Computational Neuroscience*, 29(1-2):171–182, 2010.

[22] S. Singh and M. Markou. An approach to novelty detection applied to the classification of image regions. *IEEE Trans. Knowl. Data Eng*, 16(4):396–407, 2004.

[23] T. S. Sobh. Anomaly detection based on hybrid artificial immune principles. *Information Management and Computing Security*, 21(4):288–314, 2013.

[24] G. J. Székely and M. L. Rizzo. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *J. Classification*, 22(2), 2005.

[25] R. Thawonmas, M. Kurashige, and K.-T. Chen. Detection of landmarks for clustering of online-game players. *IJVR*, 6(3):11–16, 2007.

[26] W.-K. Wong, A. Moore, G. Cooper, and M. Wagner. Bayesian network anomaly pattern detection for detecting disease outbreaks. In *Proceedings of the 20th International Conference on Machine Learning*, pages 217–223, 2003.

[27] E. G. Yakonov. Iteration methods in eigenvalue problems. *Mathematical Notes*, 34:945–953, 1983.

[28] D. Yu, G. Sheikholeslami, and A. Zhang. Findout: Finding outliers in very large datasets. *Knowledge and Information Systems*, 4(4):387–412, 2002.

[29] J. Zhang and H. H. Wang. Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge Information Systems*, 10(3):333–355, 2006.

[30] A. Zimek, R. J. G. B. Campello, and J. Sander. Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *SIGKDD Explorations*, 15(1):11–22, 2013.