The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Chunk-Based Incremental Classification of Fraud Data

Farzana Anowar, Samira Sadaoui
Department of Computer Science
University of Regina
3737 Wascana Pkwy,
Regina, SK S4S 0A2
{anowar, sadaouis}@uregina.ca

Abstract

Shill Bidding (SB) is still a predominant auction fraud because it is the toughest to identify due to its resemblance to the standard bidding behavior. To reduce losses on the buyers' side, we devise an example-incremental classification model that can detect fraudsters from incoming auction transactions. Thousands of auctions occur every day in a commercial site, and to process the continuous rapid data flow, we introduce a chunk-based incremental classification algorithm, which also tackles the imbalanced and non-linear learning issues. We train the algorithm incrementally with several training SB chunks and concurrently assess the performance and speed of the new learned models using unseen SB chunks.

Introduction

Auctions have become an ever-growing e-commerce marketplace. Nevertheless, this popularity also means that dishonest sellers will take any chance to conduct illicit activities to increase their revenues. According to the report of the Internet Crime Complaint Center of FBI, auction fraud represents one of the topmost cyber-crimes (Anowar and Sadaoui 2020). Auction fraud is made possible due to three main factors: user anonymity, bidding flexibility, and low auction fees. Auctions are vulnerable to different types of fraud. In particular, Shill Bidding (SB) is one of the leading frauds because it is the most difficult to detect. SB does not leave any discernible evidence, unlike the other types of fraud, and may look similar to the normal bidding behavior (Ford, Xu, and Valova 2012). To increase the seller's pay-off, a shill bidder (the seller himself or an accomplice) takes advantage of the ongoing bidding session to raise the price of the product by submitting artificial bids via phony accounts. Several research studies empirically demonstrated the presence of SB activities in different commercial auction websites (Ford, Xu, and Valova 2012). Moreover, SB could result in substantial financial losses for genuine buyers, as seen in several lawsuits, (Anowar, Sadaoui, and Mouhoub 2018). For all these reasons, it becomes vital to analyze the behavior of bidders to detect SB, and hence prevent innocent buyers from becoming victims.

On eBay, thousands of auctions are carried out daily. This commercial activity generates a vast volume of bidding transactions. Undoubtedly, this volume will create considerable challenges for detecting SB fraud. Furthermore, since bidding data are generated continuously and at high speed, the real-time identification of shill bidders becomes crucial. Incremental learning is the most appropriate approach to address the scalability, time-efficiency, and accuracy of fraud detection models. The most significant difference between traditional and incremental learning is that the latter does not presume the availability of adequate training data; instead, the model receives data over time (Zang et al. 2014). It has been shown that incremental approaches outperform the accuracy and speed of the non-adaptive approaches when classifying new data (Zang et al. 2014) because this type of classifiers can refine its knowledge without re-training from scratch (Bouchachia, Gabrys, and Sahel 2007).

Since labeling multi-dimensional SB data is challenging as demonstrated in (Anowar, Sadaoui, and Mouhoub 2018) and (Alzahrani and Sadaoui 2019), hence, in this study, we begin with a small labeled SB dataset. However, to learn the complete concept of the SB fraud, it is necessary to train the fraud classifier on much more data. Therefore, the example-incremental classification has been the focus of our study to learn gradually from new SB data and improve the detection and misclassification rates in the long run. Auction data arrive continuously, and to be able to detect fraud in each auction and prevent a monetary loss for the winner, we organize SB data into chunks. An SB chunk consists of the bidding behavior of the participants of an auction. So, we aim to design a chunk-based incremental classification algorithm, which also addresses the imbalanced and non-linear learning. Solving the imbalanced learning problem in the fraud detection domain is vital.

To tackle our difficult fraud learning task, it is required to select a classification algorithm that can be trained incrementally and possess a high-speed execution in classifying new unseen data. To this end, the Stochastic Gradient Descent (SGD) has been chosen for two important reasons: 1) it is independent of the size of the training datasets, which is essential in the real-world auction scenarios since chunks come in different sizes w. r. t. the number of bidders in each

auction, and 2) it has few hyper-parameters to be optimized (Losing, Hammer, and Wersing 2016). Based on the SGD algorithm, we develop the chunk-based kernel incremental classifier. However, SGD, a linear model, maybe ineffective when processing the non-linear SB data. So, all the training chunks are converted into linear data before feeding them to the SB classifier. Then we train the proposed algorithm incrementally with several training SB chunks and concurrently assess the detection and misclassification rates and speed of the produced learned models with testing chunks.

Related Works

In this section, recent studies regarding the incremental classification has been reviewed. For binary classification problems, the authors in (Gu et al. 2018) developed a chunk-based incremental learning algorithm (called CICSHL-SVM) using the Cost-Sensitive Hinge Loss. The algorithm adjusts to a chunk of instances at once and is able to ensure the model stability with cost-sensitive Bayes risk. The authors tested the new method with eleven datasets. The experimental results confirm the effectiveness of CICSHL-SVM and also demonstrate that it outperforms the static version and single incremental version of CSHL-SVM.

The study (Vinagre, Jorge, and Gama 2014) developed an incremental matrix factorization method based on Incremental SGD algorithm (ISGD) for the item prediction problem considering only positive feedbacks. Through four datasets, all the experiments reveal that ISGD has a competitive accuracy despite of being simple, and is significantly faster than four known algorithms. Out of four scenarios, ISGD came out as a clear-cut winner for two cases only. For the remaining cases, ISGD provided better speed. Also, the authors only took into account the true values (positively rated items) for training whereas the false values (either users disliked the item or did not interact with it) have been considered as missing values.

Developing an efficient classifier that copes with data stream is a challenging task for the machine learning community. Hence, (Krawczyk and Woźniak 2015) suggested one-class classification as a promising approach to analyze data stream. The authors presented a modification of the weighted one-class SVM (with RBF kernel) for non-stationary data by using two mechanisms: "adaptation mechanism" to adjust the SVM decision boundary to the incoming data and "forgetting mechanism" to ensure limited memory consumption and increase the ability of a classifier to accept new data. The experiments have been carried out with a total of five datasets (two semi-synthetic and three real-world data). Since this research deals with the one-class learning task, the authors utilized only the training instances belonging to the target class. Then, for the testing phase, all the instances have been employed. The proposed method outperforms two other classifiers in terms of run time and accuracy.

Another work (Ford, Xu, and Valova 2012) introduced an SB classifier that can adjust to new auction data based on a feedforward back-propagation ANN using the sliding window concept. If a bidder is predicted as normal, it is then sent to the classifier for the incremental training with a window size of nine, which means the model uses the nine recent bidders to produce the training and validation datasets. If a bidder is categorized as fraud, it is then sent to a verifier (using Shaffer theory), and the corrected data fed to the classifier. Nevertheless, the authors did not consider the effect of class imbalance during training. Also, they classified the bidders based on their involvement in all the auctions. In this case, it is impossible to determine which auctions are being corrupted by fraud, and as a result, the financial loss cannot be prevented. Moreover, we note that ANN can suffer from local minima occasionally and is computationally costly since there are several parameters to be optimized. Besides, ANN cannot explain the behavior of the network and thus can reduce trust.

Initial and Incremental Chunks

In this paper, we employ actual SB data that have been produced from eBay auctions (Alzahrani and Sadaoui 2019). After a rigorous preprocessing operation, the SB dataset consists of 807 auctions, 1054 bidders, 6321 instances, and eight fraud predictors; each instance denotes the conduct of a bidder in an auction. Next, (Anowar, Sadaoui, and Mouhoub 2018) devised a robust two-step labeling approach and obtained 5694 normal instances and 627 suspicious instances.

To conduct the example-incremental learning task, the original SB dataset is divided into training data (70%) and testing data (30%) using the stratified splitting method to make sure to have approximately equal suspicious samples in each subset since the cardinality of the suspicious class is low. Testing data is utilized to validate the new fraud model after each incremental adaptation phase.

Nevertheless, both datasets are imbalanced, and if the SB detection model is built with skewed training data, suspicious instances will be misclassified because learning algorithms will favor the typical class. Also, imbalanced data often degrades the predictive performance of classifiers (Anowar and Sadaoui 2020). So, solving the imbalanced learning problem in the fraud detection domain is crucial. For this purpose, a data sampling method is employed. Since over-sampling might lead to over-fitting and under-sampling might remove essential data, the ensemble of these two methods usually provides better results, as demonstrated in (Anowar, Sadaoui, and Mouhoub 2018) with the hybrid technique SMOTE-ENN. SMOTE over-samples the minority class with synthetic data and ENN under-samples the majority class by deleting instances that form TomekLinks i.e., borderline and noisy instances that lower the predictive performance. We keep in mind that training and testing datasets should have the same characteristics. Hence, SMOTE-ENN has been applied to both datasets but separately so that no information from the training dataset is exposed to the testing dataset. As a result, 7945 training data and 3392 testing data are obtained. We should have enough instances in the test dataset to be able to evaluate the classifier's performance incrementally over several runs.

Next, we split each dataset, training and testing, into five chunks with equal sizes and equal class distributions. Still,

the initial learning chunk should be representative enough to adequately initialize the fraud classifier. The details of all the developed chunks are exposed in Table 1.

Table 1: SB chunks for incremental classification

Initial Training chunk (3925)				
Normal		Suspicious		
1960		1965		
Incremental Training Chunk (1005 each)				
chunk#1		. . .	chunk#4	
Normal	Suspicious	. . .	Normal	Suspicious
500	505	. . .	500	505
Initial Test Chunk (680)				
Normal		Suspicious		
338		342		
Incremental Test Chunk (678 each)				
chunk#1		. . .	chunk#4	
Normal	Suspicious	. . .	Normal	Suspicious
337	341	. . .	337	341

Another problem to be addressed is that real-world SB data are most likely non-linear due to the fact that bidders may change their bidding strategies and the SB features are unique (Ford, Xu, and Valova 2012). In this case, the non-linear data is transformed using a Kernel Map Approximation (KMA) function. There are several kernels, such as Linear, RBF, Polynomial and Sigmoid. RBF is selected since previously it performed much better for the original SB dataset (Anowar, Sadaoui, and Mouhoub 2018).

An Incremental Classification Approach

Our aim is to develop an efficient example-incremental fraud classification model using the SB training chunks presented in Table 1. Incremental learning extends the model's knowledge gradually when data become available (Joshi and Kulkarni 2012). Our SB classifier will be able to learn from new data chunks without preprocessing entirely the training dataset. Hereafter, we describe how to improve a regular classification algorithm to tackle the incremental, chunk-based, imbalanced and non-linear learning. As a machine learning toolkit, Scikit-learn is employed because it offers a meta-estimator that supports the incremental classification. In Algorithm 1, the steps presented are necessary to fully implement any chunk-based kernel incremental classification. First, we need to select a classification algorithm (called estimator) that supports the partial fitting to be able to process data as chunks. Now, to transform the selected algorithm into an incremental one, we connect (wrap) it with the incremental meta-estimator. Before conducting any training, we first re-balance the SB chunks using a hybrid data sampling method and then convert them into linear data using a kernel function. Next, the classifier is initialized with the first balanced linear training chunk. We keep conducting the example-incremental adaption until the SB concept has been fully learned, which means until a classification performance very close to 100% has been achieved.

To conduct the empirical analysis in the next section, we customize Algorithm 1 with an appropriate classification technique. There are several learning algorithms that support the partial fitting, such as Naive Bayes, Multi Layer Perceptron, SGD, SVM and PassiveAggressive. In our work, SGD is chosen because it performs much better than several other classifiers in terms of run time and accuracy (Vinagre, Jorge, and Gama 2014), (Diaz-Aviles et al. 2012) and (Read et al. 2012). Lowering the processing time is crucial in the fraud detection domain, so that the classifier responds very fast to incoming chunks. Also, the SGD algorithm is independent from the size of the training chunks, which is important in the real-world scenario since SB chunks come with different sizes w. r. t. the number of bidders in each auction.

Algorithm 1: Chunk-Based Kernel Incremental Classification

// Incremental Library and Estimator Selection
1: Select estimator that supports "PartialFit"
2: Import "Incremental" meta-estimator to connect with "PartialFit"
3: Wrap estimator with Incremental meta-estimator
// Classification Initialization
4: Balance initial training chunk using hybrid data sampling
5: Convert initial chunk to linear data using KMA
6: Fit estimator with initial chunk
// Incremental Classification
7: Repeat until SB concept learned:
{ **7.1:** Balance incremental training chunk using hybrid data sampling
7.2: Convert incremental chunk to linear data using KMA
7.3: Adapt estimator with incremental chunk }

After customizing Algorithm 1, we develop a Chunk-Based Kernel Incremental SGD (called CKISGD) technique. The latter can iterate several times through one data chunk, and in each iteration, it optimizes its loss function and weights. The algorithm minimizes the loss function over a large parameter space; penalty, learningRate, iterationNumber and warmStart being the meta-parameters. "Hinge" is selected as the loss function to develop a linear SVM, and "L2" as the penalty, which is the standard regularizer of linear SVMs. The Hinge function will allow to manipulate the "maximum-margin" of SVM, so that the model produces less training errors. We set the learningRate to "optimal" to regulate the model each time the weights are updated in response to the estimated error. IterationNumber is assigned to 5 i.e., if there is no change in the minimum loss value after five iterations, then CKISGD will terminate. Since we have small training chunks, we believe five iterations are enough. We also set warmStart to "true" to be able to use the parameters' values from the previous iteration when a new data chunk is fed to the SB classifier.

Besides, when the cost function is not converging anymore, SGD provides two stopping criteria that we use for our CKISGD: 1) there is no change in the minimum loss value after five iterations; 2) the validation score is not im-

proving anymore. Furthermore, CKISGD keeps two chunks in memory at a time, and the most recent chunk has more impact on the model training than the previous chunk. So, CKISGD acquires new data rapidly and retains outdated data a little longer. Hence, CKISGD does not suffer from immediate forgetting since it forgets older chunks gradually. Thus, CKISGD achieves a good balance between stability and plasticity.

Table 2: Performance evaluation for testing data

Testing Chunk	F1-score	FNR	Log-loss	Run-time (second)
Initial Chunk	0.972	0.014	0.310	0.006177
Chunk#1	0.989	0.011	0.251	0.005612
Chunk#2	0.990	0.009	0.202	0.005186
Chunk#3	0.992	0.006	0.180	0.004181
Chunk#4	0.997	0.000	0.153	0.005481

Validation

We gradually train the CKISGD algorithm using the initial and incremental training SB chunks presented in Table 1. After each adaptation phase, the performance of the new produced learned model is evaluated using the testing chunks given in Table 1. As testing chunks are fed to CKISGD, bidders are classified and actions can be taken against each infected auction. For validating our SB classification model, we employ four quality metrics: F1-score, False Negative Rate (FNR), Log-loss and Run-time. F1-score measures the effectiveness of detecting the suspicious class and FNR the misclassification rate of suspicious bidders. When the Log-loss value increases, this means the predicted labels are diverging from the actual labels. Therefore, minimising the Log-loss function corresponds to maximizing the accuracy of the classifier. Lastly, Run-time calculates the time the model took to classify unseen data.

From Table 2, it is not surprising to observe that as the number of instances increases, F1-score values augment and FNR and Log-loss values decrease gradually. The initial chunk has the lowest F1-score and the last chunk has the highest F1-score. In contrast, the first chunk has the highest FNR (1.4% of shill bidders have been wrongly classified). We see that the last chunk has a FNR of 0%. Likewise, the Log-loss values decrease as the number of chunks increases. This means that the predicted probability is getting closure to the actual labels of the SB instances. We also notice that our model did not take much time when classifying unseen chunks as the maximum time is very low (0.006177 seconds). We would like to mention that we tried incremental linear SGD but the model returned a low performance.

Conclusion

The e-auction marketplace generates a continuous and rapid flow of bidding transactions. To address the real-world SB detection problem, we developed a classification model that tackles the incremental, chunk-based, non-linear, and imbalanced learning. According to the experimental results, based on the SGD algorithm, the incremental SB classifier was able to improve gradually its detection and misclassification rates after each incremental adaptation. In the fraud detection domain, speed is an essential requirement, and the SB model was able to classify unseen data chunks very fast.

References

Alzahrani, A., and Sadaoui, S. 2019. Instance-incremental classification of imbalanced bidding fraud data. In *11th International Conference on Agents and Artificial Intelligence*, 92–102.

Anowar, F., and Sadaoui, S. 2020. Detection of auction fraud in commercial sites. *Journal of Theoretical and Applied Electronic Commerce Research, Universidad de Talca* 15(1):81–98.

Anowar, F.; Sadaoui, S.; and Mouhoub, M. 2018. Auction fraud classification based on clustering and sampling techniques. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 366–371.

Bouchachia, A.; Gabrys, B.; and Sahel, Z. 2007. Overview of some incremental learning algorithms. In *2007 IEEE International Fuzzy Systems Conference*, 1–6.

Diaz-Aviles, E.; Drumond, L.; Schmidt-Thieme, L.; and Nejdl, W. 2012. Real-time top-n recommendation in social streams. In *6th ACM conference on Recommender systems*, 59–66.

Ford, B. J.; Xu, H.; and Valova, I. 2012. A real-time self-adaptive classifier for identifying suspicious bidders in online auctions. *The Computer Journal* 56(5):646–663.

Gu, B.; Quan, X.; Gu, Y.; Sheng, V. S.; and Zheng, G. 2018. Chunk incremental learning for cost-sensitive hinge loss support vector machine. *Pattern Recognition* 83:196–208.

Joshi, P., and Kulkarni, P. 2012. Incremental learning: Areas and methods-a survey. *International Journal of Data Mining & Knowledge Management Process* 2(5):43–51.

Krawczyk, B., and Woźniak, M. 2015. One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing* 19(12):3387–3400.

Losing, V.; Hammer, B.; and Wersing, H. 2016. Choosing the best algorithm for an incremental on-line learning task. In *24th European Symposium on Artificial Neural Networks*, 369–374.

Read, J.; Bifet, A.; Pfahringer, B.; and Holmes, G. 2012. Batch-incremental versus instance-incremental learning in dynamic and evolving data. In *International Symposium on Intelligent Data Analysis*, 313–323.

Vinagre, J.; Jorge, A. M.; and Gama, J. 2014. Fast incremental matrix factorization for recommendation with positive-only feedback. In *22nd Conference on User Modeling, Adaptation, and Personalization*, 459–470.

Zang, W.; Zhang, P.; Zhou, C.; and Guo, L. 2014. Comparative study between incremental and ensemble learning on data streams: Case study. *Journal Of Big Data* 1(5):1–16.

References

Alzahrani, A., and Sadaoui, S. 2019. Instance-incremental classification of imbalanced bidding fraud data. In *11th International Conference on Agents and Artificial Intelligence*, 92–102.

Anowar, F., and Sadaoui, S. 2020. Detection of auction fraud in commercial sites. *Journal of Theoretical and Applied Electronic Commerce Research, Universidad de Talca* 15(1):81–98.

Anowar, F.; Sadaoui, S.; and Mouhoub, M. 2018. Auction fraud classification based on clustering and sampling techniques. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 366–371.

Bouchachia, A.; Gabrys, B.; and Sahel, Z. 2007. Overview of some incremental learning algorithms. In *2007 IEEE International Fuzzy Systems Conference*, 1–6.

Diaz-Aviles, E.; Drumond, L.; Schmidt-Thieme, L.; and Nejdl, W. 2012. Real-time top-n recommendation in social streams. In *6th ACM conference on Recommender systems*, 59–66.

Ford, B. J.; Xu, H.; and Valova, I. 2012. A real-time self-adaptive classifier for identifying suspicious bidders in online auctions. *The Computer Journal* 56(5):646–663.

Gu, B.; Quan, X.; Gu, Y.; Sheng, V. S.; and Zheng, G. 2018. Chunk incremental learning for cost-sensitive hinge loss support vector machine. *Pattern Recognition* 83:196–208.

Joshi, P., and Kulkarni, P. 2012. Incremental learning: Areas and methods-a survey. *International Journal of Data Mining & Knowledge Management Process* 2(5):43–51.

Krawczyk, B., and Woźniak, M. 2015. One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing* 19(12):3387–3400.

Losing, V.; Hammer, B.; and Wersing, H. 2016. Choosing the best algorithm for an incremental on-line learning task. In *24th European Symposium on Artificial Neural Networks*, 369–374.

Read, J.; Bifet, A.; Pfahringer, B.; and Holmes, G. 2012. Batch-incremental versus instance-incremental learning in dynamic and evolving data. In *International Symposium on Intelligent Data Analysis*, 313–323.

Vinagre, J.; Jorge, A. M.; and Gama, J. 2014. Fast incremental matrix factorization for recommendation with positive-only feedback. In *22nd Conference on User Modeling, Adaptation, and Personalization*, 459–470.

Zang, W.; Zhang, P.; Zhou, C.; and Guo, L. 2014. Comparative study between incremental and ensemble learning on data streams: Case study. *Journal Of Big Data* 1(5):1–16.