Machine Learning Methods for Detecting Fraud in Online Marketplaces.

Raoul Dekou[1, *], Sabljic Savo[2], Simon Kufeld[3], Diana Francesca[2], and Ricardo Kawase[1]

[1]Mobile.de, MarktPlatz 1 Europarc Dreilinden, 14532, Berlin, Germany
[2]Codecentric AG, Hochstraße 11, 42697, Solingen, Germany
[3]Inovex GmgH, Ludwig-Erhard-Allee 6, 76131, Karlsruhe, Germany
[*]Corresponding author: Raoul Dekou, rdekou@team.mobile.de

Abstract

Connecting buyers and sellers in a safe and secure environment is one of the biggest challenges in online marketplaces. Probabilistic models built upon user-item databases address the challenge, but often encounter issues such as lack of stability and robustness. These issues are magnified in fraud scenarios where datasets are highly imbalanced, noisy and malicious users deliberately adapt their behaviors to avoid detection. In this context, we leveraged the power of existing open sources machine learning libraries H2O and Catboost and designed a pipeline to collect, process and predict the likelihood of a private seller's listing data to be fraudulent. We found that the stacked ensemble model provides the best performance (F1=0.73) when compared to other commonly used models in the field. Further, our models are benchmarked on a public Kaggle Dataset, TalkingData AdTracking Fraud Detection Challenge where we compared them to other studies and highlighted their generalizability and effectiveness at handling online fraud.

In: RWTH Aachen, CEUR-WS: Proceedings of The 2021 International Workshop on Privacy, Security, and Trust in Computational Intelligence, Gold coast, Queensland, Australia, 01-11-2021, published at http://https://xuyunzhang.github.io/pstci2021/

1 Introduction

As reported in [12], retail e-commerce sales worldwide accounted for 1.86 trillion USD in 2016 and are expected to rise to 4.48 trillion USD in 2021. In the meantime, a recent report on fraud attacks trends in the first quarter of 2021[1] confirmed the shift of attacks towards retail websites and estimated that 25% of this traffic is malicious. Such increase in activity has brought enough pressure to marketplaces which need to ensure reliability and security of their services while inspiring trust towards buyers.

Unfortunately, the success of online marketplaces attracts unwanted attention from malicious users who try to abuse the platforms for personal monetary gain. *mobile.de* does not control transactions between buyer and sellers. It is a "matchmaking" platform that bridges the gap between the two sets of entities. Once the user with malicious intent creates an account, he/she also creates an attractive vehicle listing (the goal is to get as many leads as possible). To achieve this, fraudsters take a series of *lead-boosting* steps. They upload listings of high-demand vehicles into the platform and set very low yet reasonable prices for the vehicles. Since every aspect of the listing looks legitimate (the website, the seller and the vehicle), buyers lower their guard and contact the fraudster. Through a series of interactions, the fraudster is able to convince the buyer (now a victim) to send a pre-payment money transfer, usually as a "reservation" fee. Once this happens, and the damage is done, the victims realize their mistake, they contact *mobile.de*'s Customer Service and report the case. There are very few cases

[1]https://securityboulevard.com/2021/07/top-industry-specific-fraud-attack-trends-from-q1-2021/ (accessed on July 2021).

that reach this point, however, the total monthly loss can soar to thousands of Euros.

Satisfied customers (buyers and sellers) are the foundation for a valuable and successful marketplace. Thus, providing a secure enviroment and a safe experience to our customers is a top priority at *mobile.de*, and the motivation of this work which aims at preventing and detecting fraudulent activity. To achieve our goals, we tackled the fraud *detection* problem by leveraging user generated data and building machine learning models which are able to identify fraudulent activities. It is also essential to design robust models, of high precision which can also generalise well. This paper describes our approach to mitigate the case of fraudulent activity by fraudsters posing as private sellers. Our contribution is twofold. First, we describe a production pipeline to collect, process and score sellers' listings using open source machine learning libraries Catboost[2] and H2O[3]. We briefly highlight how to efficiently use these libraries to pre-select relevant candidate models and tune their hyper-parameters. Second, we demonstrate that our approach could potentially inspire other used cases by verifying our detection methods on a sample of a large dataset publicly available at Kaggle.com[4].

The remainder of this paper is structured as follows. In Section 2, we discuss existing work in the field. In Section 3, we provide deeper understanding of the problem and formalize it. In Sections 4 and 5, we describe our methodology to tackle the problem. Section 6 contains our results, followed by the conclusion and prospects

2 Related Work

Techniques used to detect fraud can be divided into two groups: expertise based and data driven. In the first technique, experts use their knowledge to build a set of rules that are tested and refined to filter out fraudulent activities. However, contrary to machine learning solutions traditional expert techniques sometimes lack the ability to model non trivial online connections [24]. The second set of techniques, data driven, i.e. Machine learning solutions, overcome this issue but yield different challenges. While the increase of activity in marketplaces generates massive datasets which require model scalability, the low occurrence of fraudulent events produces imbalanced datasets. Maintaining both a high precision and recall is often a challenge and many models provide significant misclassification errors [2] which result in genuine customers being flagged as fraudulent. Finally, there is also the need for dynamic solutions given that fraudsters adapt their behaviors to a point where they are able to bypass the detection from machine learning models.

Literature suggests various examples of application of machine learning methods which aim at detecting fraud. Najem and Kadeem [16] recent survey on fraud detection techniques in e-commerce, provides a broad view on the performance of the several models on various datasets. It highlights that Random Forest (RF) is the most used and usually the most accurate of all methods. Though Naive Bayes algorithms are easy to implement, they are limited compared to decision trees when it comes to modelling non linear problems. Such information were taken into consideration when selecting candidate models for our pipeline which consists essentially of decision trees ensembles (RF, Xgboost and Catboost). For instance, Kanei et al. [10] trained a Random Forest model for detecting fraudulent ad requests. In their study, they demonstrated that the model robustness challenge could be addressed by means of features which could not be controlled by fraudsters such as the network statistics from clients and publishers. This set-up allowed them to improve their recall rate by 10%. Renjith [20] described a pipeline using Support Vector Machine (SVM) to detect fraudulent sellers in an online marketplace. The authors specifically pointed out that a cold start problem may arise for new users when using predictive models with seller or transaction information as features. In our approach, the cold start effect was mitigated by removing these types of features. Gupta et al. [8] benchmarked ensemble models for predicting the likelihood of a click on mobile phone advertisement to be fraudulent on a publicly available Kaggle dataset. They tested two configurations: traditional and Big Data. In the traditional configuration, they combined different sampling techniques (SMOTE, stratified sampling, etc) to reduce the data size and handle the imbalanced training set. This dataset which has been widely used in previous studies [8, 14, 22], is employed in our study and results from Gupta et al. [8] are used as our baseline. In our work, we applied the same preprocessing techniques and compared our results to their best model, Two Class Decision Forest[5] with an F1 score of 0.944. Using a sample of the same dataset, Minastireanu and Mesnita [14], trained a Lightgbm model to detect fraudulent clicks and reported an accuracy of 98%. The authors specifically described an example of how feature engineering on original features set (click_time, device,

[2]https://www.catboost.ai/ (accessed on July 2021).

[3]https://www.h2o.ai/ (accessed on 16 July 2021).

[4]https://www.kaggle.com/c/talkingdata-adtracking-fraud-detection/data (downloaded on 16 July 2021).

[5]https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/two-class-decision-forest (accessed on July 2021)

channel, etc) and K fold cross validation are combined to enable high performance. Besides, by testing their model on a large data sample (18 millions users clicks), they proved the robustness of the boosting machine for the case study. In the same context, Mohammed et al. [15] investigated the scalability of Random Forest, Balanced Bagging Ensemble and Gaussian Naive Bayes on massive and highly imbalanced credit card fraud datasets. They found that random undersampling is effective at handling imbalanced datasets, and combined with RF, it is suitable for real time applications on large datasets. In their study, the Random Forest model provided the highest recall of 91%. Rajora et al. [19] benchmarked the performance of various machine learning algorithms on a credit card transaction dataset with 31 attributes. They used random undersampling technique to address the data imbalance and Principal Component Analysis (PCA) [1] as dimensionality reduction technique. On top of PCA features, a time feature corresponding to the time delay from the first transaction is part of the training set. Furthermore, the authors illustrated how the inclusion of this feature can impact the performance. RF provided a better performance without the time feature while Gradient Boosting Regression Tree performance was constant. Meng et al. [13] also used a real world credit card transactions dataset and combined Xgboost and sampling techniques to achieve great performance. SMOTE technique allowed an increase of the recall from 0.8062 to 0.9 and the AUC from 0.9795 to 0.9853. Mohammed et al. [15] reported that Neural Networks tend to overfit on fraud datasets and struggle to handle imbalanced datasets. Nevertheless, as illustrated by Adewumi and Akinyelu [2] in their survey, such techniques are also commonly used for credit card fraud detection. Najem and Kadeem [16] pointed out that hybrid methods which combine several methods to build a robust learner provide better performance than individual learners. For example, Wang et al. [23] built an hybrid mixed model consisting of Xgboost and Logistic regression (LR) and benchmarked it against common baseline models such as Xgboost, RF, SVM, Naive Bayes and Logistic Regression on the German Credit dataset published by UCI[6]. In the hybrid model, an effective feature combination was obtained by using Xgboost leaf nodes as features for the LR model. This set up, provided an AUC of 0.8321 which is far beyond the value of 0.7321 obtained with LR, the best individual model. Other studies such as [18] and [21] use meta learning techniques to enhance the performance on credit card fraud dataset. However, combining the output of different classifiers to build a model reduces the classification speed [2] which might be an issue on big datasets.

[6]https://archive.ics.uci.edu/ml/index.php (accessed on July 2021).

3 Problem statement

mobile.de supports two different types of sellers, namely dealers and private sellers. Dealers are those registered dealerships in Germany and neighbouring countries who are paying customers of *mobile.de*. These are professional sellers who make a living out of buying and selling vehicles. Private sellers are the regular common citizens who own a vehicle and use a classified market to sell it (not registered as a business). Internally, at *mobile.de* a private seller is labelled and named as FSBO (For Sale By Owner), and for the rest of this paper, we will address a private seller with the same terminology. Although there are several malicious activities which can be classified as fraud such as: account take over, falsification of documents, etc., our objective in this study is focused on a single type of users (FSBOs) that create fraudulent (fake) listings. Our pipeline overview is depicted in Figure 1. When a listing is created (or updated) our machine learning models generate a fraud probability prediction and, in case the result is above a certain threshold, the listing is manually evaluated by a Customer Service (CS) agent, who reviews the content of the listing and assigns a rating (ground truth). In addition to listings flagged by our ML models, Customer Service agents extend their reviewing process to listings which might have received users' complaints. Eventually, one way or another, every fraudulent listing is flagged in our dataset, the vast majority happening before damage is done, and in very few cases, reports come from scam victims. The main classification task is binary in the sense that the target variable to predict has two possible outcomes OK or FRAUD. The goal is to detect when a vehicle listing is (or becomes) fraudulent. It can happen at the insertion time (version 1 of the listing) or at any time later due to a modification in the data.

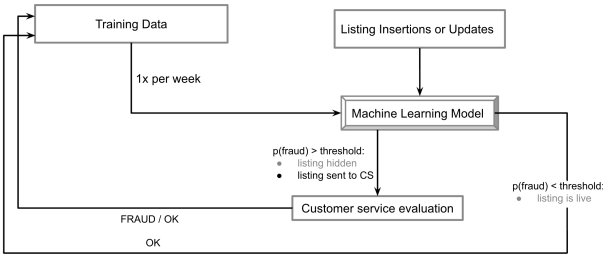

Figure 1: *mobile.de* in house data collection and pipeline overview.

4 Datasets

In this study, we used two different datasets to train and test our machine learning models, *mobile.de* in-house dataset and a tailored sample of TalkingData AdTracking Fraud Detection Challenge dataset obtained from the machine learning competition platform Kaggle.

At *mobile.de* FRAUD cases are less frequent (positive cases) than the OK cases leading to a highly imbalance dataset. The in-house dataset consists of 27 categorical variables and 10 continuous ones. To maintain the confidentiality of our data points, and to eliminate the risk of giving any clues that could lead to learnings on how to bypass our fraud detection models, we refrain from disclosing the exact names of the attributes and features.

The public dataset is taken from the China's largest independent big data service platform which covers 70% of active mobile devices in the country, handles 3 billion clicks per day out of which 90% are potentially fraudulent. Contrary to *mobile.de* case, here click fraud is the most frequent class (negative class) and occurs when a person or an automated bot acting as legitimate user clicks on an app ad without downloading the app afterwards. The raw dataset contains 200 millions clicks over a 4 day period. It includes 7 data fields (IP, app, device, OS, channel, click_time, attributed_time) and a binary target to predict (is_attributed). The target variable is imbalanced with 99.8% of negative cases.

Tables 1 and 2 summarize the preprocessing steps applied on *mobile.de* and TalkingData datasets respectively. For our in-house dataset, the testing set corresponds to samples recorded 7 days prior to the day the model was trained. The training set corresponds to 28 days of data prior to the start date of the testing set. The timely split was done to prevent the model from learning from future observations. In order to reduce the imbalance and increase the performance, we applied a random undersampling and kept 10 % of the majority class in the training set. This resulted in around 200,000 training samples and 240,000 testing ones. We kept raw missing entries within the sets, H2O and Catboost models handled them as separate categories[7,8].

For the Kaggle dataset, we borrowed the preprocessing steps from [8] and we engineered two additional features: click hour of the day and day of the week. First, we reduced the data size by randomly sampling 15% of unique IP addresses and retaining a stratified sample of 8% of the remaining set. To handle the imbalance, we applied Synthetic Minority Over Sampling Technique (SMOTE) [5] with 5 neighbours and oversampled the positive class up to 11%. We then applied a stratified split, keeping 70% of the set for training. The final set has 1,706,481 training samples and 731,349 testing ones without any missing values.

Table 1: In-house dataset preprocessing steps.

non overlapping time based split	- test (latest week) - train (28 days)
undersampling	random undersampling of the training set, 10% negative cases kept
missing values	kept and processed by machine learning models
feature engineering	yes (confidential)

Table 2: TalkingData dataset preprocessing steps.

subsampling	15% random sample of unique IPs then 8% stratified sample from the remaining set
oversampling	SMOTE with k=5 neighbours, positive class up to 11%
missing values	absent
stratified split	- test (30%) - train (70 %)
feature engineering	- click hour&day of the week - attributed_time is removed

5 Training Machine Learning Models

In this section, we briefly summarize the theoretical concepts behind the models used in our study, provide an overview of the machine learning libraries in which the models were implemented and finally describe the hyper-parameter tuning steps and our performance metrics.

As stated in [4], Random Forest is an ensemble machine learning algorithm consisting of a collection of decision trees each built from random samples. In each tree, thresholds are applied to the input features to maximize information gain while minimizing an impurity function (for e.g. Cross Entropy, Mean Squared Error, etc). The final score is given by the average scores of all trees. Besides, RF provides maximum depth and minimum sample split parameters to prevent decision trees from overfitting on the training set.

Xgboost [6] is another ensemble method which belongs to the large family of boosting algorithms. In

[7] https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/gbm-faq/missing_values.html (accessed on 16 July 2021).

[8] https://catboost.ai/docs/concepts/algorithm-missing-values-processing.html (accessed on 16 July 2021).

general, boosting models combine shallow decision trees (also called weak learners), each built sequentially considering the errors on previous trees to reduce bias and variance at the same time. Xgboost particularly is an advanced implementation of gradient boosting which includes additional features such as parallel processing and regularization techniques for handling overfitting.

Introduced in [17], Catboost is a boosting model designed to handle and process categorical data efficiently. By default, Catboost implementation uses one hot encoding technique on categorical variables except for the ones with high cardinality. In such a case, ordered targeted statistics [17] are used to maximize information gain. Contrary to other machine learning techniques which require preprocessing steps to convert categorical data into numbers, Catboost requires only the indices of the categorical features [7].

Meta learning technique aims at combining the output of several based learners to improve the prediction accuracy and utilize the strength of one learner to complement the weaknesses of others [18]. In this study, we used H2O AutoML [11] to build a stacked ensemble. AutoML brings out a simple wrapper function optimized for training and combining a large number of models in a short amount of time. This module evaluates single machine learning models (GBM[9], Xgboost, RF, Extremely Randomized Trees[10], Artificial Neural networks[11] and Generalised Linear Models[12]) and their stacked ensembles on validation sets using relevant metrics (for e.g. AUC, logloss, etc). The best performing model is then retained for deployment.

H2O is an open source distributed library software for machine learning and deep learning applications. Its attributes: frame and clusters allow to easily process tabular data of various types in a distributed fashion. H2O platform supports various interface including R, Python and Java making it easier to complete analytic workflows [3]. In our case, we used H2O Python interface to train and optimize Distributed Random Forest (DRF), Xgboost and AutoML models. The models trained are saved as MOJO (Model Object Optimized) formats which are later embedded in JAVA environment for real time predictions.

The Catboost library is another high performance open source framework for gradient boosting on decision trees. Similar to H2O, Catboost library supports Python, R and JAVA interfaces. For this study, we combined Catboost's Python and JAVA interfaces for model training and deployment.

[9]https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/gbm.html (accessed on 16 July 2021).

[10]https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html#extremely-randomized-trees (accessed on 16 July 2021).

[11]https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html (accessed on 16 July 2021).

[12]https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/glm.html (accessed on 16 July 2021).

Table 3: H2O models hyperparameters (in-house dataset).

parameter	RF	Xgb	AutoML
maximum number of models	-	-	20
number of trees	100	1000	-
maximum depth	50	35	-
number of columns for a DT split	9	-	-
columns sample rate	-	0.8	-
sample rate	-	0.8	-
learning rate	-	0.009	-
early stopping metric	logloss	logloss	logloss
early stopping rounds	-	25	3

Table 4: Catboost hyperparameters and Hyperopt "quantized" continuous distributions minimun and maximum values used for optimisation.

Parameter	Hyperopt function	min	max
l2_leaf_reg	qloguniform	0	2
learning_rate	qloguniform	0.001	0.5
subsample	quniform	0.5	1
colsample_bylevel	quniform	0.5	1

5.1 Hyperparameters tuning

The parameter optimization described in this section is limited to our in-house dataset. In fact, because of TalkingData large sample size (1,706,481 entries) carrying out an extensive hyper parameters tuning is daunting. Therefore, for this dataset, we applied a full parameter optimization only for the Catboost model and kept similar parameters for their H2O counterparts.

For H2O, 3, 5 and 10 folds Cross Validation (CV) have provided the best performance for RF, AutoML and Xgboost respectively. These models hyperparameters are depicted in Table 3. However, on the public dataset, we set the maximum number of models to 10 and the number of folds to 3 to circumvent memory limitations for AutoML.

For Catboost, Python library Hyperopt[13] allowed hyperparameters optimization. Hyperpot provides custom functions for hyperparameter search. Each parameter value is retrieved from a list of candidates taken from a specific "quantized" continuous distribu-

[13]https://github.com/hyperopt/hyperopt (accessed on 16 July 2021).

Table 5: Area Under the Receiver Operating Characteristic Curve of the best single learner of each model family derived from H2O AutoML *leaderboard()* method (in-house dataset).

Metric	AUC
Stacked Ensemble (all models)	0.9850
Stacked Ensemble (best of each family)	0.9848
Gradient Boosting Machine	0.9826
Extreme Gradient Booosting	0.9821
Random Forest	0.9790
Extremely Randomized Trees	0.9719
Generalized Linear Model	0.9690
Articifcial Neural Network	0.9200

tion such as qloguniform and quniform (see Table 4). Besides, models are trained for 500 iterations, using 3 folds CV, the logarithm loss function and Area Under the Receiver Operating Characteristic Curve (AUC) evaluation metric.

5.2 Performance metrics

In an imbalanced classification task, the positive class denotes the less frequent value of the target and the negative class is its complement. When scoring a model, an optimal solution can be derived from the confusion matrix [9]. True positive (TP) and True negative values (TN) occur when the output of the model matches with the ground truth label on positive and negative classes respectively. Conversely, False Positive (FP) and False Negative (FN) occur when the model provides predictions which mismatch with the true labels. To convert model probabilities into classes, we chose a threshold in order to maximize the F1 score on the testing set accordingly. F1 score is the harmonic mean between the precision and recall and evaluates the accuracy of the model at predicting the positive class. Another popular evaluation metric is the Area Under the Receiver Operating Characteristic Curve. Contrary, to the previous metrics, it is used to assess the ability of a classifier to distinguish between classes independently of any selected threshold.

6 Results

In order to retain candidate models for our evaluation, we first benchmarked a large pool of machine learning models. For this purpose, H2O AutoML objects provide *leaderboard()* method which allows to rank the models trained to build the stacked ensemble on chosen dataset and metric. These models are optimised with AutoML predefined random grid parameter searches which are different from our production hyper-parameters tuning described in the previous section. Table 5 summarizes the AUC obtained on our in-house test dataset but limited to the best algorithms of each family (GBM, Xgboost, RF, Extremely Randomized Trees, Artificial Neural networks and Generalised Linear Models). Tree based models outperform Artificial Neural Networks and Generalised Linear Models. They suit well to complex non linear problems [16]. Especially, GBM and Xgboost yield the best AUC of 0.982 followed by Random Forest of 0.9790 AUC. Besides, Najem and Kadeem [16] survey on fraud detection techniques in e-commerce demonstrated that RF has the highest frequency usage and is the best performing one across various use cases. Based on these observations, we initially retained AutoML, Xgboot and RF for our benchmark. Catboost model, which is not part of H2O was benchmarked separately and added later for the comparison.

Table 6: Machine learning models performance summary (in-house dataset).

Model	F1	Precision	Recall	AUC
AutoML	0.7293	0.7206	0.7833	0.9850
Xgb	0.7134	0.7104	0.7165	0.9794
Catboost	0.7127	0.7375	0.6895	0.9809
RF	0.6810	0.7274	0.6401	0.9786

Tables 6 and 7 illustrate performance metrics obtained from the different models on *mobile.de* and TalkingData datasets respectively. On the first one, AutoML best model (stacked ensemble) yields an F1 score of 0.73 which is higher than the one of 0.71 obtained with Xgboost and Catboost and of 0.68 with Random Forest. It has been reported in [11] that stacked ensemble models usually produce better performance than individual models (Xgboost, Random Forest, etc) used in an AutoML run in accordance with our findings. On Talking Dataset, Catboost model yields the best performance with an F1 score of 0.988. Catboost model is designed to process heterogeneous data with categorical variables efficiently [17]. The features cardinality is highlighted in Table 8. One hot encoding on one side and ordered targeted statistic applied on variables of high cardinality have a significant impact on the model performance. Catboost also provides *get_feature_importance()* method which gives the contribution of each feature to the ensemble model. The output of this method is summarized in Figure 2, the app id for marketing and the IP address of click are the most important features.

In order to assess the generalizability of our modelling approach at detecting fraud, we compared our models with the work of Gupta et al. [8]. Their best model, Two Class Decision Forest classifier provides a precision of 0.992 and a recall of 0.902 corresponding to an F1 score of 0.9442. All the models used in our experiment outperform their results in terms of

Table 7: Machine learning models performance summary (TalkingData dataset).

Model	F1	Precision	Recall	AUC
Catboost	0.9888	0.9902	0.9873	0.9994
AutoML	0.9800	0.9848	0.9752	0.9987
Xgb	0.9787	0.9804	0.9771	0.9982
RF	0.9780	0.9801	0.9758	0.9985

Table 8: Count of distinct values per columns in Talking data training set.

feature	count of unique values
IP	123099
device	1450
OS	558
channel	496
app	383
hour	24
dayofweek	4

F1 (see Table 7). Especially, our best model Catboost demonstrates a comparable precision and a better recall. Relying on F1 score alone to compare our models would be problematic since in the TalkingData's context the positive class correponds to the non fraudulent clicks. In the TalkingData adTracking Fraud Detection Challenge, Kaggle competitors' machine learning models were evaluated based on AUC. Using such a metric, our Catboost model yields an AUC of 0.9994 compared to 0.997 from Gupta et al. [8].

7 Conclusions

We presented a case study which described the application of ensemble methods to detect fraud in a large scale online marketplace (*mobile.de*). The business value of such an investigation is twofold. First, to enable a trustworthy customers' experience and enhance customers' satisfaction. Second, to reduce Customer Service operational cost in order to resolve fraudulent cases.

To achieve our goals, we designed a Machine Learning pipeline based on sellers' listings data and optimized a way to address common challenges in fighting fraud (fraudsters adaptability, dataset imbalance, high false positive rate, etc). The main contribution of this study is that it proposes a pipeline using open source data science libraries to collect, process and score sellers listings to efficiently detect fraud. Our best model AutoML has provided an F1 score of 0.73 outperforming Catboost, Xgboost and Random Forest. These models were later tested on a TalkingData public dataset from Kaggle competition platform and yielded great robustness at detecting fraud and outperformed previously proposed models. The best model on this set, Catboost provides an F1 score of 0.9888 which is significantly higher than the value of 0.9442 reported in [8].

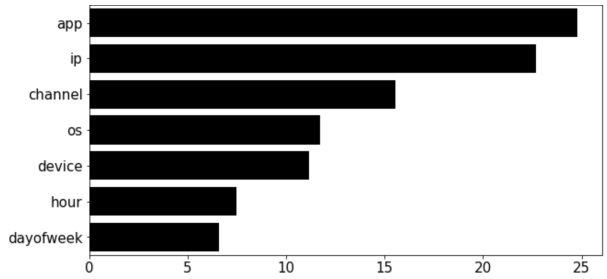

Figure 2: Catboost model feature importance (TalkingData dataset).

With regard to the prospects of the study, we will first explore dimensionality reduction techniques [19] and encoding methods in order to improve the performance of the classifiers. Second, we will leverage the power of Big Data tools (for e.g Spark) to train and optimize the models on larger samples of data. In addition to that, we aim at investigating different meta learning techniques combining Catboost and H2O models to build robust classifiers and further prevent fraud in our website.

Furthermore, in our future work we will tackle the problem of detecting fraud "as soon as possible". It is crucial that fraudulent listings are detected before it reaches the audience. To this end we plan to include further features such as buyers' and sellers' user activity. Finally, we would like to highlight that the work present in this paper is currently in production, protecting buyers and sellers at *mobile.de*, and due to that we refrain from disclosing more technical details that could help malicious users to bypass our detection system.

8 Acknowledgements

We would like to thank the Customer Service team at *mobile.de* for their countless hours of manual work in detecting fraud, and for providing us the ground truth to start our work. We would also like to thank members of TnS and Data teams at *mobile.de* who have directly and indirectly been involved in this work, with special thanks to Moritz Aschoff and Matthias Radtke.

References

[1] Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

[2] Adewumi, A. O. and Akinyelu, A. A. (2017). A survey of machine-learning and nature-inspired based credit

card fraud detection techniques. *International Journal of System Assurance Engineering and Management*, 8(2):937–953.

[3] Aiello, S., Click, C., Roark, H., Rehak, L., and Stetsenko, P. (2016). Machine learning with python and h2o. *Edited by Lanford, J., Published by H*, 20:2016.

[4] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[5] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

[6] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

[7] Ghori, K. M., Abbasi, R. A., Awais, M., Imran, M., Ullah, A., and Szathmary, L. (2019). Performance analysis of different types of machine learning classifiers for non-technical loss detection. *IEEE Access*, 8:16033–16048.

[8] Gupta, N., Le, H., Boldina, M., and Woo, J. (2019). Predicting fraud of ad click using traditional and spark ml. In *KSII The 14th Asia Pacific International Conference on Information Science and Technology (APICIST)*, pages 24–28.

[9] Hossin, M. and Sulaiman, M. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1.

[10] Kanei, F., Chiba, D., Hato, K., Yoshioka, K., Matsumoto, T., and Akiyama, M. (2020). Detecting and understanding online advertising fraud in the wild. *IEICE Transactions on Information and Systems*, 103(7):1512–1523.

[11] LeDell, E. and Poirier, S. (2020). H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*.

[12] Lee, S.-J., Ahn, C., Song, K. M., and Ahn, H. (2018). Trust and distrust in e-commerce. *Sustainability*, 10(4):1015.

[13] Meng, C., Zhou, L., and Liu, B. (2020). A case study in credit fraud detection with smote and xgboost. In *Journal of Physics: Conference Series*, volume 1601, page 052016. IOP Publishing.

[14] Minastireanu, E.-A. and Mesnita, G. (2019). Light gbm machine learning algorithm to online click fraud detection. *J. Inform. Assur. Cybersecur*, 2019.

[15] Mohammed, R. A., Wong, K.-W., Shiratuddin, M. F., and Wang, X. (2018). Scalable machine learning techniques for highly imbalanced credit card fraud detection: a comparative study. In *Pacific Rim International Conference on Artificial Intelligence*, pages 237–246. Springer.

[16] Najem, S. M. and Kadeem, S. M. (2021). A survey on fraud detection techniques in e-commerce.

[17] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pages 6638–6648.

[18] Pun, J. and Lawryshyn, Y. (2012). Improving credit card fraud detection using a meta-classification strategy. *International Journal of Computer Applications*, 56(10).

[19] Rajora, S., Li, D.-L., Jha, C., Bharill, N., Patel, O. P., Joshi, S., Puthal, D., and Prasad, M. (2018). A comparative study of machine learning techniques for credit card fraud detection based on time variance. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1958–1963. IEEE.

[20] Renjith, S. (2018). Detection of fraudulent sellers in online marketplaces using support vector machine approach. *arXiv preprint arXiv:1805.00464*.

[21] Suganya, S. and Kamalra, M. (2016). Meta classification technique for improving credit card fraud detection. *International Journal of Scientific and Technical Advancements*, 2(1):101–105.

[22] Thejas, G., Dheeshjith, S., Iyengar, S., Sunitha, N., and Badrinath, P. (2021). A hybrid and effective learning approach for click fraud detection. *Machine Learning with Applications*, 3:100016.

[23] Wang, M., Yu, J., and Ji, Z. (2018). Credit fraud risk detection based on xgboost-lr hybrid model. In *Proc. Int. Conf. Electron. Bus.*, volume 2, pages 336–343.

[24] Zhang, Z., Zhou, X., Zhang, X., Wang, L., and Wang, P. (2018). A model based on convolutional neural network for online transaction fraud detection. *Security and Communication Networks*, 2018.