Human Or Robot?

Facebook Recruiting IV: Human or Robot?

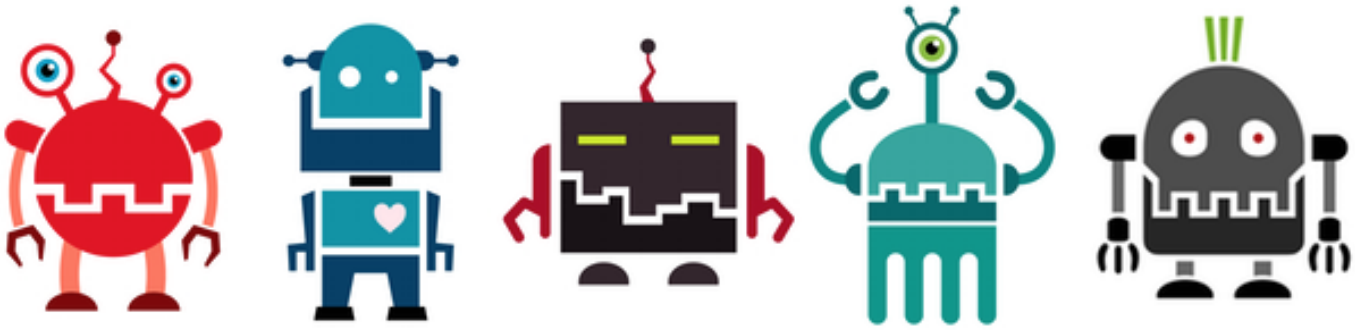
Data Science Workshop, 2020

Orel Alon – Orelalon1993@gmail.com

Yizhar Alamgor - yizharal2@gmail.com

Yonatan Rosenberg - yonatanr2@mail.tau.ac.il

Tom Amsterdam – Amsterdam.tom@gmail.com



Abstract

Our project aims to leverage data science and machine learning to solve a real-life problem of identifying robot bidders from human bidders in online auctions. Our ML pipeline involves pre-processing, feature extraction, feature selection, model selection and classification tests. Our experiments reveal different performance achieved by different models, our best performance achieves 0.936 AUC and is between the 21th and 20th position of the competition.

1. Introduction

1.1. Overview

The project is based on a past Facebook recruiting competition from Kaggle. [1] The project aims at predicting for a specific bid whether the bidder is a robot or a human based on their behavior. The motivation is to facilitate a fair competition on the website and therefore to improve the users' satisfaction. The project is based on real-life dataset of an online auction website working with Facebook.

1.2. Problem formulation & Description

Our project is a supervised binary classification problem – determining a bidder between a human or a robot based on limited past bidding records. More formally the problem can be formulated: Given sets of $X_{Bidders}$ of N bidders and X'_{bids} of M bids and set Y of size N with 2 labels (human, robot), a machine classifier should predict for each sample of $X_{Bidders}$ the appropriate corresponding label from Y.
We use 2 evaluation criteria:

I. *Area under the ROC curve (AUC-ROC*) which is used in the competition in order to be able to compare the results with the leaderboard.
II. *Average Precision* is used on the train data in order to assess the quality of the prediction due to data skewedness.

The main point of focus of the project is handling complex time-series based dataset.

1.3. Related Work

There is a substantial amount of work in the field on online fraud prevention and on Bots Identification which relates to our specific project. For consumers, the chances of landing a winning bid in online auctions has become increasingly difficult with the abundance of "bidding robots" (such as "BidRobot" and "Auction

Sniper" [2]). Chau & Faloutsos research tackles the problem of fraud detection in electronic auctions that use a Random Forest model leveraging price-based features to differentiate fraudsters from normal bidders. Packer & Huang tackled the Facebook competition and achieved 0.8747 AUROC using AdaBoost [3]. Gu & Shi [4] published a research based on the Facebook Competition in which they tested different models and achieved 0.94 AUROC in the private leaderboard. As for writing this article the notebooks in Kaggle are undisclosed [1]

2. The Dataset

2.1. Dataset Description [1]

The datasets include basic account information and bidding records for all the bidders. There are no pre-built features at all, one most construct their own features with the bidding records provided. The dataset is comprised of 2 different datasets:

i. *Bidders Dataset*: Basic information regarding the bidder accounts and the appropriate labels. Split into 2 different files – Test and Train (labels of test bidders are not provided)

ii. *Bidding Dataset*: Information regarding each bid in each auction.

Bidders Dataset			
#	Field	Detail	Comments
1	Bidder_Id	Unique bidder identifier	
2	Payment_account	Account Identifier	Obfuscated
3	Address	Mailing Address Identifier	Obfuscated
4	Outcome	Bid Tag – 1 for Bot	

Bidding Dataset			
#	Field	Detail	Comments
1	bid id	Unique Bid Identifier	
2	bidder id	Unique bidder identifier	
3	Auction	Unique Auction Identifier	
4	Merchandise	Auction Category	
5	Device	Bid Phone Model	Transformed
6	Time	Bid Timestamp	Transformed
7	Country	Bid's Country	
8	IP	Bid IP Address	Obfuscated
9	URL	Referral URL for bid	

2.2. Dataset Analysis

We began by examining the data to understand important information regarding the dataset distribution and information regarding key behaviors that differentiate human and bot bidders.

i. General Dataset Information – There are 7.6M bids and 6614 unique bidders. The training set has 2087 bidders of which 1984 are labeled as human and 103 are labeled as robots. The test set had 4700 bidders, of which 4630 participated in bidding.

ii. Statistical Analysis-

Statistic	Human	Robot
# of Bids	1414	4004
# of Bids Per Auction	6	23
# of Auctions Won	6	18
# Devices	164	74
# IP	581	2388
# URL	335	545

The preliminary data analysis helped us to develop crucial insights regarding the data-science process and the differences between Robots and Human

i. Dataset Skewedness – There are ~ 95% humans what makes very few bots to learn (~100 in train)
ii. Different Bidding Strategies – Bots are more active, bid more and win more auctions.
iii. Different Bidding Behaviors- Bots tend to change the parameters they use more than humans – use more IP addresses, devices and URLs.

After performing Time-Series analysis (point of focus), we understood additional insights regarding the difference between humans and robots (more information in paragraph 4.4)

i. Different Time Bidding Patterns - Bots time patterns are different than humans – less variance and dependency on time of the day.
ii. Faster Bids - Bots bid faster than humans (consecutive bids time differences).

3. The Solution

3.1. General Approach

Our approach to the problem displayed is based on the following stages:

i. Data Visualization & Analysis – understanding distinguishing patterns between the robots and humans and develop hypotheses based on the differences.
ii. Data Pre-Processing – Cleaning that data and handling both NaN values and bidders without activity. Encoding the categorical variables (Merchandise, Payment Accounts, Address and etc.) and correlating the bids and bidders.
iii. Time-Series Analysis and Processing – Analyzing the time-series data, creating time-based insights for the auction's timelines and user activity and processing timestamps.
iv. Feature Extraction – Perform Feature extraction based on our hypotheses
v. Model Selection & Evaluation
vi. Compare, Improve and Iterate – Analyze specific samples missed or labeled incorrectly, create correlating features and properly tune the model parameters.

3.2. Data Preprocess

3.2.1. Data Cleaning

Initial data pre-process analysis raised several problems with noisy, inconsistent and missing values. There were 2 main problems to address regarding missing values:

i. Bidders without bids – 29 bidders didn't have any bid in the bids dataset. All 29 bidders were labelled as human and therefore were dropped from the dataset without affecting future classification.

ii. Missing values – there were 2701 with inconsistent and missing value for the country the bid was performed from. These bids are 0.09% of the total amount of bids and below to different bidders and therefore these entries were dropped from the dataset.

3.2.2. Outliers

Further analysis of the dataset raised several concerns regarding the consistency of the data and with outliers in the data

i. Number Of Bids Outliers – 5 robot had 1 bid while the average amount of bids per robot was 4004. Even though the data is imbalanced and skewed towards human bidders, we decided to drop the 5 robot bidders.

ii. Value Outliers - We used 3 Sigma to Median (on counts) to remove outlier values.

3.2.3. Data Embedding & Representations

The dataset contains several categorical features that needs to be represented differently in order to be used in models:

i. Merchandise – contains information regards category of the auction site campaign leading to the specific auction, there are 9 different categories of merchandise. This data was represented with one-hot encoding.

ii. Payment Account & Address – contains information regarding the payment accounts and address that the bidder had filled and does not depend on the specific bid. The field is of type String and is characterized by high variance between the bids. We didn't perform One-hot-encoding for the field by rather used a 'binning' solution by representing the address by the following Boolean bins – rare, infrequent and frequent.

3.3. Time-Series Analysis – Point of Focus

Our project main point of focus was a complex problem setting such a time-series analysis as presented below. We also believe we've also tackled non-trivial feature extraction both in regards for the time-series analysis regarding the fraud activity-based features .

Our time-series analysis methodology and work can be broken into several stages:

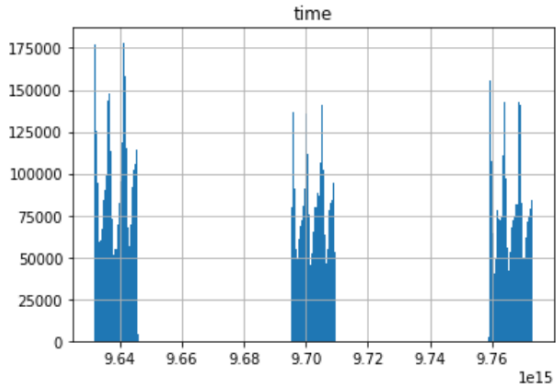
i. Bid Time Analysis– The time-columns in the dataset was obfuscated to protect privacy but in a way that preserves the time-order of the different bids. By examining and visualizing the activity after

discretizing the time information into time-slices a specific pattern occurs – 3 separate chunks each chunk seems like a 3-day period.

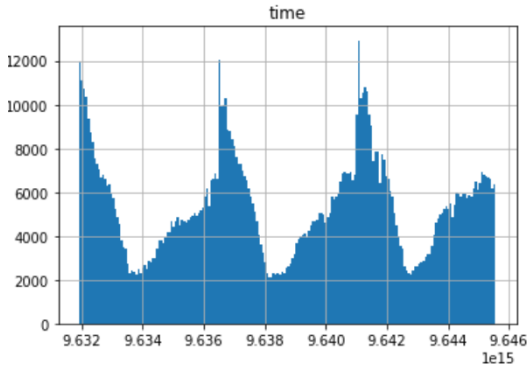
ii. Time Stamp Processing – The obfuscated time given has 10^{15} magnitude, these may cause numerical overflow problems when applying different models. We needed to processes the time information and transform it to a suitable magnitude. We transformed the time units into time intervals and then reassigned time-stamps, assuming peak activity is at 7pm.

iii. Aggregating Time-Based Features- Based on the time-analysis and the time-activity related differentiation of the robots and humans we've developed hypothesis for further feature extraction. Our main hypothesizes were that the time distribution of bids, bidding speeds, parameters lifetime length and bid streaks will be different between robots and humans. We therefore developed Time-Series based feature extraction and described in the paragraph 4.5 below.

3 Different Chunks



Each Chunk of 3 Days



3.4. Feature Extraction

Following the dataset and time-series analysis described in sections 3.2 and 4.4 we extracted and hand-crafted a list of features. Each hand-crafted feature corresponds to one of the following 4 classes of hypothesis regarding difference between humans and robots:

i. User Activity– Aggregative features and statistics of the bidder's activity in general. The underlying hypothesis is that human and robots differ in used parameters and activity.

ii. Auction Activity - Aggregative features of the bidder's activity per auction. The underlying hypothesis is that bots and human differ in their auction bidding strategy.

iii. Suspicious Activity – Features based on hand-crafted Rules and past bots activity analysis. The underlying hypothesis is that bots tend to share and re-use methods of operation.

iv. Time-Series Activity – Features based on the time analysis. The underlying hypothesis is that robots and humans differ in regards to activity over time.

Feature Extraction List		
#	**Hypothesis Class**	**Feature**
1,2,3	*User Activity*	Boolean – Rare & Infrequent IP, Address and Payment Account,
4	*User Activity*	Boolean- is payment account equals address
5-11	*User Activity*	Counts Per User - # Bids, Auction, Merchandise, Device, Country, IP, UR
12-15	*User Activity*	Average Number Of Bids Per Parameter - # Bids/Auction, Bids/Device, Bids/URL, Bids/Country
27-31	*User Activity*	Advanced Country Information Most Common Country & Fraction of Bids in Each Country Per User, Median/Max/Mean Number of Countries Per Auction
16,26	*Auction Activity*	Average Count Per User Per Auction - IP, Country
32	*Auction Activity*	Number of Auction Won
34	*Auction Activity*	Count Of Bids Per Use Per Price Percentile
43-44	*Auction Activity*	Count of Bids in first and last 10% of Won Auction
45	*Auction Activity*	Average Amount Of Bids in Won Auction
46	*Auction Activity*	Fraction of Bids per Price Percentile of Auction
17	*Suspicious Activity*	Fraction of Bids from Rare IP Address
18,20,22,24	*Suspicious Activity*	Parameter Used By a Robot in the past (Boolean Flag) – IP, Device, Country, URL
19,21,23,25	*Suspicious Activity*	Parameter Used By a Robot in the past (% from bids) – IP, Device, Country, URL
33	*Time-Series Activity*	Count Of Bids Per Use Per Time Percentile
35-38	*Time-Series Activity*	Number and Fraction of Bids in the First 10% and last 10% of the time of auction
39-42	*Time-Series Activity*	Average Min/Max and Global Min/Max of Bidder Consecutive Bids Per Auction
43	*Time-Series Activity*	Fraction of bids in each 6 hour time-frame window
44-49	*Time-Series Activity*	Min and Mean change time of parameter per bidder in consecutive bids – IP, Device, Country
50-52	*Time-Series Activity*	Max Bid count in 10/30/60 minutes timeframe
53-54	*Time-Series Activity*	Max and Mean bid streaks of consecutive bids with same parameters

3.5. Methods and Models

We use different kinds of models on the problem to see how the perform differently. First, we trained a baseline classifier which includes SVM. Then we used decision tree models- Random Forest and Gradient Boost Tree and XGBoost. For each model we experiment with tuning hyper-parameters using random search and preform feature selection to select the best features for each model. We then train the models and compare the results.

3.6. Experimental Results

Results Pre-Feature Selection

Model	Training		Cross Validation		Test	
	AUC-ROC	*Average Precision*	*AUC-ROC*	*Average Precision*	*AUC-ROC*	*Average Precision*
SVM Linear	*0.77*	*0.1486*	*0.7274*	*0.2033*	*0.7410*	
Random Forest	*1*	*0.9950*	*0.9240*	*0.4619*	*0.93654*	
Gradient Boost	*1*	*0.9904*	*0.8449*	*0.3911*	*0.91569*	
XGBoost	*1*	*1.0*	*0.8600*	*0.4656*	*0.92008*	

Results Post-Feature Selection

Model	Training		Cross Validation		Test	
	AUC-ROC	Average Precision	AUC-ROC	Average Precision	AUC-ROC	Average Precision
Random Forest			*0.9170*	*0.4742*	*0.9242*	
Gradient Boost			*0.8973*	*0.4091*	*0.9141*	
XGBoost			*0.9038*	*0.4173*	*0.9225*	

3.7. Model Improvement

3.7.1. **Model's Error Analysis**:

After selecting the best models, we split the train dataset to test & train sets (with ratio 1:3) and analyzed the misclassifications 16 out of 25 bots that were on the test set were misclassified while all humans were classified correctly. We used several methods to measure the similarity of the latter to humans/bots instances on the train set:

i. Euclidean Distance From Mean – We measured Euclidean distance of each of misclassified instance from 50 most important mean vectors of each class, results were inconclusive.

ii. Nearest Neighbors - For each misclassified instance we examined its 5 nearest neighbors from the train set. One instance had 3/5 bots as NN, 3 instances had 1/5 bot as NN, the rest had all humans as NN. The similarity prevents correct classification

3.7.2. **Resampling**

To tackle the issue of imbalance in our train data, we attempted to train our models on a synthetically generated resampled version of our train dataset and by using class-based training weights. This was done after feature extraction and before model training. We used SMOTE to add new synthetic data points to the dataset. Unfortunately, this hasn't improved the average precision scores. We believe this is because the output resampled data requires an additional stage of data cleaning.

Model	Before SMOTE		After SMOTE	
	Train	*Test*	*Train*	*Test*
Random Forest	*0.739*	*0.354*	*0.618*	*0.262*
Gradient Boost	*0.961*	*0.278*	*1.0*	*0.274*
XGBoost	*0.805*	*0.392*	*0.655*	*0.229*

3.8. Insights and applications

The main issue we had to deal with is the highly imbalanced data (95% humans). the high imbalanced data led to the fact that we had only ~100 samples from the minority class which led to insufficient learning of the minority class and thus it was misclassified a lot more. Kaggle's judgment factor was AUC. We believe that this factor is not sensitive enough for the minority class, and the problem would become more interesting if it was measured by metrics that focus in detecting the minority class.

As we can see in figures 4.2.1 we can say that time series analysis produced the most influential features (***max bid count in 10, 30, 60 mins, device, url, ip change times***) and therefore we can say that time series analysis is very efficient in detecting unhuman behavior.

3.9. Conclusion

In this real-life important problem, different machine learning models have different results leveraging different features. In auction fraud classification time-based features and auction activity features are important for differentiating humans and robots. We believe that the solution can be generalized to other fraud and bot recognition problems in different fields. The main problems is managing the inherent skewedness of the datasets due to different numbers of robots and humans.

4. Appendix

4.1. References

References

[1] Kaggle, " Facebook Recruiting IV: Human or Robot?," [Online]. Available: https://www.kaggle.com/c/facebook-recruiting-iv-human-or-bot.

[2] "Auction Sniper," [Online]. Available: https://auctionsniper.com/.

[3] C. Packer and W. Huang, "Bid-war: Human or Robot?," [Online]. Available: https://pdfs.semanticscholar.org/167c/96db9b78e9629861ac699f105b87943c2bc6.pdf.

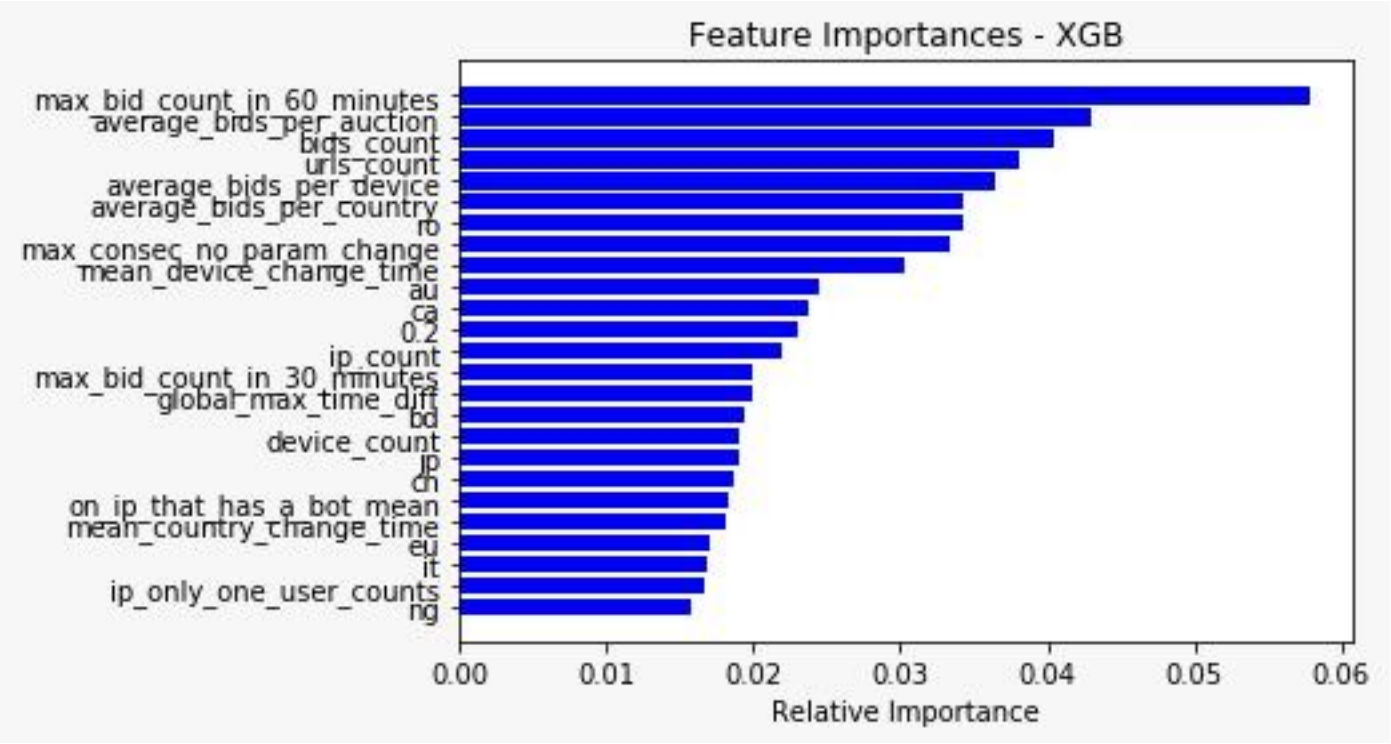
[4] X. Gu and S. Shi, "Human or Robot," [Online]. Available: http://cs229.stanford.edu/proj2017/final-reports/5161146.pdf.

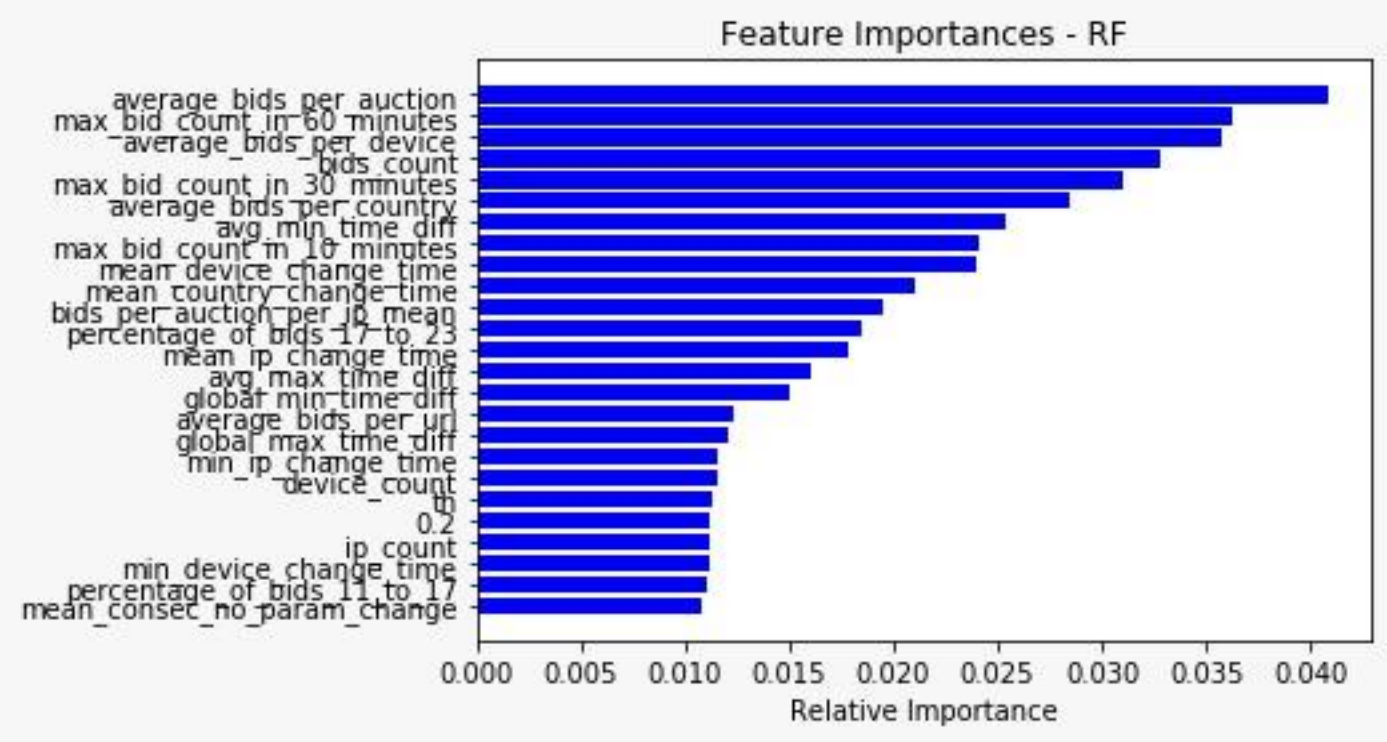
[5] "Robot, Bid," [Online]. Available: http://www.bidrobot.com/cool/.

[6] V. Nivargi and M. Bhaowal, "Machine Learning Based Botnet Detection," [Online]. Available: http://cs229.stanford.edu/proj2006/NivargiBhaowalLee-MachineLearningBasedBotnetDetection.pdf.

4.2. Important Figures

4.2.1. Feture importance of XGBoost and Random forest (highest scores)




Feature Importances - RF
average_bids_per_auction
max_bid_count_in_60_minutes
average_bids_per_device
bids_count
max_bid_count_in_30_minutes
average_bids_per_country
avg_min_time_diff
max_bid_count_in_10_minutes
mean_device_change_time
mean_country_change_time
bids_per_auction_per_ip_mean
percentage_of_bids_17_to_23
mean_ip_change_time
avg_max_time_diff
global_min_time_diff
average_bids_per_url
global_max_time_diff
min_ip_change_time
device_count
ip
0.2
ip_count
min_device_change_time
percentage_of_bids_11_to_17
mean_consec_no_param_change
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040
Relative Importance

4.2.2. sdsd