Metódy detekcie anomálií v reálnych údajoch

Analýza a návrh riešenia

Bc. Laura Vištanová

Vedúci práce: RNDr. Ľubomír Antoni, PhD.

Úvod

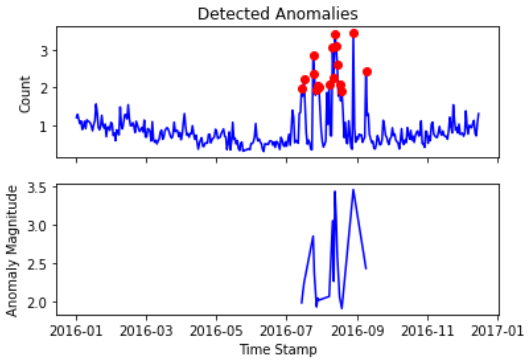
Žijeme v dobe, v ktorej sme zvyknutí, že sme neustále obklopení rôznymi zariadeniami, ktoré o nás zbierajú, ukladajú a spracúvajú dáta. Dáta sa vo vysokom množstve zbierajú a ukladajú každý deň, skutočnou výzvou pri nich však je vedieť s nimi narábať. V rámci analýzy dát vieme vydolovať informácie rôzneho typu. V tejto práci sa zameriame na analýzu, ktorá vyhľadáva v dátach také pozorovania, ktoré vyčnievajú z radu ostatných pozorovaní. Takéto pozorovania nazývame anomáliami, outliermi.

Detekcia anomálií v dátach má široké využitie v rôznych odvetviach. Využívajú ju v zdravotníctve pri analýze výsledkov EEG, MRI vyšetrení, pri skúmaní mozgovej aktivity alebo krvného tlaku. Metódy tiež vieme použiť pri analýze ľudskej činnosti, na kontrolu spotreby vody, elektriny v dome. V tejto práci sa zameriame na oblasť, ktorá sa zaoberá odhaľovaním podvodníkom, v anglickej terminológii „fraud detection“, a to konkrétne na odhaľovanie podvodníkov pri bankových transakciách.

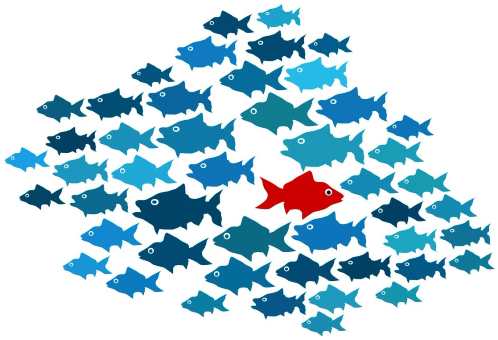
1 Definícia anomálie

Definícia anomálie nie je typická definícia akú by sme očakávali vo vedeckej práci. Anomáliou jednoducho nazývame také pozorovanie, ktoré vnímame ako ťažko vysvetliteľnú odchýlku od normálneho stavu. Nevieme presnejšie definovať tento pojem, pretože jeho spresnenie už závisí od konkrétneho problému, ktorý skúmame. Je zrejmé, že anomália vo výsledku EEG vyšetrenia bude vyzerať inak, ako v prípade anomálie v spotrebe domu, keďže aj normálne správanie systému je výrazne odlišné.

Môžeme sa tiež pozrieť na obrázok 1 a obrázok 2, na ktorých vidíme 2 rôzne situácie, ale v každej z nich anomália znamená niečo iné. Na obrázku 1 máme priebeh spotreby domu v jednej miestnosti, červené body sa vyhodnotili ako anomálie. Voľným okom vidíme, že naozaj v tých bodoch nadobúda funkcia vyššie hodnoty. Na druhom obrázku by asi každý hneď odhalil anomáliu, je to jednoducho červená rybka plávajúca opačným smerom ako ostatné.



Obrázok 2



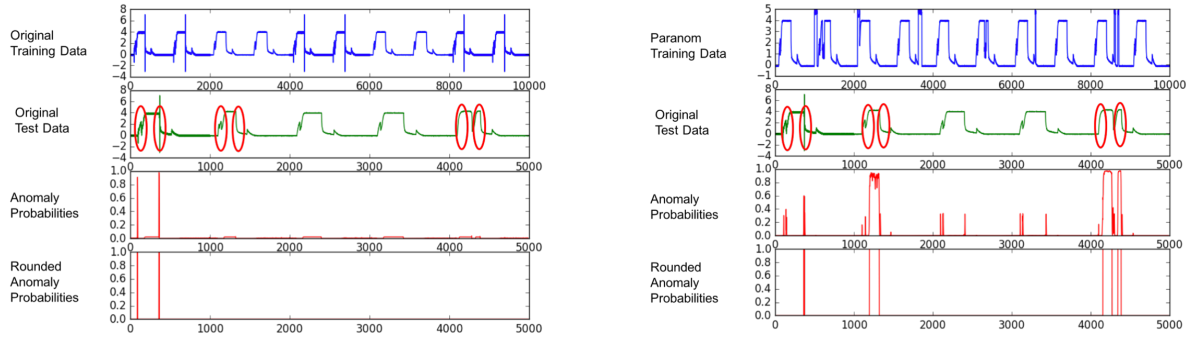
Obrázok 1

Anomálie majú jednu dôležitú vlastnosť, ktorá definuje základný problém detekcie anomálií. Anomálie sú z definície také pozorovania, ktoré sa výrazne odlišujú od normy, z čoho vyplýva, že sa v dátach objavujú iba zriedka. Keďže ich zastúpenie je v dátach veľmi malé, je náročné sa naučiť ich odhaliť.

Pre detekciu anomálií existuje viacero algoritmov, ktoré aj napriek nízkemu počtu anomálií sa dokážu naučiť ich odhaliť. Existuje však spôsob, ktorým vieme potencionálne vylepšiť tieto algoritmy. Predpokladáme, že keby sme mali k dispozícii väčší počet anomálií, vedeli by sme vytvoriť model, ktorý by bol presnejší, vedel by ich lepšie odhaliť. Keďže vlastných anomálií máme iba daný nízky počet, jedným zo spôsobov, ktoré sa dajú použiť je generovanie umelých anomálií.

2 Generovanie anomálií

Ako bolo v predchádzajúcej kapitole popísané, anomálie sú pozorovania, ktoré sa výrazne odlišujú od normy, z čoho vyplýva, že ich je nízky počet. Ak by sme chceli zlepšiť výsledky algoritmu na detekciu anomálií, môžeme mu pomôcť tak, že do tréningovej množiny pridáme umelé anomálie. Myšlienka vybrať sa v práci touto cestou pochádza z článku J. Gottschlich: Paranom: A Parallel Anomaly Dataset Generator, v ktorom sa podarilo zlepšiť presnosť modelu pridaním umelých anomálií do tréningovej množiny. Autor nepopisuje hlbšie myšlienky algoritmu, zaujímavé sú ale výsledky. Na obrázku 3 môžeme vidieť výsledky pôvodného modelu, ktorý vznikol z pôvodnej tréningovej množiny a výsledky nového modelu, ktorý vznikol použitím dátovej sady doplnenej o umelé anomálie. Môžeme vidieť, že kým pôvodný model odhalil iba 2 anomálie zo 6-tich, tak nový model odhalil až 5 anomálií, čiže neodhalil už iba jednu.



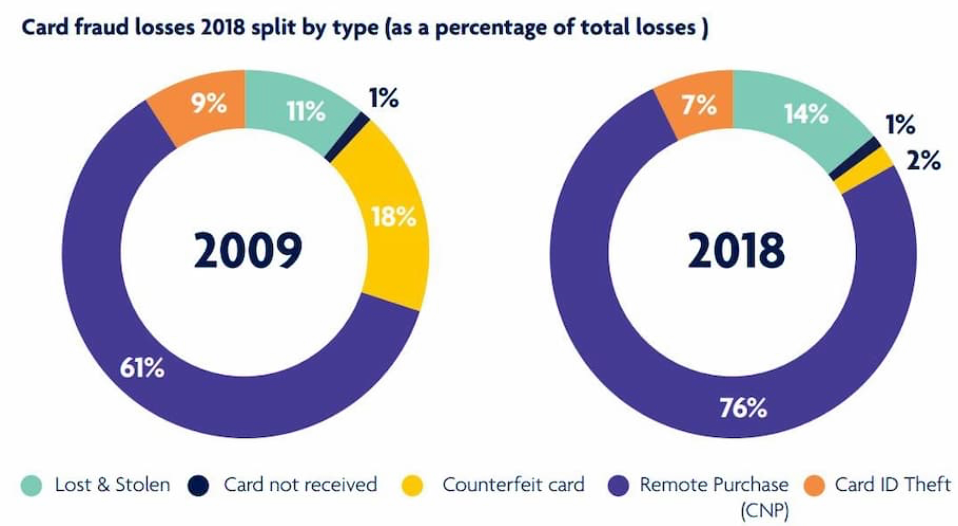
Obrázok 3

Výsledky tohto článku nás nabádali skúmať bližšie problematiku generovania anomálií. Zistili sme, že sa jedná o oblasť, ktorá ešte nie je dostatočne preskúmaná. Dostupných článkov je veľmi málo, všetky sú z posledných rokov. Napriek tomu, že anomáliami sa zaoberá veľa autorov, je spísaných veľa kníh a článkov, konkrétne o generovaní anomálií sa nám podarilo nájsť iba relatívne málo informácií.

3 Odhaľovanie podvodníkov v bankových transakciách

Podvodné transakcie predstavujú každý rok vysoké straty ako pre klientov, tak aj pre obchody, ktoré sú zodpovedné za straty. Väčšina podvodov sa stane cez CNP transakcie. CNP

transakcie (card-not-present transakcie) sú také platby, počas ktorých nie je fyzicky prítomná karta a vlastník karty. Typické CNP platby sú online platby alebo automaticky opakované mesačné platby. Pomer CNP platieb k ostatným druhom platieb v posledných rokoch neustále rastie, na obrázku 4 môžeme vidieť porovnanie z rokov 2009 a 2018. Rovnako každý rok rastú aj škody spôsobené podvodníkmi. Motivácia odhaliť podvodníkov pri platbách je vysoká, pretože za všetky straty sú zodpovedné obchody, do ktorých išla platba. Štatistiky z roku 2020 hovoria, že každý jeden dolár v podvodnej transakcii stál obchodníkov v priemere 3.36 dolárov. Je to spôsobené tým, že obchody musia vrátiť peniaze, ale aj prešetriť všetky transakcie, takže aj samotné analyzovanie a vyhodnotenie nahlásených transakcií ich niečo stojí.



Obrázok 4

Banky a obchodníci sa snažia rôznymi spôsobmi chrániť pred podvodníkmi. Bežne sa používa určenie hodnoty risk score v reálnom čase pre transakcie, na základe ktorého sa vyhodnotí, či je transakcia bezpečná. Rozvojom mobilných zariadení sa pridalo aj overenie majiteľa buď cez odtlačok prstu, overenie hlasu alebo rozoznávanie tvárí. Ďalšia možnosť na ochranu pred podvodníkmi je použiť metódy strojového učenia, čo bude aj náš prípad.

4 Dátová sada

V roku 2019 bola na Kaggle stránke vypísaná súťaž na odhaľovanie podvodníkov v CNP transakciách. Súťaž bola vypísaná spoločnosťou IEEE Computational Intelligence Society, pričom samotné dáta o transakciách poskytla spoločnosť Vesta, ktorá sa zaoberá

odhaľovaním podvodníkov v transakciách. Súťaž trvala pol roka, víťazom sa podarila dosiahnuť presnosť 94,6 %. Pre účely tejto práce sme sa rozhodli, že budeme skúmať víťazný model, pôvodnú tréningovú množinu doplníme o umelých podvodníkov a zopakujeme postup víťazov na modifikovanej dátovej sade. Výsledky potom porovnáme s pôvodnými výsledkami autorov, pozrieme sa, či sa nám podarí ešte zlepšiť výsledky.

Vesta poskytla do súťaže 2 dátové sady, ktoré majú spolu 434 atribútov. Medzi atribútmi sa nachádzajú informácie o kartách, adresách, vzdialenostiach a rôzne iné. Kvôli citlivosti dát nepoznáme význam všetkých atribútov, vieme ale, že niektoré atribúty sú také, ktoré si vytvorila už samotná Vesta, pretože ich považovala za dôležité. Je podstatné poznamenať, že pri všetkých transakciách máme poznačené, či sa jednalo o podvodnú transakciu alebo nie. V praxi máme často pozorovania bez toho, aby sme vedeli ktoré sa považujú za anomálie a ktoré nie, vtedy je ešte náročnejšie sa ich nejakým spôsobom naučiť.

4.1 Označenie podvodných transakcií

Aby sme vedeli správne odhaľovať podvodníkov, je dobré vedieť akým spôsobom sa jednotlivé transakcie označili za podvodné. Základným pravidlom je, že transakcia sa označí za podvodnú, ak ju niekto nahlási. Následne sa vyhľadajú všetky neskoršie transakcie, ktoré s ňou mohli súvisieť (rovnaká adresa, rovnaký email,...) a tiež sa označia za podvodné. Ak do 120 dní danú transakciu nikto nenahlási, tak sa považuje za platnú.

Samozrejme označenie jednotlivých transakcií v reálnom svete nie je až také jednoduché, v praxi napr. nechceme označiť za podvodné všetky transakcie asociované s daným bydliskom, nechceme totiž zablokovať všetky karty obete. V každom prípade pre potreby tejto práce je postačujúce pracovať so základným pravidlom uvedeným vyššie.

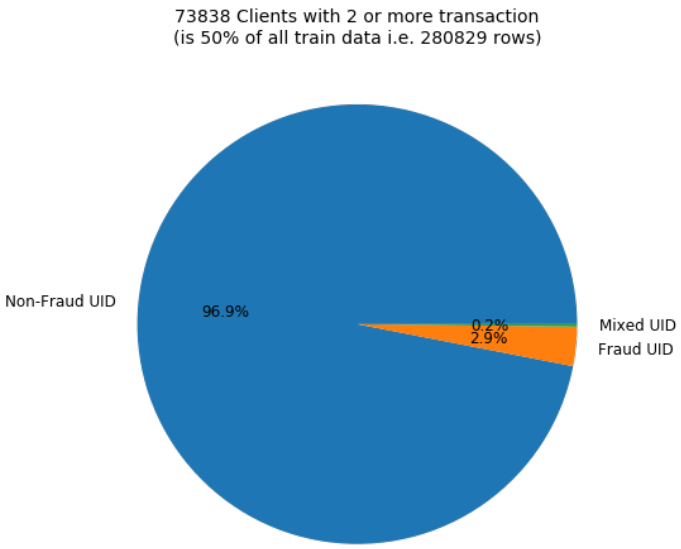
5 Víťazný model

V tejto kapitole predstavíme základné myšlienky víťazného modelu. Predstavíme myšlienku autorov, ktorá viedla k zlepšeniu presnosti modelu a tiež algoritmy ktoré použili pri hľadaní podvodníkov.

5.1 Identifikácia kariet

Autori víťazného modelu sa rozhodli, že nebudú skúmať podvodné transakcie, ale zamerajú sa na odhaľovanie podvodných kariet. V pôvodnej dátovej sade nemáme určené pri jednotlivých transakciách, že ktoré sú navzájom prepojené. Vzhľadom k tomu, že máme k dispozícii množstvo informácií o jednotlivých transakciách je možné ich prepojiť a týmto spôsobom identifikovať jednotlivé karty. Ak sa pozrieme na výsledky, tak môžeme vidieť, že sa podarilo v tréningovej sade odhaliť spolu 73 838 klientov, pričom uvažovali iba takých klientov, ktorí mali aspoň 2 transakcie.

Je zaujímavé sa zamyslieť aj nad tým, že ako vyzerajú jednotlivé transakcie v rámci jedného klienta. Výsledky na obrázku 5 ukazujú, že 99,8 % klientov má čisto buď len podvodné transakcie (2,9 %) alebo čisto len platné transakcie (96,9 %). Z toho vyplýva, že môžeme v podstate hovoriť o podvodných/platných *kartách*, ak zanedbáme tých 0,2 % u ktorých boli aj platné, aj podvodné transakcie.



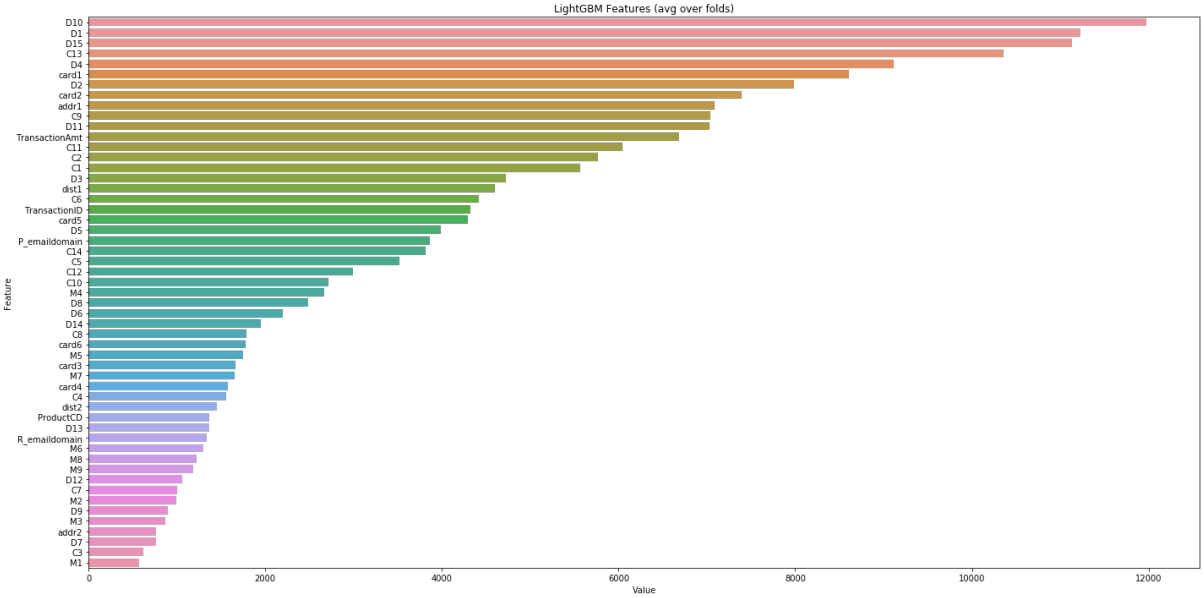
Obrázok 5

5.2 Postup pri identifikácii kariet

Autori víťazného modelu zvolili pri identifikácii nasl. postup:

1. Spojili dáta z tréningovej a testovacej množiny, pretože tieto dátové sady obsahujú jednak spoločných klientov, ale aj rozdielnych.

2. Na spojených dátach urobili klasifikáciu do dvoch tried. Táto klasifikácia jednotlivé riadky rozdelila späť do dvoch tried, na tréningovú a testovaciu množinu.
3. Následne sa pozreli na to, ktoré atribúty rozhodovali o tom, do ktorej množiny zaradiť jednotlivé transakcie. Výsledné atribúty sú znázornené na obrázku 6.



Obrázok 6

Ako môžeme vidieť, najdôležitejšie v klasifikácii boli atribúty, ktoré vyjadrovali rôzne vzdialenosti (napr. vzdialenosť IP adries) a informácie typu card, ktoré označujú typ karty, kategóriu karty, banku, krajinu.

Relatívne vysoko sa objavili aj atribúty s označením C, ktoré vyjadrujú počty. Význam väčšiny týchto atribútov je skrytý, ale vieme, že je medzi nimi aj atribút, ktorý obsahuje počet adries, ktoré sa dajú spojiť s danou transakciou.

Na obrázku 7 môžeme vidieť transakcie, ktoré sa podarilo spojiť pod jedného klienta. Ako môžeme vidieť, všetky transakcie sú podvodné, majú rovnakú hodnotu card1 a tiež si môžeme všimnúť, že stĺpec D3n odkazuje vždy na stĺpec day predchádzajúcej transakcie.

	TransactionID	isFraud	TransactionAmt	card1	addr1	D1n	day	D3n	dist1	P_emaildomain	UID
1694	2988694	1	240.0	15775	251.0	-81.0	1.0	0.0	NaN	yahoo.com	2988694.0
10046	2997046	1	260.0	15775	251.0	-81.0	3.0	1.0	NaN	yahoo.com	2988694.0
34029	3021029	1	250.0	15775	251.0	-81.0	9.0	3.0	NaN	yahoo.com	2988694.0
36812	3023812	1	315.0	15775	251.0	-81.0	10.0	9.0	NaN	yahoo.com	2988694.0
40459	3027459	1	390.0	15775	251.0	-81.0	11.0	10.0	NaN	yahoo.com	2988694.0
43926	3030926	1	475.0	15775	251.0	-81.0	12.0	11.0	NaN	yahoo.com	2988694.0
43941	3030941	1	445.0	15775	251.0	-81.0	12.0	12.0	NaN	yahoo.com	2988694.0
44717	3031717	1	445.0	15775	251.0	-81.0	12.0	12.0	NaN	yahoo.com	2988694.0
44727	3031727	1	445.0	15775	251.0	-81.0	12.0	12.0	12.0	NaN	2988694.0
58485	3045485	1	295.0	15775	251.0	-81.0	15.0	12.0	NaN	yahoo.com	2988694.0

Obrázok 7

5.3 Význam odhalenia kariet

Je dôležité si uvedomiť, že pri učení podvodných transakcií nie je vhodné používať hodnotu atribútu UID. Môže totiž dôjsť k preučeniu, pretože by si model nevedel poradiť s transakciami, ktoré patria k novým vlastníkom. Otázne teda je, prečo považovali autori za dôležité odhaliť jednotlivé karty.

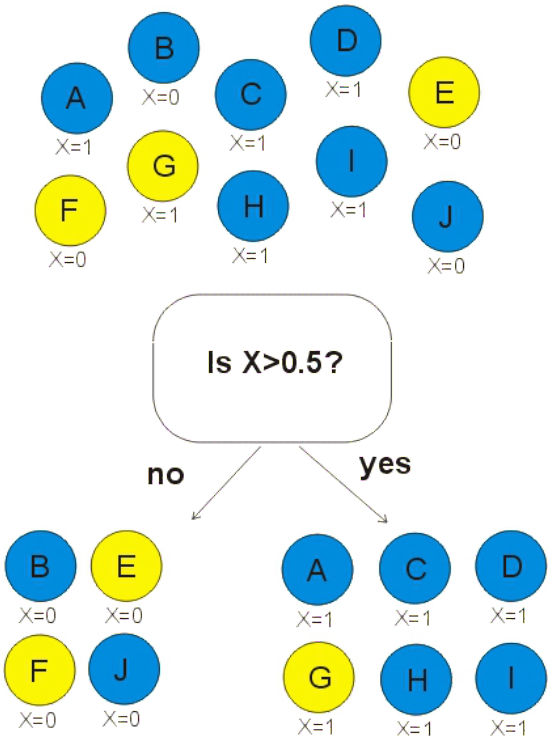
Potreba odhalenia kariet spočíva v tom, že na základe UID sa modifikujú atribúty dátovej sady. Presnejšie, každá hodnota atribútu sa nahradí priemernou hodnotou daného atribútu pre celú skupinu UID. Ukážeme si to na nasl. tabuľke, ktorú použili aj autori pri popísaní postupu:

TransactionID	UID	FeatureX	GroupX	IsFraud
A	1	1	0.75	1
B	1	0	0.75	1
C	1	1	0.75	1
D	1	1	0.75	1
E	2	0	0.33	0
F	2	0	0.33	0
G	2	1	0.33	0
H	3	1	0.66	1
I	3	1	0.66	1
J	3	0	0.66	1

Obrázok 8

Ako môžeme vidieť, tabuľka obsahuje 10 transakcií, ktoré patria pod 3 karty. Pôvodný atribút FeatureX obsahoval binárne dáta, ktoré sa nahradili atribútom GroupX. Ak sa pozrieme na prvú skupinu s UID = 1, tak môžeme vidieť, že do tejto skupiny patria transakcie A, B, C a D. Každá transakcia okrem transakcie B má hodnotu atribútu FeatureX rovnú 1, preto ich priemer bude hodnota 0.75. Túto hodnotu teda priradíme každej transakcii s UID = 1 do atribútu GroupX.

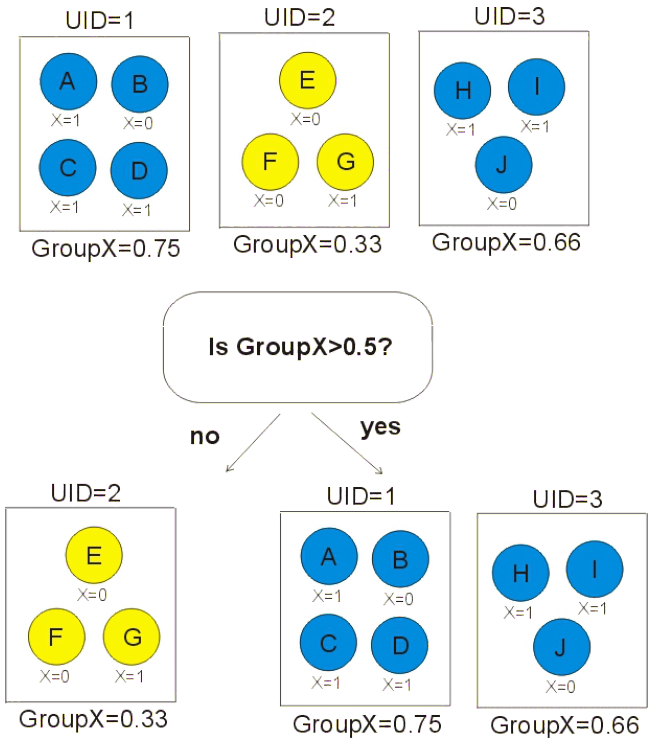
Význam tejto modifikácie dát si ukážeme na príklade z tabuľky. Ako môžeme vidieť, z desiatich transakcií sú len 3 platné, všetky ostatné sú podvodné. Predstavme si, že máme jednoduchú klasifikáciu, ktorá rozdelí tieto transakcie do dvoch množín (platné/podvodné) na základe hodnoty atribútu FeatureX. Nech je prahová hodnota pre tento atribút 0,5.



Obrázok 9

Ako môžeme vidieť na obrázku 9, ak by sme klasifikovali podľa hodnoty FeatureX, tak by sme neodhalili podvodné transakcie B, J a naopak by sme nesprávne vyhodnotili transakciu G za podvodnú. Pozrime sa teraz, ako dopadne klasifikácia, ak použijeme atribút GroupX, rovnako s prahovou hodnotnou 0,5.

Na obrázku 10 máme výsledky klasifikácie s modifikovaným atribútom GroupX. Môžeme vidieť, že v tomto prípade boli všetky transakcie klasifikované správne, teda má zmysel uvažovať modifikované atribúty. Je zrejmé, že pri danom postupe sa všetky transakcie patriace pod jedno UID klasifikujú do rovnakej kategórie, pretože majú rovnaké hodnoty atribútu.



Obrázok 10

5.4 Použité modely

Víťazný model pozostáva z 3 hlavných modelov, pretože autori zistili, že na jednotlivých typoch transakcií pracujú niektoré algoritmy lepšie ako ostatné. Použité modely sú nasledovné:

1. Catboost (0.963915 public / 0.940826 private) – predikoval dobre pre všetky skupiny dát
2. LGBM (0.961748 / 0.938359) – bol najlepší pre tie, kde sa podarilo identifikovať klienta
3. XGB (0.960205 / 0.932369) – bol najlepší pre tie, kde sa nepodarilo identifikovať klienta

6 Generovanie umelých anomálií

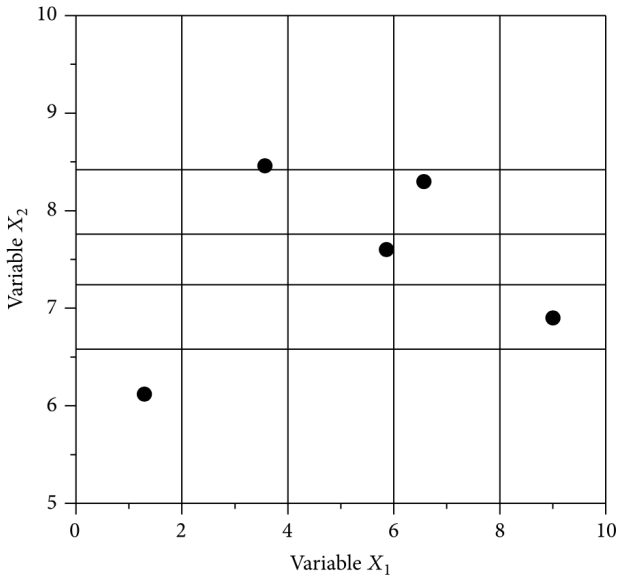
V tejto kapitole si uvedieme základné prístupy generovania umelých anomálií. Článok [7] je prehľadový článok o spôsoboch generovania umelých anomálií, ktorý obsahuje popis viacerých algoritmov. V tomto článku popíšeme spôsoby založené na výbere z rozdelenia a popíšeme štruktúru GAN sietí, ktoré môžu byť tiež vhodné pri generovaní umelých anomálií.

6.1 LHS vzorkovanie

LHS vzorkovanie je jedným z najjednoduchších spôsobov generovania umelých anomálií. Jediné, čo potrebujeme na vstupe sú prípustné intervaly pre jednotlivé atribúty. Ak máme k dispozícii množinu reálnych anomálií, tak intervaly zvolíme na základe hodnôt jednotlivých atribútov v množine reálnych anomálií. Následne vznikne z intervalov pre d atribútov d-dimenzionálny obdĺžnik.

Predpokladajme, že chceme vytvoriť n umelých anomálií. Každý interval rozdelíme na n podintervalov tak, aby pravdepodobnosť každého podintervalu bola rovnaká. Následne získame mriežku v d-dimenzionálnom obdĺžniku, z ktorej vyberieme n bodov tak, aby pre dvojicu ľubovoľných bodov platilo, že neexistuje atribút, v ktorom by mali hodnotu z rovnakého podintervalu. Pri danom rozdelení d-dimenzionálneho obdĺžnika vždy vieme vybrať n takýchto bodov.

Pozrime sa na výsledok algoritmu pri dvoch atribútoch a generovaní piatich pozorovaní, ktorý sa nachádza v článku [15]. Keďže máme dva atribúty, vznikne obdĺžnik, ktorého strany rozdelíme na 5 intervalov.



Obrázok 11

Ako môžeme vidieť, strany obdĺžnika nie sú rozdelené rovnomerne v prípade oboch atribútov. Pre premennú jedna (na osi x) môžeme vidieť, že má naozaj rovnomerné rozdelenie na intervale [0,10], ale pre premennú 2 to už neplatí. Rozdelenie jednoznačne nie je rovnomerné. Vo vnútri

obdĺžnika vidíme 5 čiernych bodov, tie znázorňujú 5 vygenerovaných bodov. Vidíme, že naozaj medzi nimi nie je dvojica takých, ktoré by mali spoločný podinterval.

Ak to považujeme za vhodné, nemusíme pri určovaní intervalov uvažovať prísne pozorované minimum a maximum pre každý interval, môžeme si zvoliť nejakú percentuálnu hodnotu o ktorú predlžíme každý interval, napr. 10 %. V uvedenom príklade by teda namiesto intervalu [0,10] bol interval [-1, 11], aj keď samozrejme by sme dolnú hranicu intervalu neposúvali do záporných čísel ak by sme mali atribút pri ktorom to nemá zmysel.

6.2 GAN siete

Veľmi zaujímavým spôsobom generovania umelých anomálií sa zdá byť prístup cez GAN (Generative Adversarial Network) siete. Tento prístup bol navrhnutý Ianom Goodfellowom a jeho kolegami v roku 2014.

Základnou myšlienkou GAN sietí je to, že spolu súperia dve neurónové siete. Ak máme k dispozícii tréningovú množinu, tak pomocou tohto súperenia vieme dosiahnuť vytvorenie nových dát, ktoré úplne zapadnú medzi ostatné. Táto metóda sa najčastejšie používa na generovanie obrázkov, kde na vstupe máme nejaký obrázok a GAN sieť generuje ďalšie, veľmi podobné obrázky.

Obrázok 12

Na obrázku 12 môžeme vidieť obrázok, ktorý bol vytvorený pomocou StyleGAN na základe iných vstupných portrétov reálnych osôb. Ako môžeme vidieť, daný obrázok vyzerá ako reálna fotografia človeka.

Vývoj GAN sietí od roku 2014 prešiel rapídnymi zmenami. Ak sa pozrieme na obrázok 13, tak vidíme, že vygenerované tváre boli na začiatku ešte rozmazané a neboli úplne dokonalé,

kým v roku 2017 máme obrázok, ktorá vyzerá úplne reálne, nedokážeme ho odlíšiť od ozajstných fotiek.

Obrázok 13

6.2.1 Štruktúra GAN sietí

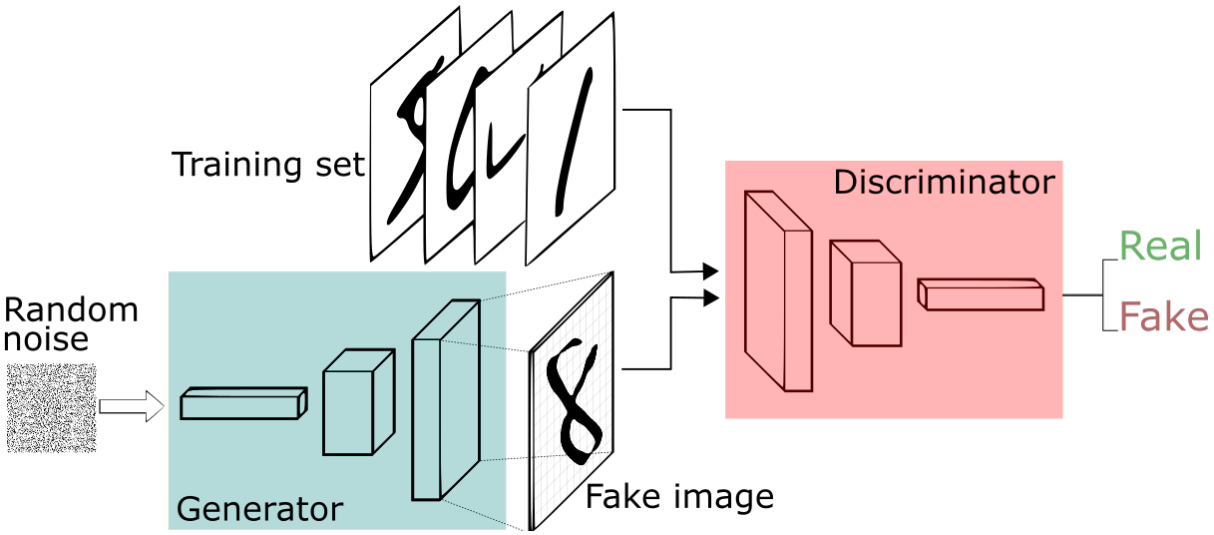
GAN siete, ako už bolo spomenuté, pozostávajú z dvoch neurónových sietí, ktoré spolu súťažia.

Prvá neurónová sieť sa nazýva *generátor*. Jej úlohou je generovať také dáta, ktoré sa čo najlepšie podobajú na dáta tréningovej množiny. Na vstupe dostane nejaký náhodný šum, z ktorého následne generuje nové dáta.

Druhá neurónová sieť sa nazýva *diskriminátor*. Jej úlohou je klasifikovať vstupné dáta do jednej z dvoch množín – reálne/falošné dáta. Na vstupe dostáva jednak dáta tréningovej množiny, ale aj dáta vygenerované generátorom.

Celý algoritmus pozostáva z dvoch častí, ktoré sa opakujú dovtedy, kým už diskriminátor nie je schopný rozoznať reálne vstupy od umelo vygenerovaných. V každej iterácii algoritmu sa obe siete zlepšujú. Generátor sa snaží vždy zlepšovať vygenerované dáta, aby ich diskriminátor neodhalil medzi reálnymi dátami a ako sa zlepšuje generátor, tak aj diskriminátor sa musí neustále zlepšovať, pretože dostáva stále reálnejšie vyzerajúce dáta od generátora.

Na obrázku 14 môžeme vidieť náčrt GAN siete. Vidíme, že generátor má na vstupe náhodný šum a na výstupe vygenerované falošné dáta. Diskriminátor má zase na vstupe zmiešané reálne a falošné obrázky, pre každý z nich sa rozhodne či je pravý alebo nie.



Obrázok 14

Zdrojový kód pre GAN sieť nájdeme na stránke [18].

6.2.2 Použitie GAN sietí pri odhaľovaní podvodných transakcií

Myšlienka generovania reálne vyzerajúcich obrázkov pomocou GAN sietí sa dá preniesť aj do riešenia problému tejto práce. K dispozícii máme síce malý počet, ale reálnych podvodných transakcií. Pomocou nich by sme mohli skúsiť generovať umelé podvodné transakcie a pridať ich do tréningovej množiny. Následne na modifikovanej tréningovej množine by sme mohli porovnať výsledky víťazného modelu s výsledkami pôvodnej množiny. O použití GAN sietí na generovanie umelých anomálií sa píše v článku [6]. Tento článok plánujeme dôkladnejšie preštudovať a pomocou neho vyskúšať GAN siete na generovanie umelých transakcií.

Záver

V tomto článku sme sa venovali generovaniu umelých anomálií v reálnych dátach. Poukázali sme na dôležitosť analýzy anomálií, na široké uplatnenie týchto metód v reálnom svete a tiež na problém malého počtu anomálií v dátach.

V prvých dvoch kapitolách sme si predstavili definíciu anomálie, jej základné vlastnosti a zmysel generovania umelých anomálií vo všeobecnosti.

V tretej kapitole článku sme sa pozreli bližšie na problém odhalenia podvodných transakcií, dôležitosť riešenia tohto problému a tiež na možné spôsoby bojovania proti podvodníkom.

Štvrtá a piata kapitola sa zamerala na riešenie konkrétneho problému zo súťaže Kaggle. Predstavili sme si dátovú sadu s ktorou budeme pracovať, jej základné vlastnosti. Následne sme predstavili víťazný model súťaže, algoritmy, ktoré použili a tiež základnú myšlienku ktorou sa im podarilo zlepšiť presnosť odhalenia podvodných transakcií.

Posledná, šiesta kapitola bola venovaná dvom algoritmom na generovanie umelých podvodníkov. Prvý algoritmus bol jednoduchý na pochopenie aj na implementáciu, napriek tomu ale dosahuje na reálnych dátach dobré výsledky, preto je vhodné ho spomenúť a vyskúšať. Použitie GAN sietí už nie je také jednoduché, ale vyzerá veľmi sľubne. Ak dokáže vygenerovať obrázky, ktoré nedokážeme rozoznať od ozajstných fotografií, tak by sieť nemala mať problém ani s generovaním numerických dát.

Práca v tejto oblasti je zatiaľ ešte relatívne na začiatku. V budúcnosti máme v pláne hlbšie porozumieť použitým dátam, víťaznému modelu a najmä generovaniu umelých podvodných transakcií, ktoré bude hlavným cieľom diplomovej práce. V rámci diplomovej práce vyskúšame viacero spôsobov generovania umelých podvodných transakcií, ktoré následne porovnáme jednak s pôvodnými výsledkami bez umelých podvodníkov, ale aj medzi sebou, aby sme videli, ktorý spôsob generovania anomálií je najlepší pre daný problém.

Zdroje

[1] Štatistiky: https://www.merchantsavvy.co.uk/payment-fraud-statistics/

[2] GAN siete: https://umelainteligencia.sk/gan-siete-zblizka/

[3] Dátová sada: https://www.kaggle.com/c/ieee-fraud-detection

[4] Víťazný model: https://www.kaggle.com/cdeotte/xgb-fraud-with-magic-0-9600

[5] Generovanie anomálií: https://arxiv.org/pdf/1801.03164.pdf

[6] https://research.fb.com/wp-content/uploads/2018/11/AnoGen-Deep-Anomaly-Generator.pdf

[7] https://arxiv.org/pdf/2006.03646.pdf

[8] https://risk.lexisnexis.com/insights-resources/research/2020-true-cost-of-fraud-retail

[9] https://squareup.com/us/en/townsquare/what-is-a-card-not-present-transaction

[10] https://spd.group/machine-learning/credit-card-fraud-detection/

[11] https://traderdefenseadvisory.com/financial-protection/credit-card-fraud/

[12] https://thedatascientist.com/anomaly-detection-why-you-need-it/

[13] https://www.merchantsavvy.co.uk/payment-fraud-statistics/

[14] https://www.smartrvworld.com/rving-identity-safety/cartoon-thief/

[15] https://www.hindawi.com/journals/mpe/2015/450492/

[16] https://github.com/cysmith/neural-style-tf

[17] https://arxiv.org/pdf/1802.07228.pdf

[18] https://wiki.pathmind.com/generative-adversarial-network-gan