ETHEREUM FRAUD DETECTION

By Ethan Kunin

CRYPTOCURRENCY BACKGROUND

- Bitcoin invented in 2009
- Laszlo Hanyecz pays 10,000 BTC for 2 boxes of pizza
- Mt. Gox is hacked hacked, 850,000 BTC are stolen
- Ethereum proposed in 2013, completes crowdfunding in 2014
- Ethereum is minted in 2015 by Vitalik Buterin
- 2017 ICO bubble crashes
- August 2020, Michael Saylor invests $250 million of MicroStrategy's corporate reserves in Bitcoin
- Decentralized Finance gains momentum and protocols are being executed on the Ethereum blockchain

CRYPTOCURRENCY FRAUD

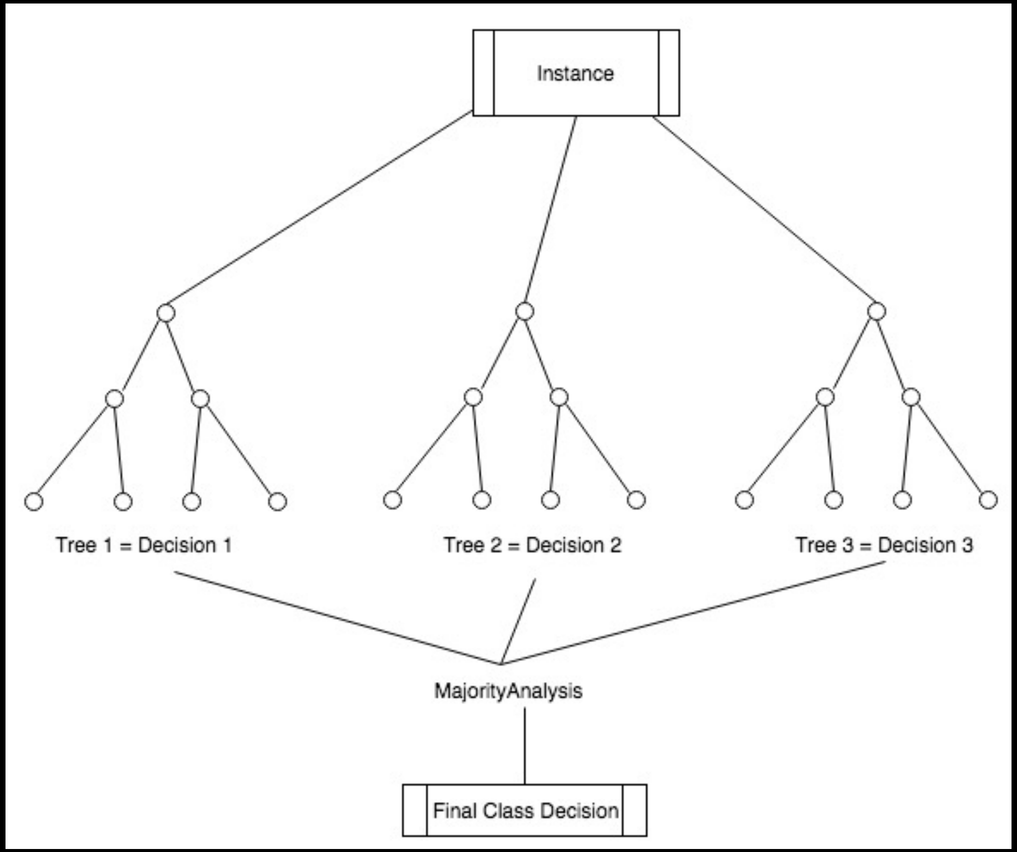
- Fraud is common in cryptocurrency transactions because they are **decentralized** and **anonymous**
  - Investing in fraudulent projects
  - Buying a good or service and not receiving it in return
  - Exchanges can be attacked
  - Flash loans

KEY FEATURES USED FOR MODELING

- **Total ERC20 transactions:** ERC20 tokens are representations of the applications using the Ethereum blockchain. ERC20 is a standard protocol designed by Ethereum developers that all tokens which trade on the Ethereum blockchain must follow. Wallets can measure the total amount of ERC20 transactions
- **Time difference between first and last transaction:** Measures the minutes between the first wallet transaction and most recent transaction. Can be used as a proxy for how long the wallet has been in use
- **ERC20 unique received addresses:** Measures the number of unique wallets that have sent ERC20 tokens to the respective wallet user
- **Weights Received non-ERC20 token users:** Represents if a wallet user has received ERC20 tokens (binary)

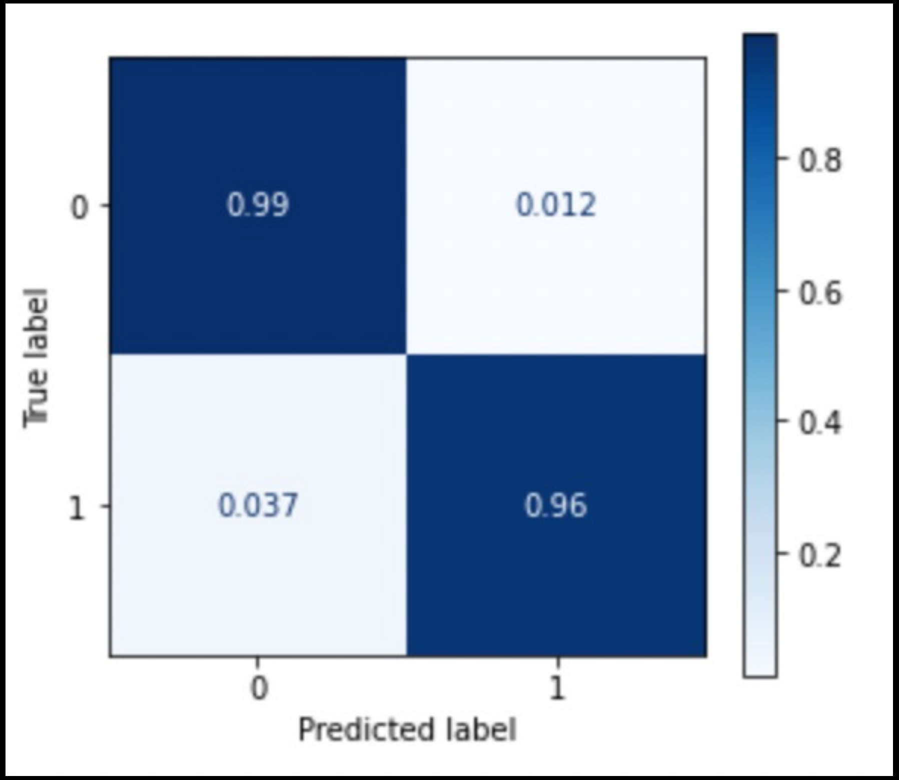
BUILDING A RANDOM FOREST MODEL TO DETECT FRAUD

- Trained on a dataset with **10,000** observations
- Features are based on **Ethereum** and **ERC20** transactions
- Due to the nature of the model type, it is **outlier resistant**
- **98% accurate** for detecting fraud and valid transactions
  - Accurately classifies valid transactions 99% of the time
  - Accurately classifies fraudulent transactions 96% of the time



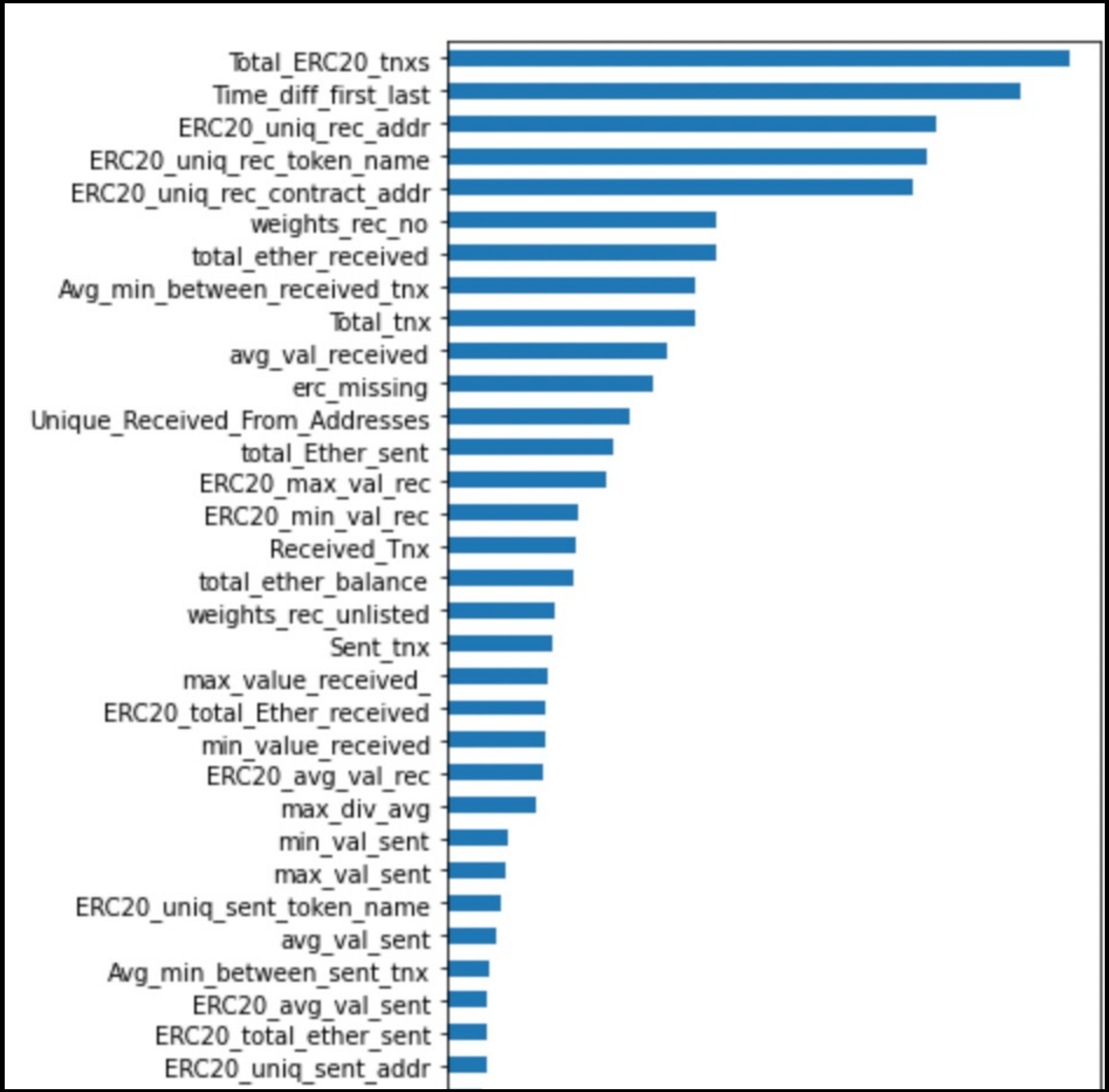
MODEL CONTINUED…

- Purpose of the model is to optimize for recall
  - True Fraud transactions divided by True Fraud transactions + Falsely classified Fraud transactions
- Cost of notifying customer that a valid transaction may not be valid is very low
- Cost of not notifying a customer that a fraudulent transactions is fraudulent is very high ($756/transaction)



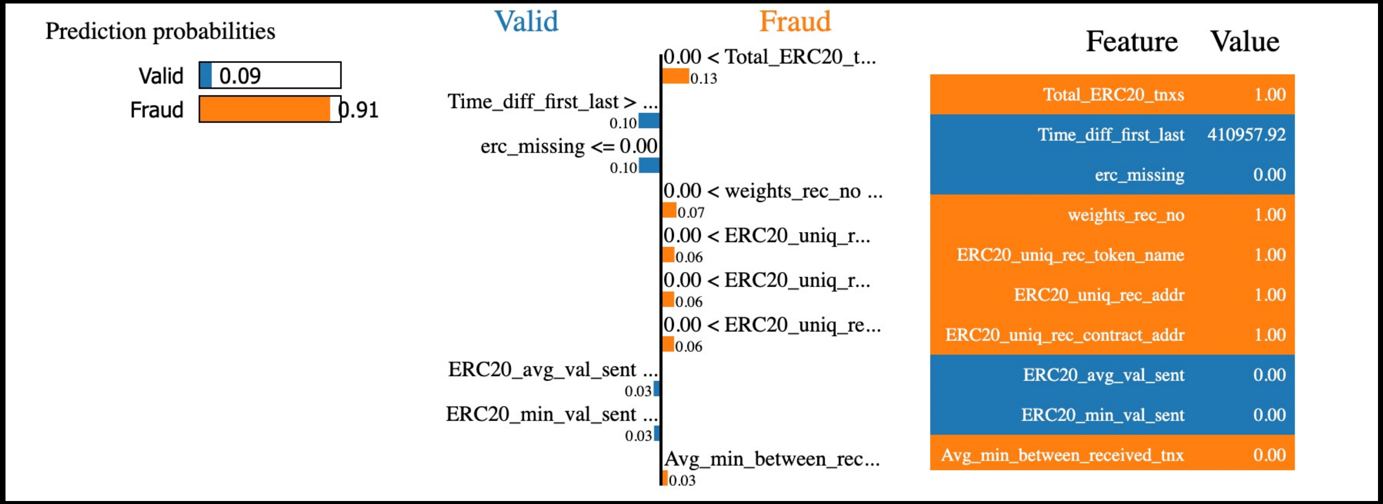
IMPORTANT FEATURES

- The most important features for classifying a transaction as fraudulent or valid are: **Total ERC20 tnx, time difference between first and last transaction, ERC20 unique received addresses, and ERC20 unique received token names**
- Greater number of transactions increases likelihood fraud
- Greater time difference between first and last transaction increases likelihood of fraud
- If a user uses ERC20 tokens they are less likely to be defrauded



MODEL DECISION MAKING PROCESS

- Random Forest Model Model
  - Each observation is processed by many different types of Decision Trees and classified as Fraud of Valid transaction
  - On the right, these are the various subcategories which the Decision Tree uses to classify the transaction type
  - The model builds 80 of these trees using different feature types so it will be able to generalize on new data
  - It then votes Fraud or Valid based based on the majority of votes by the trees



Feature	Value
Total_ERC20_tnxs	1.00
Time_diff_first_last	410957.92
erc_missing	0.00
weights_rec_no	1.00
ERC20_uniq_rec_token_name	1.00
ERC20_uniq_rec_addr	1.00
ERC20_uniq_rec_contract_addr	1.00
ERC20_avg_val_sent	0.00
ERC20_min_val_sent	0.00
Avg_min_between_received_tnx	0.00

MODEL COMPARISONS

- Tried two different types of models optimizing each for recall score
- Random Forest had a top recall score of 96% compared to 93% for K-Nearest Neighbors
- Used the Random Forest because it did a better job at generalizing for new data

Model	Total Customers	Expected Valid Tnx	Expected Fraud Tnx Attempts	Defrauded Tnx	Total Cost*	Avg. Expected Loss**
Dummy	10,000	9,500	500	250	$191,000	$19.11
K-Nearest Neighbors	10,000	9,500	500	40	$30,560	$3.27
Random Forest	10,000	9,500	500	19	$14,516	$1.55

* Calculated by multiplying average ETH fraudulent transaction value ($764) by number of expected fraudulent transactions

** Total cost divided by total number of customers

WALLET IMPLEMENTATION

- **Recommend** that wallet providers integrate this model into their product so that customers can be notified when they are making possible fraudulent transactions
  - Each fraudulent transactions costs customers $764 on average
- Customers put **faith** into their wallet providers that their coins will be kept safe
- Wallets are susceptible to **risk** when they are connected to exchanges or information on keys get leaked

THANK YOU FOR LISTENING

Are there any questions?